



Decision-making of IoT device operation based on intelligent-task offloading for improving environmental optimization

Wenquan Jin¹ · Sunhwan Lim² · Sungpil Woo² · Chanwon Park² · Dohyeun Kim³

Received: 22 August 2021 / Accepted: 1 January 2022 / Published online: 11 February 2022
© The Author(s) 2022

Abstract

Computation offloading enables intensive computational tasks to be separated into multiple computing resources for overcoming hardware limitations. Leveraging cloud computing, edge computing can be enabled to apply not only large-scale and personalized data but also intelligent algorithms based on offloading the intelligent models to high-performance servers for working with huge volumes of data in the cloud. In this paper, we propose a gateway-centric Internet of Things (IoT) system to enable the intelligent and autonomous operation of IoT devices in edge computing. In the proposed edge computing, IoT devices are operated by a decision-making model that selects an optimal control factor from multiple intelligent services and applies it to the device. The intelligent services are provided based on offloading multiple intelligent and optimization approaches to the intelligent service engine in the cloud. Therefore, the decision-making model in the gateway is enabled to select the best solution from the candidates. Also, the proposed IoT system provides monitoring and visualization to users through device management based on resource virtualization using the gateway. Furthermore, the gateway interprets scenario profiles to interact with intelligent services dynamically and apply the optimal control factor to the actual device through the virtual resource. For implementing the improved energy optimization using the proposed IoT system, we propose two intelligent models to learn parameters of a user's residential environment using deep learning and derive the inference models to deploy in the intelligent service engine. The inference models are used for predicting a heater energy consumption that is applied to the heater. The heater updates the environment parameters to reach the user-desired values. Moreover, based on two energy consumption values, the decision-making model brings a smaller value to operate the heater to enable reducing the energy consumption as well as providing a user-desired environment.

Keywords Internet of things · Edge computing · Intelligent systems · Decision-making · Energy management · Neural networks · Smart homes

✉ Dohyeun Kim
kimdh@jejunu.ac.kr

Wenquan Jin
wenquan.jin@jejunu.ac.kr

Sunhwan Lim
shlim@etri.re.kr

Sungpil Woo
woosungpil@etri.re.kr

Chanwon Park
cwp@etri.re.kr

¹ Big Data Research Center, Jeju National University, Jeju, Republic of Korea

² Autonomous IoT Research Section/Intelligent Convergence Research Laboratory, Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea

³ Department of Computer Engineering, Jeju National University, Jeju, Republic of Korea

Introduction

Recently, the Internet of Things (IoT) tries to connect massive entities such as devices, applications, and information through the Internet to provide heterogeneous services in various industrial domains. As an emerging engineering paradigm, the IoT inspires industries to deploy smart and autonomous systems based on connected devices. The pervasive IoT applications involve multiple technologies including identification, networking of sensor and actuator, cloud and edge computing, and big data with approaches of stochastic and heuristic [1,2]. Heterogeneous IoT applications have been deployed in many industrial domains for monitoring, control, security, and intelligent systems [3–5]. According to the development of IoT technologies, the number of Internet-connected devices is increasing rapidly for providing ubiquitous and seamless services in various indus-

trial domains without human touch to operate smart and autonomous scenarios day and night [6,7]. For managing the massive IoT devices, the identifiers including Identifier (ID) and Uniform Resource Identifier (URI) are assigned to identify cyber resources for linking the physical resources through the registered information of devices in the IoT networks [8]. The virtualization enables the cyber resources to represent the physical resources on the Internet and enhances the functionality through integrating multiple resources to provide improved services [9,10]. However, these services require a sufficient environment for computation and network to support complex computation and service scenarios based on the virtualization of resources in the cyber world.

Most IoT devices are developed for a constrained hardware environment with limited power supply, processor, storage, and network capability which is difficult to support a sufficient computation and network requirement [11–13]. For supporting IoT devices to implement complex application scenarios based on a rich hardware environment, cloud computing is a solution to provide applications and databases to handle the massive IoT data and service requests [14–16]. However, supporting a low-latency delivery of data between IoT devices and cloud servers is difficult due to network bandwidth and computation load [17,18]. Edge computing brings the cloud to the edge of networks for deploying the computation and storage close to the constrained IoT devices to support computation based on low-latency communications [19–21]. As an intermediate layer between device and cloud, the entities in this layer bridge IoT devices and the Internet-connected entities such as web servers and clients to support message translation, computational offloading, and device management [22–25]. An edge gateway is a device that is deployed in the entry of the network edge to provide services to the IoT devices and the Internet-connected entities based on processor, storage, and network modules [26–28]. For extending the computation and storage of IoT devices to provide improved services, the edge gateway is the key to enable complex computation and service scenarios close to the constrained environment.

Nevertheless, heavy computation tasks can be offloaded to cloud servers to process the increasing data and computing demands through remote execution [29–31]. Computation offloading enables intensive computational tasks to be separated into multiple computing resources to overcome the hardware limitation. Leveraging the cloud computing, the edge computing can be supported to apply not only large-scale and personalized data but also Artificial Intelligence (AI) algorithms based on offloading AI approaches to high-performance servers to work with huge volumes of data in the cloud [32]. Machine Learning (ML) is a stochastic approach to implement the AI base on sample data to make predictions or decisions. For providing an intelligent approach based on ML, first, the learning model derives an inference model

using training data with an ML algorithm. Then, the inference model can be deployed to predict the environment with required input parameters. Besides, online ML is a mechanism to extend the knowledge of the existing inference model through continuous input data [33,34]. The updating process of the existing inference model using new training data can be executed parallelly in the cloud, while the intelligent service is provided based on the existing inference model. Therefore, deploying AI approaches to the cloud and providing web services to edge computing can enable intelligent operations of IoT devices in personalized spaces based on edge gateways.

For providing improved energy optimization, we propose a gateway-centric IoT system to enable intelligent and autonomous operation of IoT devices through supporting decision-making based on multiple intelligent services in edge computing. The proposed IoT system is comprised of IoT client, Application Service Provider (ASP), Intelligent Service Engine (ISE), Edge Gateway (EG), and IoT device to provide the smart service to users through offloading the smart models in the ISE. The ISE provides services based on the smart models that derive the intelligent and optimized control factors. The models can be implemented based on deep learning with user-specific data for providing personalized smart services in the edge computing environment. Through interpreting the deployed scenario profile, intelligent services can be invoked dynamically by the EG, and the service results are used by the decision model to be selected as the optimal control factor. Based on the proposed IoT architecture, a smart IoT device operation scenario is developed, which derives the best energy consumption value to operate the heater in the proposed edge computing environment. For providing intelligent services from the ISE, two intelligent models are proposed that are implemented based on deep learning approaches with a user's residential environment. The models derive a heater energy consumption that is used for operating the heater to update the environment parameters to reach user-desired values. Therefore, the decision-making model can select a smaller value from the ISE services to apply on the heater to enable reducing the energy consumption as well as providing a user-desired environment, which achieves the proposed energy optimization for operating the IoT devices in edge computing.

The rest of this paper is organized as follows. The second section reviews the related works regarding intelligent and optimization solutions based on IoT and edge computing and introduces referred technologies of the proposed IoT system. The third section presents the proposed gateway-centric IoT system architecture with the elements including IoT client, ASP, ISE, EG, and IoT device. The fourth section introduces the intelligent IoT scenario for providing improved energy optimization. The fifth section introduces the decision-making model and offloaded intelligent and optimization solutions. The sixth section presents the imple-

mentation details of the proposed IoT system and its results. The seventh section evaluates the outputs of the IoT system. Finally, we conclude this paper and introduce our future directions in last section.

Related works

The IoT is a system that is comprised of heterogeneous elements to provide autonomous and intelligent services in various industrial domains. For developing the communication between the EG and IoT device, we use the IoTivity that is a framework to develop a light-weight IoT system based on CoAP [35–37]. IoTivity is implemented through Open Connectivity Foundation (OCF) is an IoT standard specification including the definition of connectivity, management, functions, and security [38–40]. The oneM2M is another popular international IoT standard that includes all required functions for operating an IoT system in large-scale environments based on a high-performance machine to provide the sufficient computing and data storage [41,42]. Therefore, oneM2M is not suitable for edge computing using constrained devices. The Light-Weight Machine-to-Machine (LWM2M) is an Open Mobile Alliance standard that aims to develop an easy-deployment client and server modules for constrained devices [43]. However, the LWM2M only considers using CoAP for communication between device and device without interworking with multiple protocols. Cloudlet and mobile edge computing are solutions for implementing cloud computing in mobile devices to provide a small-scale computing and data repository on the edge of the networks [44–48]. EdgeX is an emerging edge computing framework that provides various functions based on microservices and easily extends the functionality and interacts with other solutions [49]. However, the core services are fixed, which is not allowed to provide a domain-specific data format and service scenario. The proposed architecture includes the gateway that enables the devices have multiple protocols to interworking with the cloud from the network edge.

Deep learning trains a large dataset to derive an inference model based on connected nodes of a network such as Deep Neural Network (DNN), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN). LeCun et al. [50] present an overview of deep learning including several popular models and future directions of deep neural networks. In edge computing, task offloading distributes computing resources to multiple machines for reducing computational load. Recently, most of task offloading solutions are used for enabling learning approaches in the high-performance computing unit [51]. Computational offloading solutions can be categorized as fully offloaded and dynamically offloaded [52]. Qiao et al. [53] proposed a col-

laborative task offloading and output transmission approach to efficiently merge mobile edge computing technology in vehicular networks. Zhang et al. [54] proposed an efficient offloading scheme to improve reliability for vehicular data transmission failure through deep learning. Crutcher et al. [55] proposed a utilized prediction approach through offloading the learning mechanism to the cloud for reducing resource consumption. Xu et al. [56] proposed an intelligent edge computing based on an online learning algorithm that is used for offloading the learning approach to the cloud for renewable power supply. Rui et al. [57] proposed an online algorithm to select edge nodes based on distributed decision-making. Zhang et al. [58] proposed an online learning offloading framework to handle the computation offloading problem in heterogeneous edge networks. Different from online learning, we propose parallel inferences to derive the results and select the optimal control factor to apply to the environment.

The deep learning models also can be enabled by embedded devices directly to provide intelligent services in the environment. Liu et al. [59] proposed a deep learning application for food recognition based on providing services in an edge computing environment. However, the application requires a high-performance computing unit in a mobile device for processing image data that is a large size data. Zhang et al. [60] used deep Q-learning in scheduling voltage and frequency for real-time systems in embedded devices. Alsheikh et al. [61] proposed an intelligent edge computing architecture with two layers that deploy deep learning in the mobile device and deploys Apache Spark to the cloud for distributing deep leaning tasks on multiple devices. Biswas et al. [62] proposed an advanced hardware architecture to provide a CNN training approach in the low-power embedded machines. Zhang et al. [63] proposed a distributed real-time object detection in an embedded device based on NVIDIA Jetson TX1 that brings similar performance using lower power. Beatriz et al. [64] implemented a low-power and real-time deep learning-based multiple object visual tracking using the NVIDIA Jetson TX2 development kit that integrates a powerful ML-specific processor in a mobile device. Nevertheless, offloading the large-scale computing to the cloud is proposed in this paper for distributing the computational load.

Controlling energy consumption is important in buildings for keeping the environment at a comfortable level with minimal cost. For the thermal comfort, multiple environmental parameters are considered including temperature, humidity, and air velocity [65]. Optimization models are proposed to provide optimal factors for controlling the actuators to update the environment [66–68]. Leveraging fuzzy inference models, the environmental parameters also can be updated based on parameter settings by experts [69]. Also, using the user data to train an RNN model can forecast energy consumption for the user-desired environment [70]. Furthermore, based on

the decision making, the control factor can be derived from multiple prediction models for a user-desired environment using lower energy consumption.

Proposed gateway-centric IoT system based on edge computing

The proposed IoT system is composed of IoT client, Application Service Provider (ASP), Intelligent Service Engine (ISE), Edge Gateway (EG), and IoT devices to provide device management, visualization, and intelligent services. Multiple entities can be deployed for each IoT element in the proposed IoT system. At the edge of a network, an EG is the center to link a group of IoT devices.

For depicting the proposed IoT system, a four-tier hierarchical architecture is presented in Fig. 1. The hierarchical architecture is divided into four layers including the layer of client, cloud, edge, and device.

The device layer includes sensors and actuators with IoT devices to provide IoT services based on collecting environment data through sensors and effecting the environment through actuators. IoT devices can be deployed to make a group for serving an environment where can be an environment of indoor and outdoor. IoT devices are equipped with limited computing capabilities for the constrained environment. Therefore, the computational tasks are distributed to the edge and cloud layer.

The edge layer is a local network that bridges IoT devices to the cloud layer through EGs. An EG is a gateway device that is deployed at the edge of the Internet with IoT devices. An EG is used for a group of IoT devices to forward registration, sensing, status data to the ASP, and deliver command data to IoT devices from the ASP. For supporting energy optimization based on intelligent services, a service scenario is used through deploying a profile on an EG. Based on the service scenario, the EG interacts with the intelligent layer and IoT devices. The decision-making model is deployed in EG to derive the control factor for operating IoT devices. The model requires the reference factors that are provided by the ISE based on intelligent services.

The cloud layer includes servers that are equipped with high-performance computing units and sufficient repository to support functions of data storage and computation. The servers are categorized into the service layer and intelligence layer for distinguishing the types of services base on the consumers. The APS is a service provider that provides services to clients for presenting information to users and delivering commands to IoT devices. The clients of these services are the IoT clients in the client layer as well as other devices such as ISEs in the intelligent layer. The ISEs run intelligent solutions based on a large amount of data which are retrieved from the APS. Also, for registering IoT devices to

manage these devices, the registration service is provided to IoT devices through the edge layer. In the intelligence layer, ISEs are deployed to provide intelligent services to the IoT environment. ISEs are service providers that are deployed on the Internet to provide intelligent services through supporting ML algorithms based on the pre-trained models. ISEs can be provided by multiple organizations to solve various industrial domains' problems. Therefore, users can have multiple choices to solve their problems by interacting with ISEs.

The client layer includes users with IoT clients to consume services from the cloud layer. The services are used for delivering information to users through these client devices through visualization functions using user interfaces. A client device is an IoT client that is deployed on the Internet to access services from the proposed IoT system. The generated data from the IoT device that can be stored in the DB as well as provided to IoT clients directly. For interacting with IoT devices and the smart space, the user interfaces of an IoT client support functions of management, control and data collection on an application based on a smartphone, desktop computer, or web front-end client.

Figure 2 shows a network architecture to present the interactions between elements in the proposed IoT system. ASP and ISEs are deployed in the cloud layer to provide services through the Internet for public network access. An Edge network is a local network that includes EGs and IoT devices to provide services for a private environment such as homes, factories, and buildings.

The ASP provides multiple services to clients which are devices such as mobile devices and desktop devices with general users, administrators, and experts. In the cloud network, the users through client devices consume IoT services including management, visualization, control, and smart services. The resources of ASP provide services to the Internet as well as the edge network. Clients request the ASP for retrieving data that are stored in the system or generated in real time, and sending commands to the system including IoT devices. For delivering service profiles to the edge network, the ASP includes web client to forward the request to an EG for applying a scenario. From the edge network, IoT devices request the registration service to upload device profiles and sensing data through EGs. ISEs are deployed on the Internet to provide intelligent services for a specific domain, private space, or collaborated in service scenarios. Heterogeneous intelligent functions can be supported using the services of ISEs. An intelligent function in an ISE that is pre-developed based on analysis of data in a specific domain or environment for supporting advanced solutions such as prediction and decision-making. For supporting intelligent services, ISEs interact with the ASP to request required data from the repository. Based on the intelligent functions, the controlling data are delivered to the edge network through

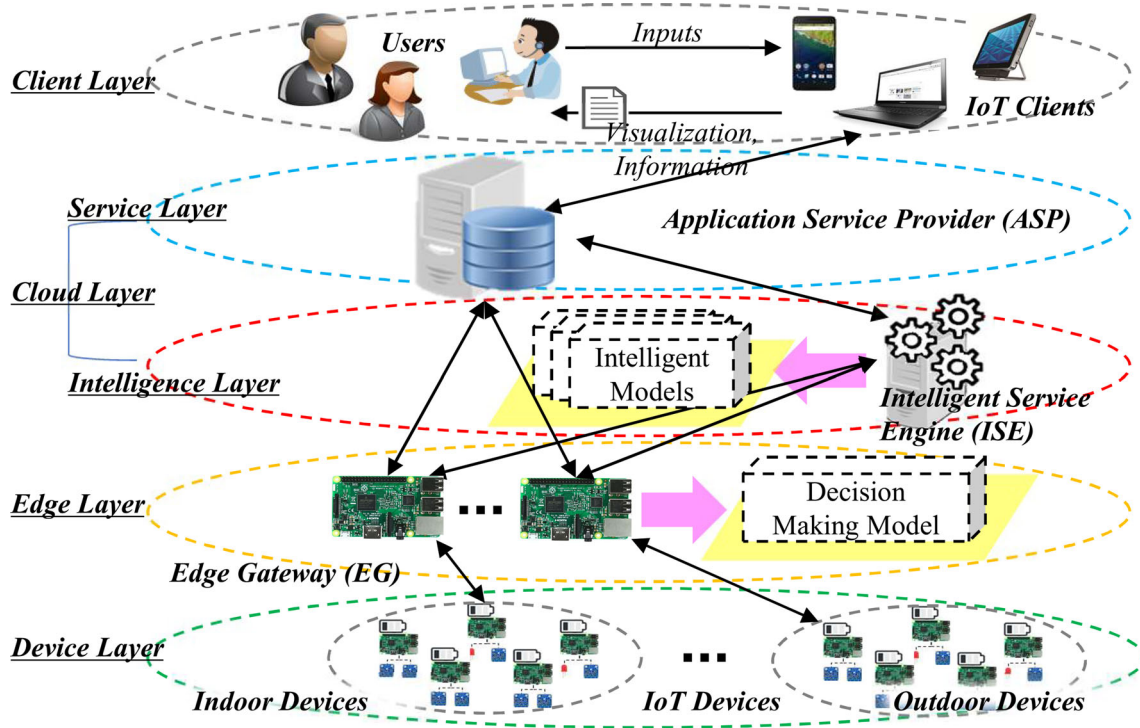


Fig. 1 Proposed IoT system hierarchical architecture

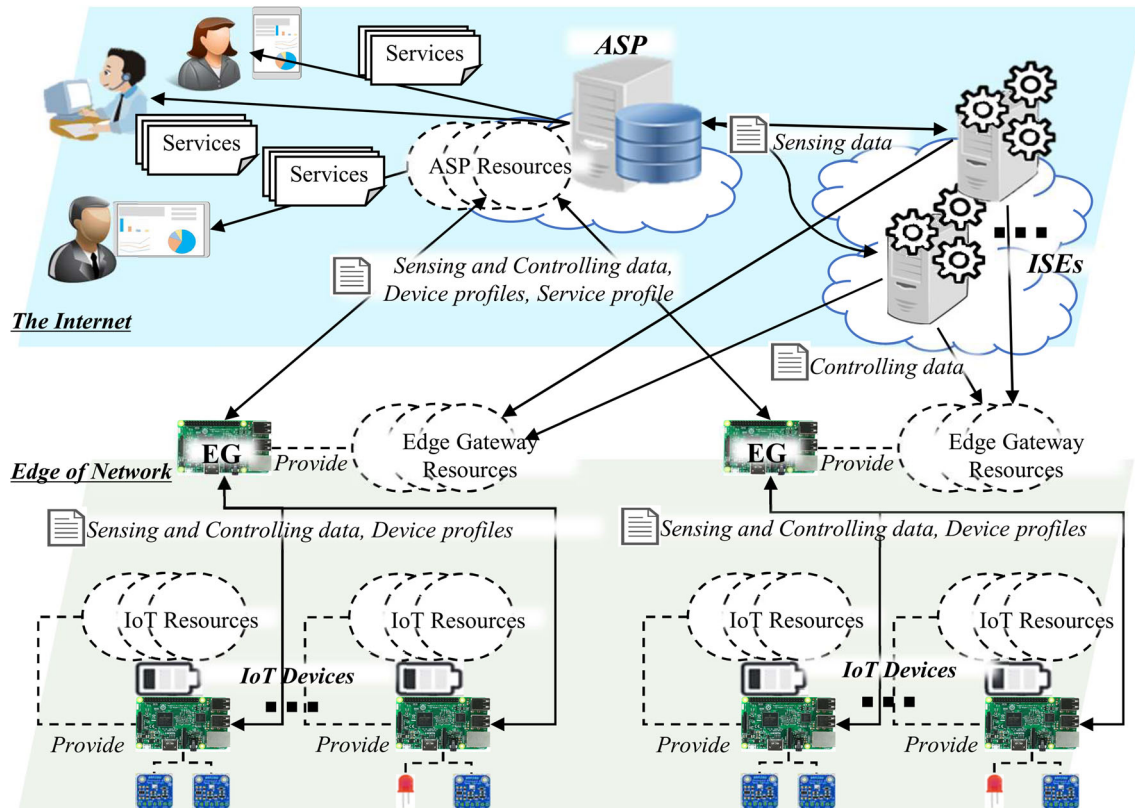


Fig. 2 Gateway-centric IoT architecture for intelligent edge computing

EGs for updating the environments using actuators on the IoT devices.

In the proposed edge computing, EGs are deployed to interact with the Internet including the APS and ISEs as well as IoT devices. The resources of EGs provides services to the Internet for forwarding messages to the IoT devices. For registering the information of devices to the ASP, EGs forward the profile to the ASP and store the data in its repository. The mapping information in the repository enables messages that can be forwarded to IoT devices through the EGs.

Proposed intelligent edge computing scenario

Based on the proposed IoT architecture, multiple intelligent models can be deployed in the cloud layer to provide intelligent services that are used for the decision-making to derive the optimal factor of IoT device operation. With the information of specific IoT entities, this scenario is presented in a profile that is interpreted by the EG to apply on the edge computing.

Figure 3 shows a sequence diagram to present the IoT device operating scenario based on the proposed IoT architecture. First, IoT devices register the information of IoT resources to the EG to be discoverable in the edge network. Then, the EG registers the information to the ASP to be discoverable on the Internet. The resources of IoT devices are represented in the edge network through the resources of EG. The represented resources are virtual resources that forward the request messages to the IoT devices. Once IoT devices are registered to the ASP, the EG collects environmental data including sensing data and actuator state from IoT devices and sends the data to the ASP for accumulating historical data to be used in intelligent functions. Then, the IoT client discovers the registered IoT resource on the Internet through accessing the ASP, and publish the service scenario profile for a group of IoT resources. The service scenario profile is deployed in an EG that operates a scenario based on the profile. In the process of operating a scenario, the EG communicates with intelligent services of an ISE to call the intelligent functions to get control values which are the energy consumption values for operating a heater in a testbed. The functions are built by intelligent ML models using the analysis of data. Therefore, the historical data are required in this process. Once the control values are received by the EG, then the EG runs an internal function to compare the values to get a minimum energy consumption value for controlling the heater.

Figure 4 shows a data pipeline for operating the IoT device operation scenario on the proposed intelligent edge gateway. The data pipeline presents the process of device registration, data collection, device discovery, scenario deployment, intel-

ligent services operation, and control through the data flow based on network communications. The IoT device includes resources that are used for sensing and actuating through IoT services. The information of the IoT device including resources is involved in the device profile. The IoT device sends the profile to the EG, and the EG sends the profile to the ASP. For collecting data from the IoT device, the requests from the EG, and the resources return the data to the EG. Then, EG requests to the ASP for storing the historical data. Through the discovery service from the ASP, the IoT client retrieves information of devices and resources, and deploys a service scenario profile to an EG through the ASP. Once the service scenario profile is applied to the EG, the EG requests intelligent services from the ISE to call the functions for getting results of intelligent models. The intelligent models run ML functions which are required historical data to predict a power value that is used in a heater for affecting the indoor environment. The EG calls multiple ML functions through the intelligent services from the ISE, then compares the results, and applies the minimal energy consumption to the IoT device.

The proposed intelligent edge computing provides a user-desired environment through intelligent services. The ISE includes intelligent models that are implemented through ML models with the user data to predict heater energy consumption for operating the heater to update the user environment as depicted in Fig. 5. Tow time status is presented which are time t for the current environment and time $t+1$ for the future environment. The edge gateway derives the heater's energy consumption HP for $t+1$ environment. The heater consumes the energy to update the current environment to be $t+1$ environment. The prediction model is trained by the user data that are comprised of time-sequence (TS), user-desired indoor temperature (IT), user-desired indoor humidity (IH), outdoor temperature (OT), outdoor humidity (OH), and heater energy consumption (HP). Therefore, using these data, the model can provide a predictable heater energy consumption for a user-desired indoor temperature and humidity based on the previous data. In the $t+1$ future environment, the updated $IT(t+1)$, $IH(t+1)$, $OT(t+1)$, and $OH(t+1)$ are applied on the environment that is a user-desired environment through operating the heater.

Decision-making based on intelligent-task offloading

The proposed energy optimization approach is provided through interacting with multiple intelligent services based on the decision-making model in the edge gateway. The purpose of energy optimization reduces the energy consumption of IoT device operation and update the environment to be the user-desired environment.

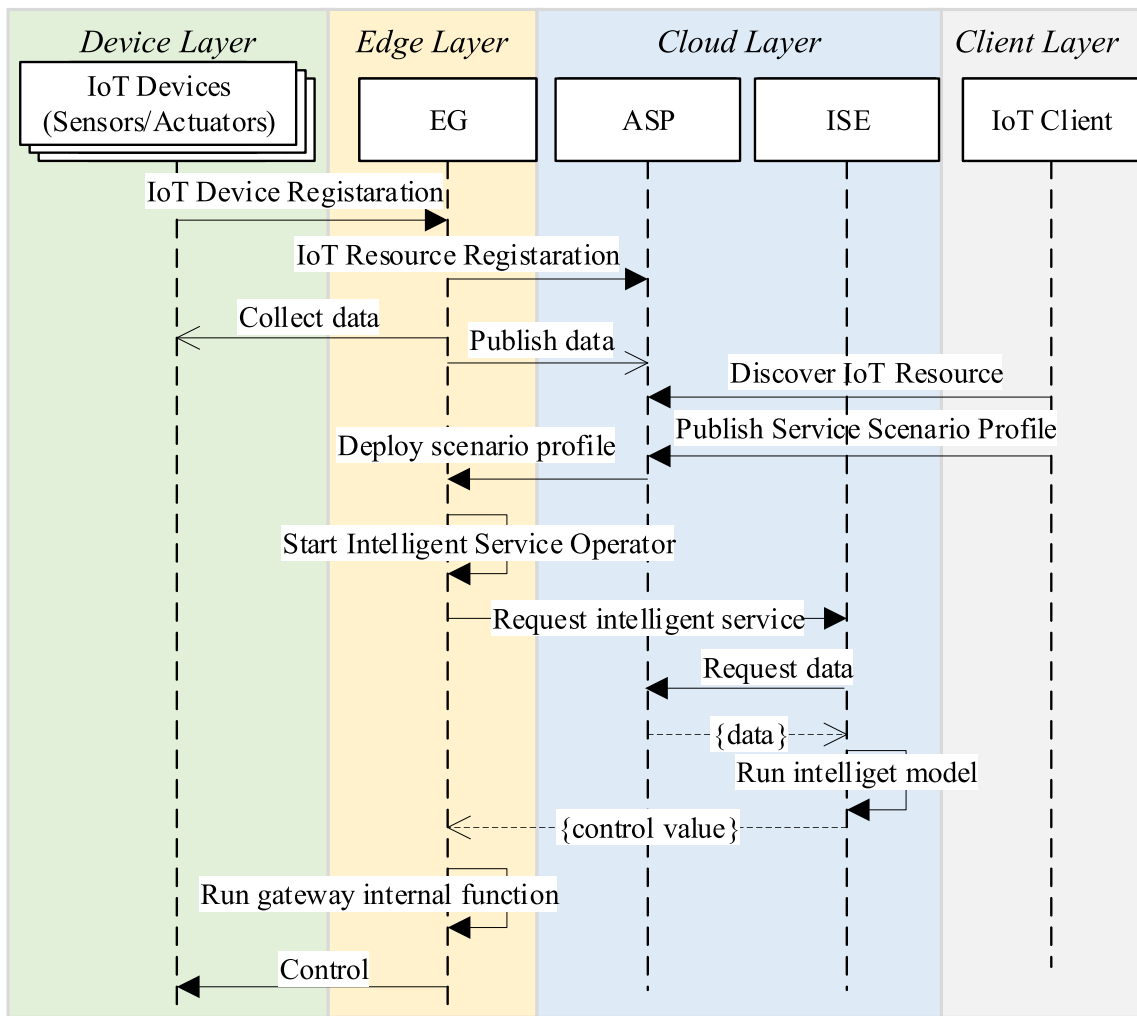


Fig. 3 IoT device operating process based on intelligent services

For applying intelligent services in the proposed gateway-centric IoT system, an RNN-based environment optimization (REO) model and PMV-based comfort optimization (PCO) model are developed and deployed in the ISE to provide the energy consumption. The energy consumption value is used for operating a heater in an indoor environment to update indoor temperature and humidity to achieve a user requirement. The power value selector (PVS) model is the decision-making model that is deployed in the EG to select a smaller energy consumption value to apply to the heater.

Figure 6 presents the proposed decision-making approach based on invoking intelligent services from the ISE. The intelligent services are provided through offloading the intelligent tasks to the ISE. The tasks are processed by models that are REO and PCO in the proposed IoT system. The models are derived from RNN models that are trained with the user data for inferencing a heater energy consumption. The heater energy consumption is used for updating the indoor environment to be a user-desired environment. The EG includes the

PVS model to make the decision for the selection of heater energy consumption. The EG invokes the results from the ISE, and selects a smaller value to apply on operating the heater. Therefore, the heater can consume a smaller energy consumption to make a user-desired environment based on the inference model that is trained by the user data.

Figure 7 depicts the REO-based environment management that is adopted in the proposed ISE to provide energy consumption. The energy consumption is the heater’s power value (HP) that is derived from the power prediction model. The model is an inference model that is derived from an RNN model. The prediction model outputs the HP for the future and the future HP is used in the heater model that is an emulator of the environment to update the environmental parameters. Then, the power prediction model outputs a new HP for the next future using the updated environmental parameters. The data of parameters include 96 time-sequences that are collected for 1 day.

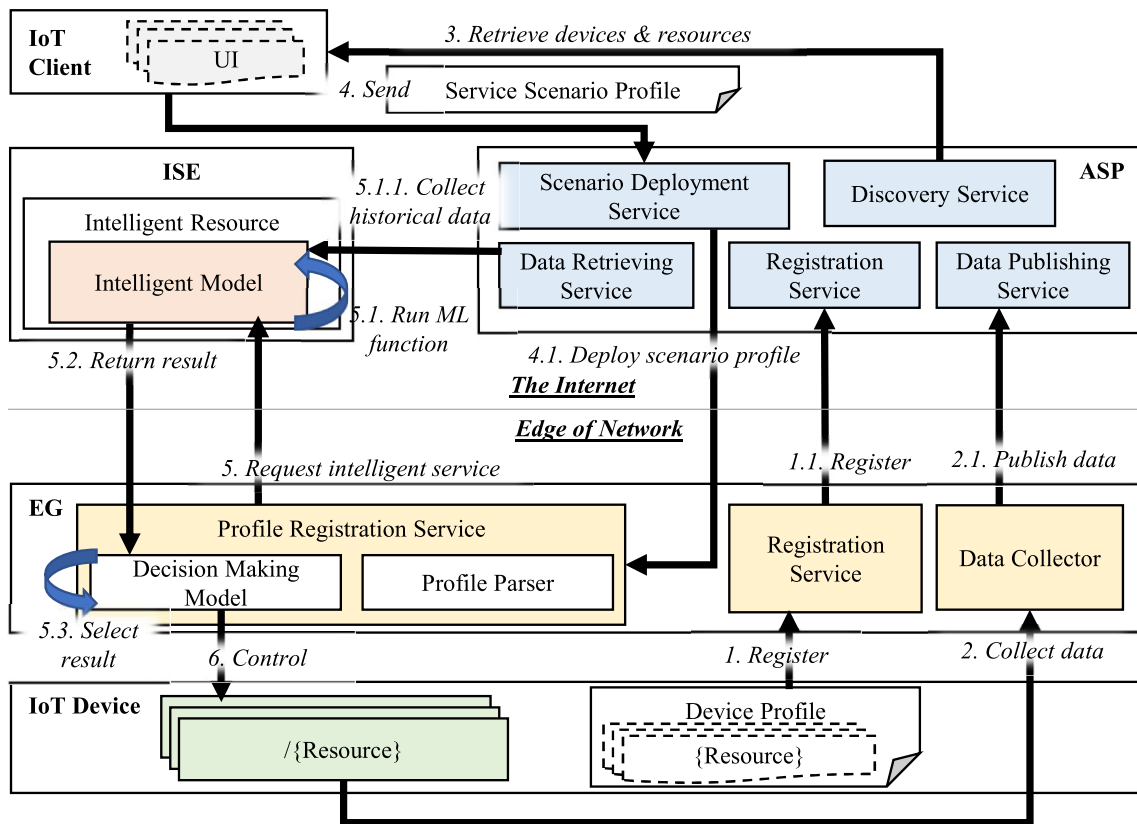


Fig. 4 Data pipeline of intelligent edge computing scenario

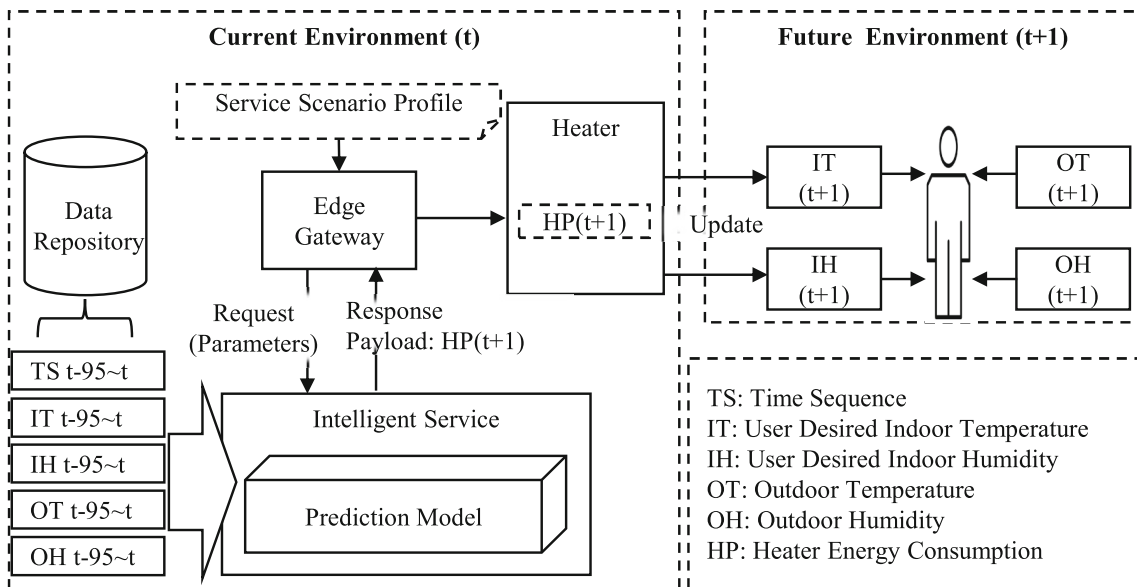


Fig. 5 User environment updating scenario

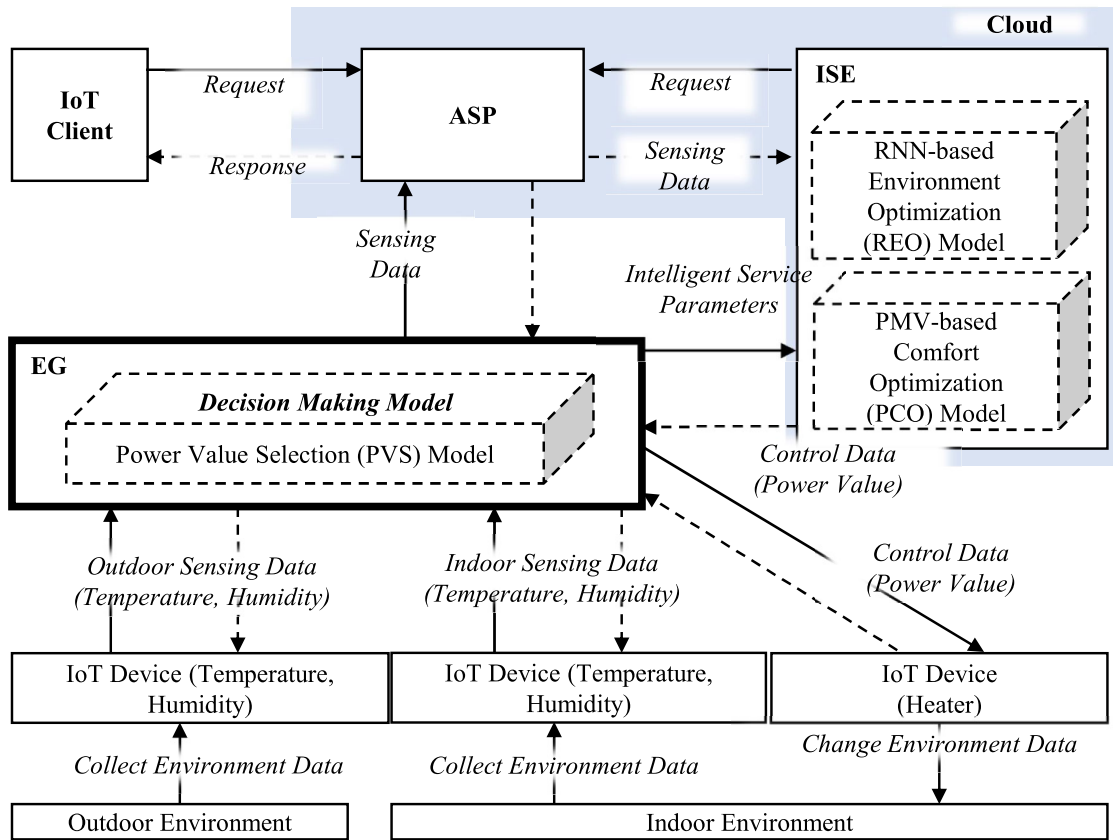


Fig. 6 Proposed decision-making based on intelligent-task offloading using edge gateway

Fig. 7 REO-based environment management architecture

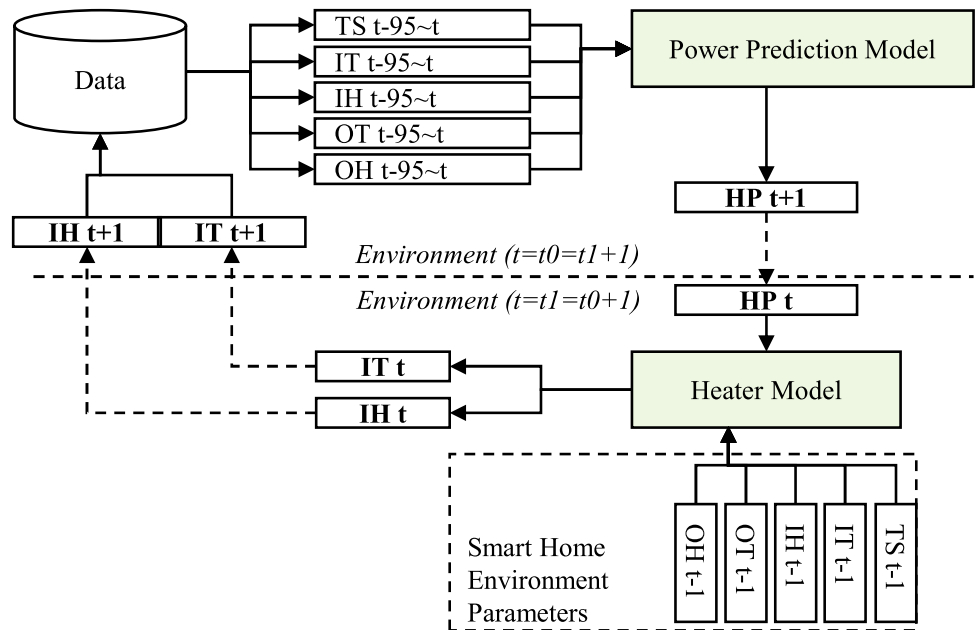


Fig. 8 PCO-based environment management architecture

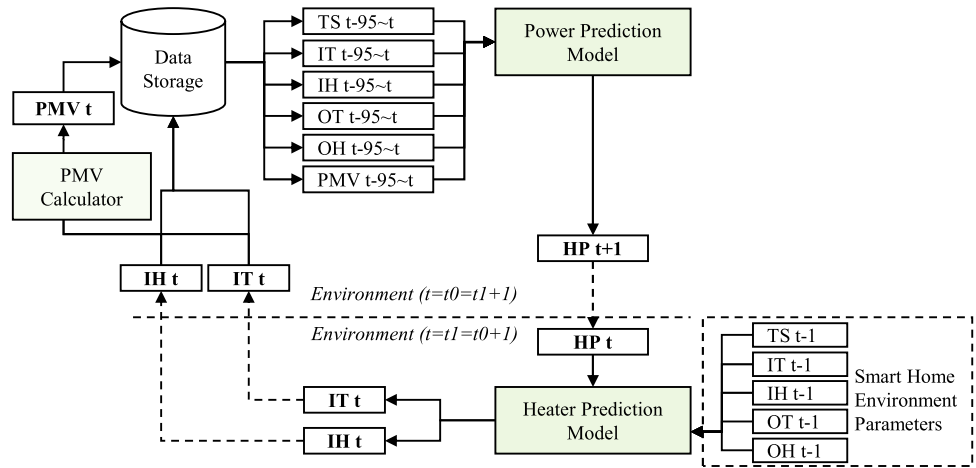


Fig. 9 RNN model for intelligent services

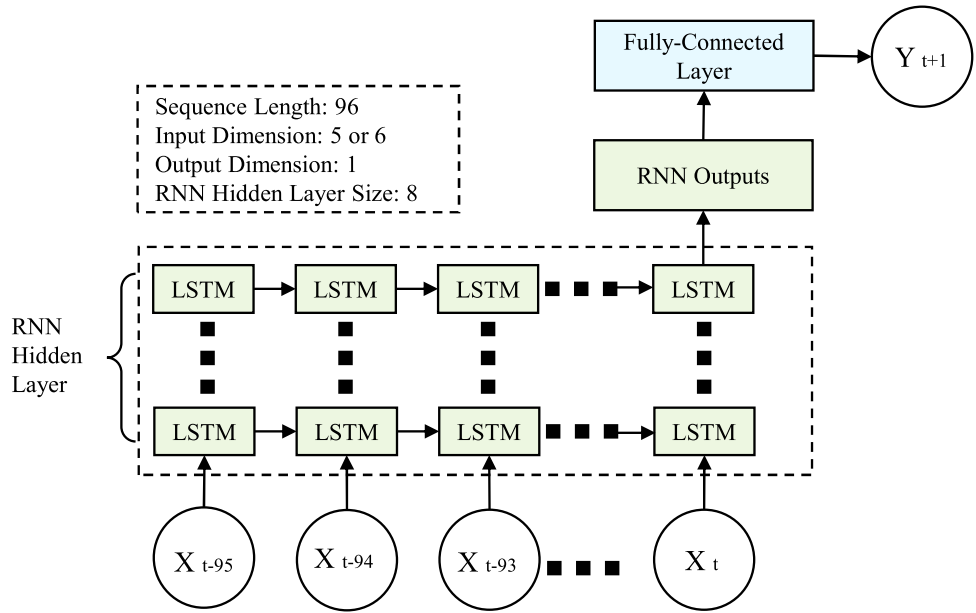


Fig. 10 Environment management architecture using REO and PCO model based on PVS model

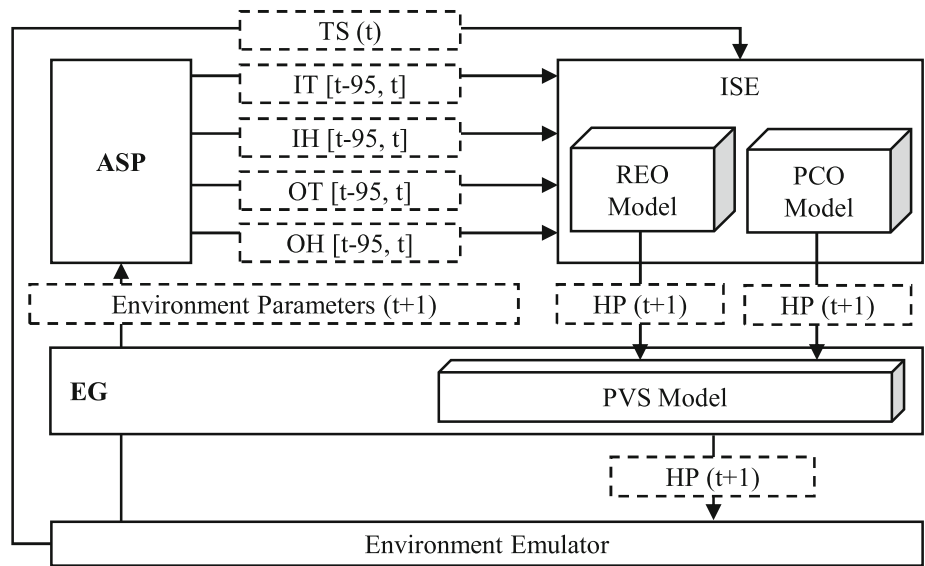


Figure 8 depicts the PCO-based environment management that is also adopted in the proposed ISE. The diffidence with the REO is the PMV parameter is included in the inference model. The PMV data for 96 time-sequences are provided by the PMV calculator. The PMV parameter is included in the power prediction model for providing an enhanced user-desired environment to express the user's wish.

The REO and RCO model is implemented by the RNN model with the user data. An RNN model is deep learning that derives a prediction model through training the leaning using a dataset. In the training processing, the weights of the entry to the nodes of the network are adjusted. Using the derived prediction model, heater energy consumption is provided for operating the heater to update the indoor environment. As shown in Fig. 9, the proposed RNN model is comprised of eight hidden layers based on LSTM cells. The sequence length is 96 that is designed for receiving 1-day data. Each input has multiple dimension which is implemented for REO is 5-dimensions and RCO is 6-dimensions. The output is the energy consumption value for operating a heater. Therefore, the output dimension is 1 that is provided by a fully connected layer. The t is presented for values of the environmental parameters in the time-sequence. Therefore, the output Y is a predicted value in $t+1$ time. The input X includes 96 values that are in time-sequence from $t-96$ to t .

Figure 10 depicts the proposed environmental management that is comprised of PVS, REO, and PCO to provide optimal heater energy consumption. In the overall system, the ASP and ISE are developed in the cloud, and EG and environment emulator are deployed in the network edge. The environment emulator includes IoT device and environment that are demonstrated through the emulator. From the emulator, the sensing values are collected to the ASP that provides the data to the ISE through the EG. In the ISE, the REO and PCO provide the heater power value to the PVS for making the decision to apply on the environment for operating the heater.

The approach of the decision-making in the PVS model is presented by Eqs. (1) and (2)

$$M = \{m_1, m_2, m_3, \dots, m_n\} \quad (1)$$

$$r = \min(M), \quad (2)$$

where the M is a set of models that are deployed in ISE. The set M can include more than two model for comparing and making decision to apply on the environmental control. In the proposed IoT architecture, we include two models that are REO and PCO for predicting the energy consumption. For selecting the minimal energy consumption, the results of the models in the set M are considered to control the environment.

Implementation details and results

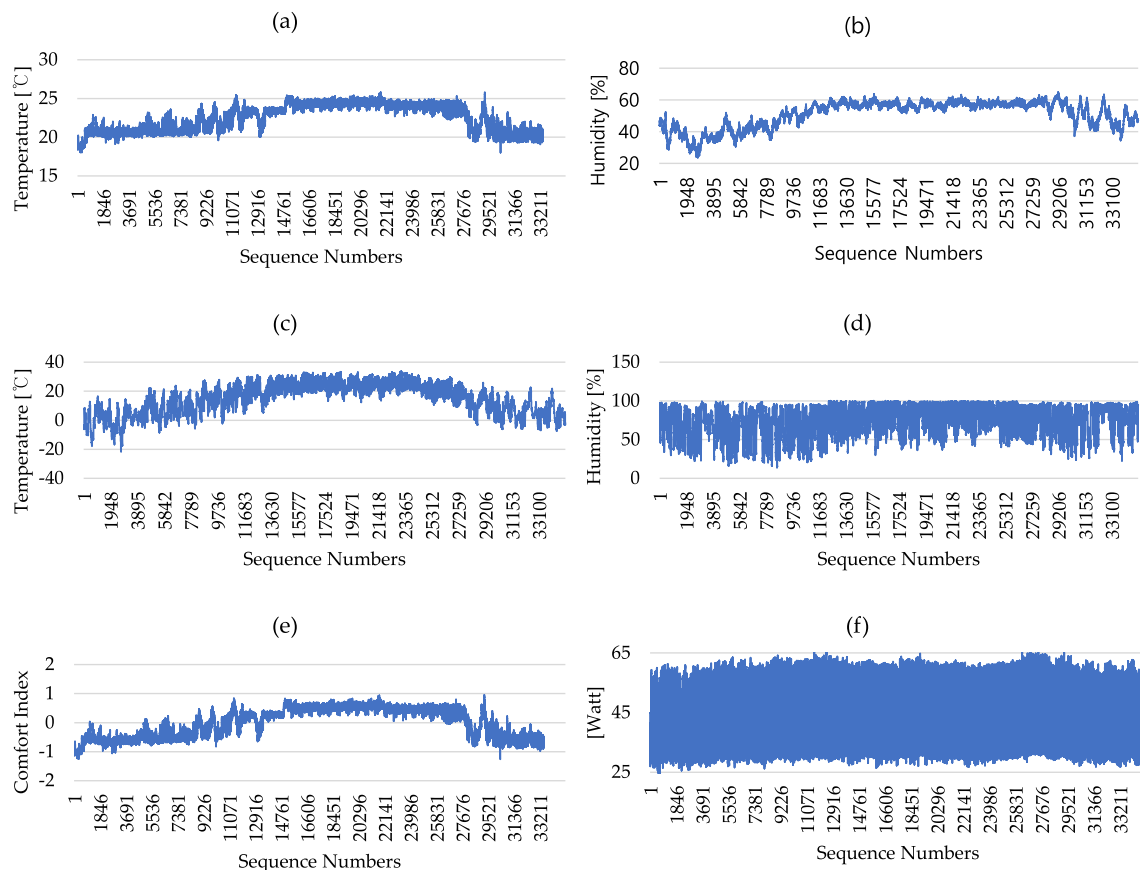
For implementing the proposed intelligent edge computing, various frameworks and libraries are included to develop each element. Several frameworks and platforms are published for building the IoT and edge computing to provide services, process and store data, manage information, and secure privacy as well as data. OCF is mainly used for implementing the IoT management and communication in the proposed system, as presented in Table 1. The implementation framework of OCF is IoTivity that is included EG and IoT devices to enable device registration and delivering the commands. We implement ASP and ISE on the high-performance computer with Windows 10 Pro 64 bit operating system. The server application of the ASP is implemented based on Spring Framework 4.3.3. MyBatis 3.4.2 is used for managing the MySQL database in the ASP. The libraries Commons File Upload 1.3.1, Apache HTTP Client 4.5.3, Fasterxml JSON 2.8.6, and ORG JSON 20160810 are used for implementing the functions in the ASP. The libraries JQuery 3.1.1, Bootstrap v3.3.7 are used for implementing the client functions and interface which are displayed in the web client. In the development of ISE, the Spring Boot 2.1.0 is used for implementing the services, Apache HTTP Client 4.5.3 is used for requesting EGs. The library org.tensorflow:tensorflow:1.8.0 is used for operating the inference models REO and PCO in the server application. For developing the REO and PCO model with the user data, libraries of TensorFlow 1.10.0, Numpy 1.15.1 are adopted in the Python 3 environment. EG and IoT device is implemented for constrained environments based on Android platform with IoTivity. EG includes server application for providing services to IoT device and ASP through Jetty. The environment emulator also includes an ML solution to provide a virtual environment that is implemented based on a DNN with the user dataset.

Figure 11 shows environmental data for implementing inference models that are used on REO, PCO, and environment emulator. The models are derived from DNN and RNN models that are trained with the environmental data. The data are collected by Oak Ridge National Laboratory (ORNL) which is located in Campbell, USA [81]. The collected data are comprised of many columns that are pre-processed to keep the indoor and outdoor temperature and humidity and heater energy consumption. The heater energy consumption affects the indoor temperature and humidity in the room where the heater and sensors are deployed. The data are collected every 15 min from 00:00 on 1 October 2013 to 23:45 on 31 December 2013, from 00:00 on 1 January 2014 to 23:45 on 30 September 2014.

Figure 12 shows the implementation result including the details of each entity in the proposed IoT system. The deployed entities are IoT client, ASP, ISE, EG, and IoT

Table 1 Development environment

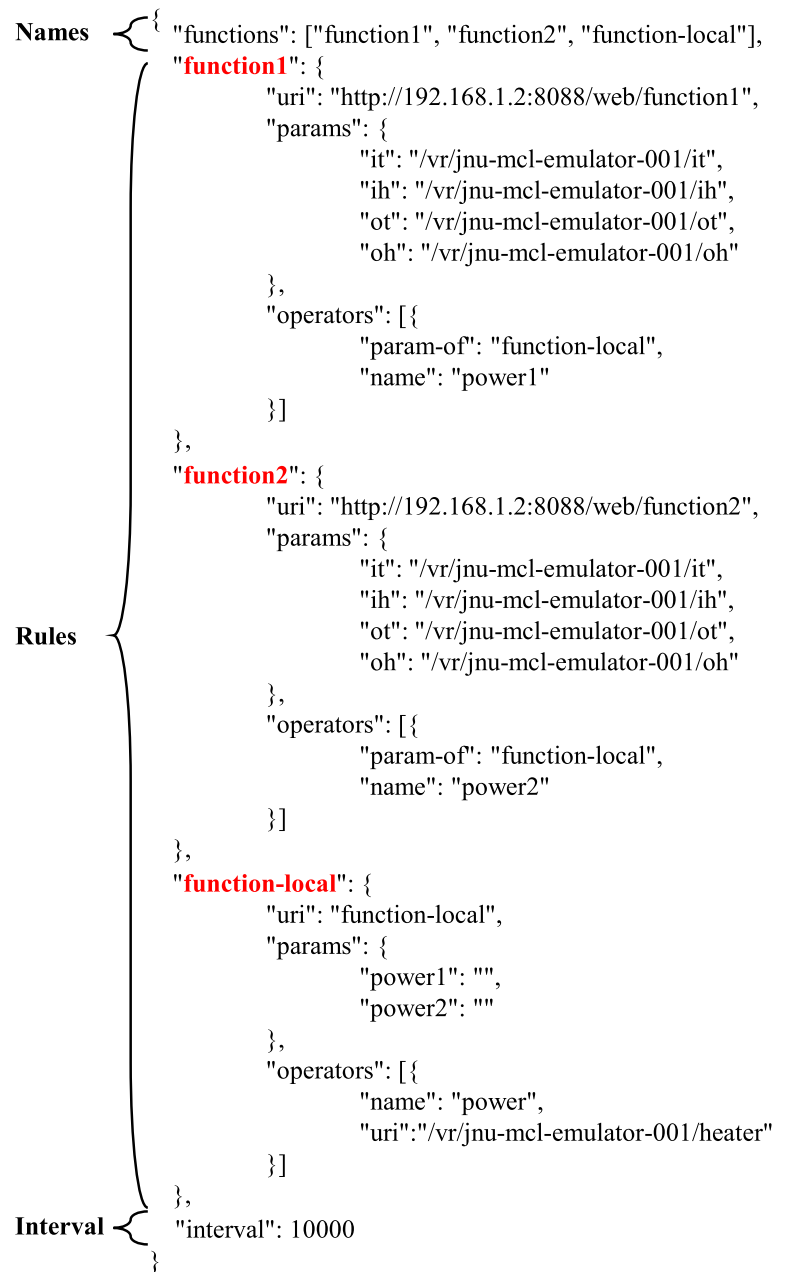
Components	Platform	Frameworks and Library
ASP	Windows 10 Pro 64 bit	Spring Framework 4.3.3, MyBatis 3.4.2, Commons File Upload 1.3.1, Apache HTTP Client 4.5.3, Fasterxml JSON 2.8.6, ORG JSON 20160810, JQuery 3.1.1, Bootstrap v3.3.7
ISE	Windows 10 Pro 64 bit	Spring Boot 2.1.0, Apache HTTP Client 4.5.3, org.tensorflow:tensorflow:1.8.0, TensorFlow 1.10.0, Numpy 1.15.1
EG	Android Things 1.0, SDK API Level 28	Jetty 9.1.0.v20131115, slf4j- 1.7.21, Volley 1.1.0, Fasterxml JSON 2.9.7, IoTivity 1.3.1
IoT device emulator	Android 7.0, SDK API Level 24	IoTivity 1.3.1, Volley 1.1.0, Fasterxml JSON 2.9.7
Environment emulator	Windows 10 Pro 64 bit	Spring Boot 2.1.0, org.tensorflow:tensorflow:1.8.0

**Fig. 11** Environmental data for experimental environment: **a** user-desired indoor temperature. **b** User-desired indoor humidity. **c** Outdoor temperature. **d** Outdoor humidity. **e** PMV based on user desired environment. **f** Heater energy consumption

device with the environment emulator. The ID of EG is `jnu.mcl.emulator.001` that is deployed in a local network with an IoT device. The IoT device collects the sensing data from the environment emulator that provides indoor/outdoor temperature and humidity data. The environment is updated by the heater that is based on a DNN. The IoT device provides services through OCF resources including resources of `/heater`, `/it`, `/ih`, `/ot`, `/oh` that are registered to the ASP through the EG. Through the registration, the virtual resources are generated in the EG that represents the actual resources to

the Internet. The ASP provides the services for accessing the registered resources by the IoT clients. Using the interfaces, users can retrieve the information of resources and sending commands for controlling actuators and collecting sensing data. In the ISE, the REO and PCO models are deployed to provide intelligent services. The intelligent services are exposed through the resources of `/web/function1` and `/web/function2` that are provided by the server application based on Spring Boot. The EG calls intelligent services

Fig. 13 Service scenario profile



Using the proposed REO and PCO models, the EG controls the heater to update the environment. In this process, the IoT device collects the environment data and sends the data to the ASP through the EG. Then, by the request from the EG, the ISE requests the data and operates an inference model. The model derives a controlling factor to the EG, and EG applies on the heater. For experimenting with the scenario, we apply 479 rows with REO, PCO, and PVS model on the proposed system to get results of updated IT, IH, and heater energy consumption data, as shown in Fig. 16. From the second row, only outdoor temperature and humidity data are applied with the updated indoor temperature and humid-

ity. Each result is compared with the original data to get the accuracies and differences for presenting the performance.

Performance evaluation

For providing improved energy optimization using decision-making based on the EG, we proposed REO and PCO models that are deployed in the ISE to provide intelligent services. The PVS model is deployed in the EG to select a smaller value for operating the heater. The REO and PCO models are derived by the RNN with the user-desired data that includes user-desired environmental data and heater energy consump-

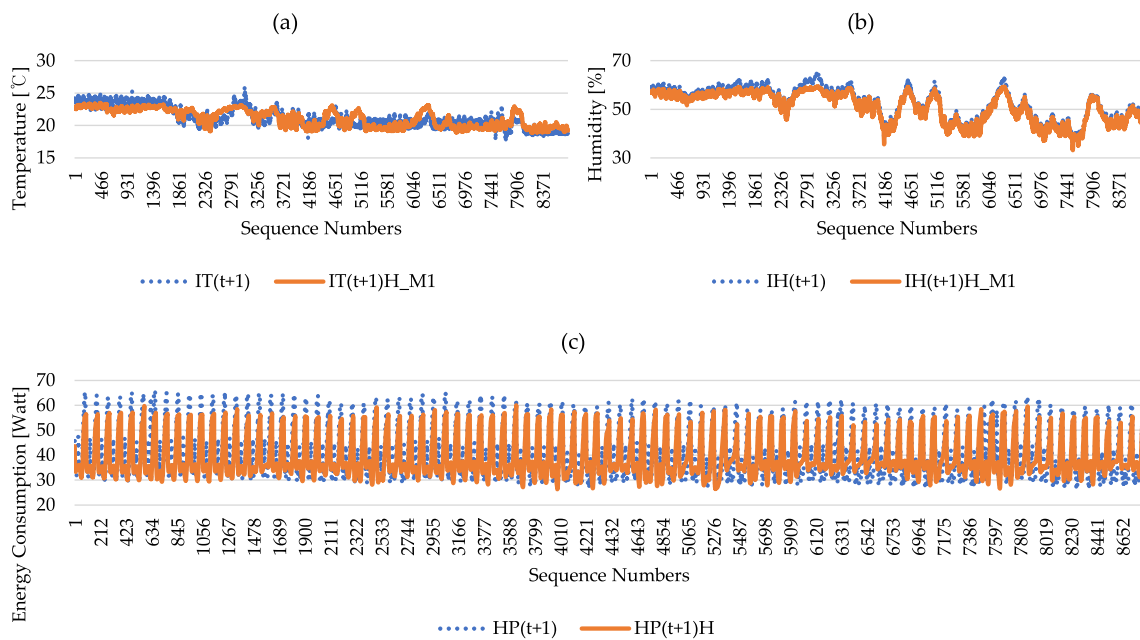


Fig. 14 Prediction results of REO model. **a** User-desired indoor temperature. **b** User-desired indoor humidity. **c** Heater energy consumption for user-desired indoor environment

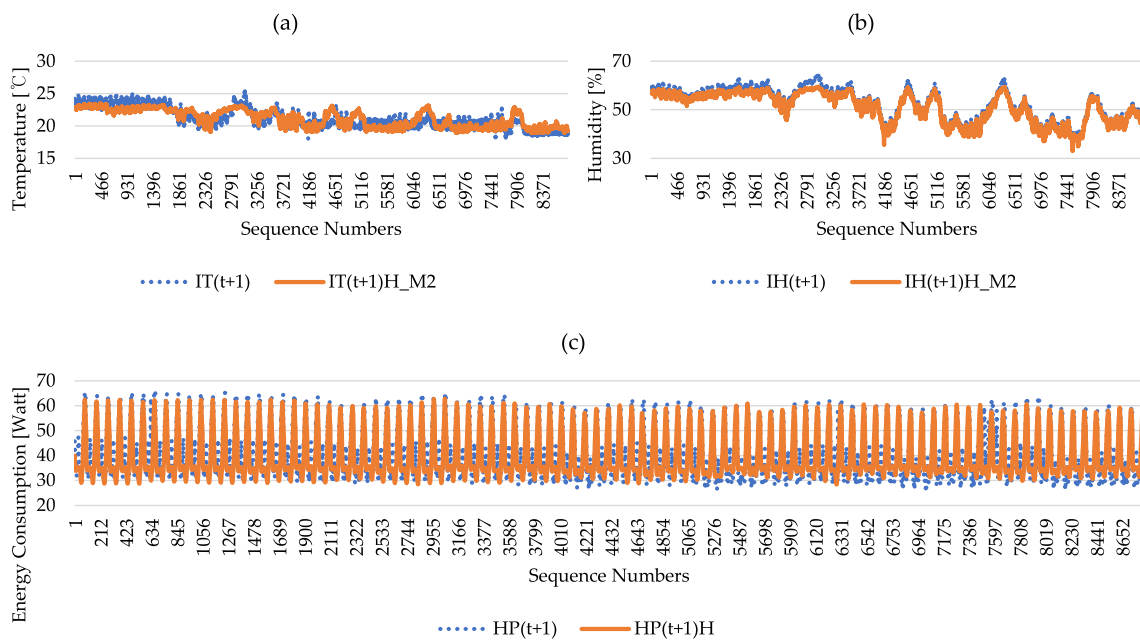


Fig. 15 Prediction results of PCO model. **a** User-desired indoor temperature. **b** User-desired indoor humidity. **c** Heater energy consumption for user-desired indoor environment

Table 2 Performance of prediction results between REO and PCO

Measure	ITHM1	ITHM2	IHHM1	IHHM2	HPHM1	HPHM2
MAPE	0.0392	0.0392	0.0314	0.0314	0.106	0.0856
MAD	0.8293	0.8297	1.65	1.6488	4.257	3.3106
MSE	0.9569	0.9587	3.107	3.104	29.6308	20.1247
RMSE	0.9782	0.9791	1.7627	1.7618	5.4434	4.4861

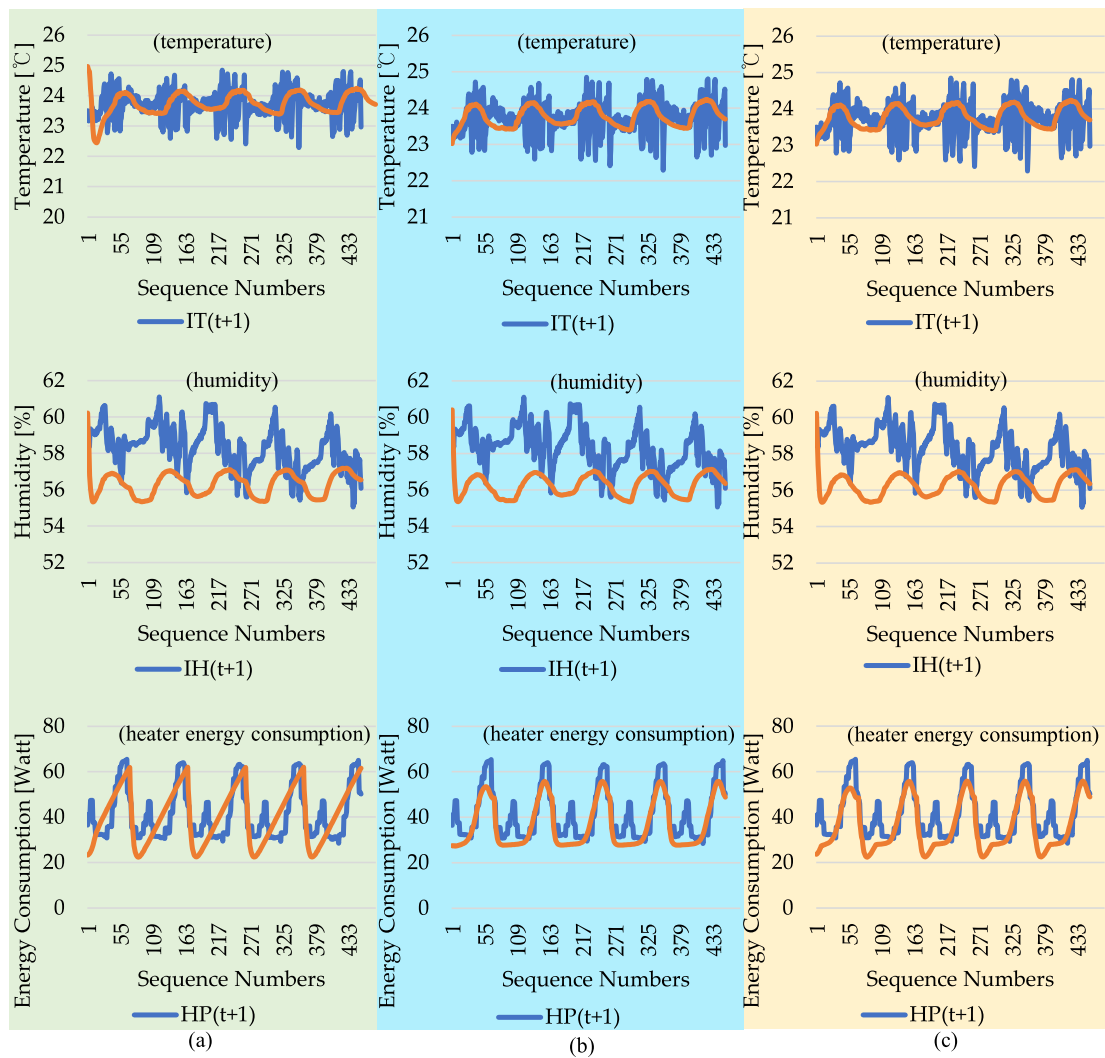


Fig. 16 Experiment results: **a** Only REO model. **b** Only PCO model. **c** PVS model based on REO and PCO models

Table 3 Performance of experiment results

Measure	ITEM1	ITEM2	ITEM	IHEM1	IHEM2	IHEM	HPEM1	HPEM2	HPEM
MAPE	0.0171	0.0152	0.0153	0.0355	0.0363	0.0365	0.2129	0.1328	0.1546
MAD	0.4037	0.3586	0.3593	2.0904	2.1344	2.1471	8.1481	5.5803	6.3183
MSE	0.2934	0.2398	0.2402	5.8864	6.0854	6.1501	84.7111	51.9885	63.6227
RMSE	0.5417	0.4897	0.4901	2.4262	2.4669	2.4799	9.2039	7.2103	7.9764

tion. The heater energy consumption is used for updating the environment to be a user-desired environment. The REO and PCO models are used for predicting the energy consumption for the user-desired environment. Table 2 presents the performance of the REO and PCO models. In the table, ITHM stands for indoor temperature, IHMH stands for indoor humidity, HPHM stands for heater energy consumption, the number 1 stands for the REO model, and 2 stands for the PCO model. The performance is presented through the

accuracy of updated indoor temperature and humidity, and heater energy consumption. The accuracy is calculated with the original data based on Mean Absolute Percentage Error (MAPE), Mean Absolute Deviation (MAD), Mean Squared Error (MSE), and Root Mean Square Error (RMSE). The results illustrate that the prediction performance of REO and PCO is sufficient to inference the heater energy consumption to make a user-desired environment.

Fig. 17 Performance comparison of experiment results with original heater energy consumption

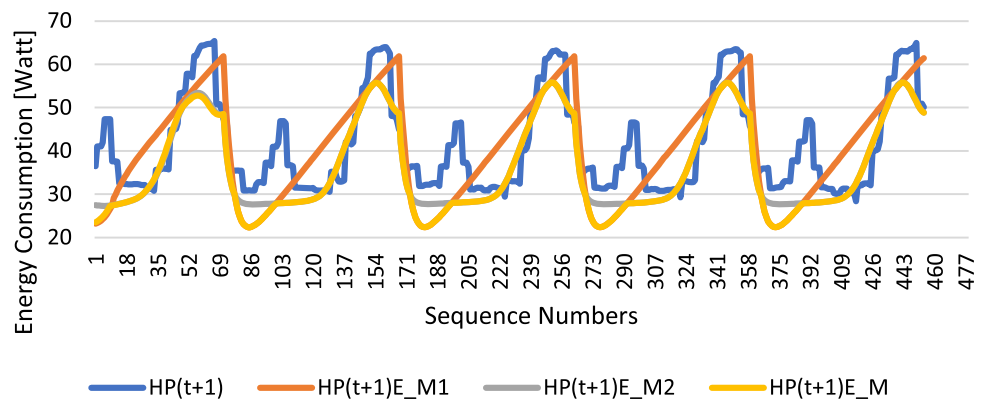
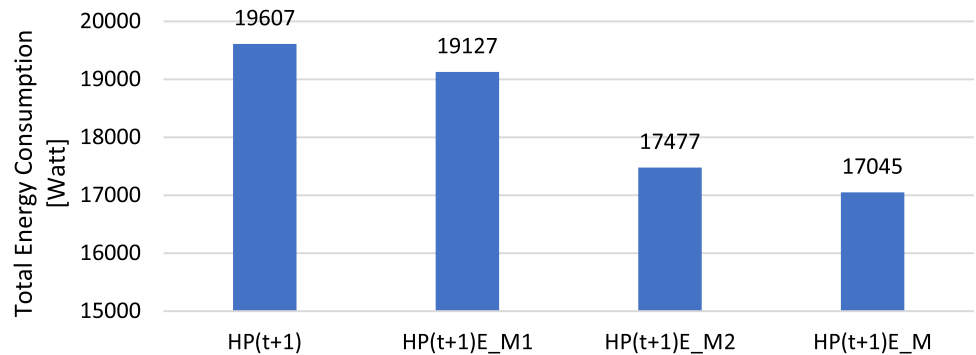


Fig. 18 Total energy consumption comparison of experimental results with original total energy consumption



Then, we apply REO and PCO on the ISE to collect the results 479 times separated through operating the IoT device using the EG. Also, the PVS model is applied to collect the optimized energy consumption with indoor temperature and humidity. However, compare with the original data through MAPE, MAD, MSE, and RMSE, the results are insufficient than the independent operation of models, as shown in Table 3. In the table, ITEM stands for indoor temperature, IHM stands for indoor humidity, HPEM stands for heater energy consumption, and extended number 1 stands for REO-model-only, 2 stands for PCO-model-only, and no-extended number stands for PVS model based on the REO and PCO models.

Figure 17 shows the performance comparison of experimental results with heater energy consumption. For this analysis, 479 values are collected through operating the heater using REO, PCO, and PVS based on REO and PCO models. For adopting the REO model, the result is presented by HP(t+1)E_M1, the result of the PCO model is presented by HP(t+1)E_M2, and the result of PVS model is presented by HP(t+1)E_M. The HP(t+1) is the original data that are similar to compare with others. Therefore, in the time-sequence, the heater energy consumption is enabled to affect the heater to update the environment to be a user-desired environment. Although, at some moment, the consumption is predicted less than the original. For this issue, we can set a threshold to increase the heater energy consumption.

The proposed improved energy optimization is enabled by selecting a smaller energy consumption from multiple intel-

ligent solutions that provide the heater energy consumption for making a user-desired environment. As shown in Fig. 18, the comparison is presented for the total energy consumption of original data (HP(t+1)), REO (HP(t+1)E_M1), PCO (HP(t+1)E_M2), and PVS(HP(t+1)E_M). As we declared, the predicted energy consumption is smaller than the original data in some moments. The total value is presented, the original data are highest, and the optimized result is the smallest. According to the proposed decision-making approach, the result must be the smallest because of the selection mechanism. The MAPE is 15.46% for the comparison between the result and original data for the time with the same initial environmental data and outdoor data. Nevertheless, the updated indoor temperature and humidity data are similar to the original data using the reduced energy consumption.

The calculation of the total energy consumption using each model is presented in Eq. (3)

$$E = \sum_{i=1}^n m (TS, IT, IH, OT, OH), \tag{3}$$

where the m is a model that is used in the experiment. The model requires the parameters TS, IT, IH, OT, and OH to predict the energy consumption for a time. The total consumption is sum of the all times for the predicted energy consumption. In this experiment, the total count of the prediction is 479 which is also the value of n.

Conclusions and future directions

We proposed an IoT system based on the EG to enable intelligent and autonomous operation of IoT devices. For providing improved energy optimization using the decision-making approach in the EG, inference models are developed based on a deep learning approach to provide intelligent services through the ISE. Then, the EG can call intelligent services to make a decision for selecting an optimal factor to apply on the IoT device. In the experiment, the optimal factor is the heater energy consumption that is used for operating the heater to update the environmental parameters such as indoor temperature and humidity. For providing the intelligent services to the EG, the REO and PCO models are deployed in the ISE. REO and PCO models are derived by learning a user's residential environment data through the RNN model. In this training process, the inputs are the user-desired environment data, and the output is the heater energy consumption that is used for operating the heater to affect the environment. Furthermore, the PVS model is deployed in the EG to make the decision for selecting a smaller value from the REO and PCO to operate the heater. Therefore, the proposed approach enables reducing energy consumption as well as providing a user-desired environment.

As future directions, we will offload more deep learning approaches to the ISE for providing intelligent services from various perspectives. Also, the decision-making approach of the EG needs to consider various elements to derive an optimal factor. Moreover, multiple control factors can be provided through complex consideration based on the decision-making approach.

Acknowledgements This work was supported by Institute of Information and communications Technology Planning and Evaluation (IITP) grant funded by the Korea government (MSIT) (2020-0-00048, Development of 5G-IoT Trustworthy AI-Data Commons Framework). Any correspondence related to this paper should be addressed to Dohyeon Kim.

Declarations

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Stergiou C, Psannis KE (2017) Recent advances delivered by mobile cloud computing and internet of things for big data applications: a survey. *Int J Netw Manag* 27(3):e1930
2. Li X, Li D, Wan J, Liu C, Imran M (2018) Adaptive transmission optimization in sdn-based industrial internet of things with edge computing. *IEEE Internet Things J* 5(3):1351–1360
3. Rathore P, Rao AS, Rajasegarar S, Vanz E, Gubbi J, Palaniswami M (2017) Real-time urban microclimate analysis using internet of things. *IEEE Internet Things J* 5(2):500–511
4. Hu J, Yang K, Marin ST, Sharif H (2018) Guest editorial special issue on internet-of-things for smart cities. *IEEE Internet Things J* 5(2):468–472
5. Montori F, Bedogni L, Bononi L (2017) A collaborative internet of things architecture for smart cities and environmental monitoring. *IEEE Internet Things J* 5(2):592–605
6. Perera C, Liu CH, Jayawardena S, Chen M (2014) A survey on internet of things from industrial market perspective. *IEEE Access* 2:1660–1679
7. Javed F, Afzal MK, Sharif M, Kim B-S (2018) Internet of things (iot) operating systems support, networking technologies, applications, and challenges: a comparative review. *IEEE Commun Surveys Tutorials* 20(3):2062–2100
8. Aftab H, Gilani K, Lee J, Nkenyereye L, Jeong S, Song J (2019) Analysis of identifiers on iot platforms. *Digital Commun Netw*
9. Shafiq S. I, Sanin C, Szczerbicki E, Toro C (2015) Virtual engineering object/virtual engineering process: a specialized form of cyber physical system for industrie 4.0. *Proc Comput Sci* 60:1146–1155
10. Schneider GF, Wicaksono H, Ovtcharova J (2019) Virtual engineering of cyber-physical automation systems: the case of control logic. *Adv Eng Inform* 39:127–143
11. Want R, Schilit BN, Jenson S (2015) Enabling the internet of things. *Computer* 48(1):28–35
12. Jin W, Kim D (2018) A sleep scheme based on mq broker using subscribe/publish in iot network. *Int J Adv Sci. Eng Inf Technol* 8:539–545
13. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M (2015) Internet of things: a survey on enabling technologies, protocols, and applications. *IEEE Commun Surveys Tutorials* 17(4):2347–2376
14. Botta A, Donato W, Persico A, Pescapé A (2016) Integration of cloud computing and internet of things: a survey. *Futur Gener Comput Syst* 56:684–700
15. Jiang L, Da Xu L, Cai H, Jiang Z, Bu F, Xu B (2014) An iot-oriented data storage framework in cloud computing platform. *IEEE Trans Industr Inf* 10(2):1443–1451
16. Jin W, Kim DH (2018) Design and implementation of e-health system based on semantic sensor network using ietf yang. *Sensors* 18(2):629
17. Charyyev B, Arslan E, Gunes M (2020) Latency comparison of cloud datacenters and edge servers. In: *GLOBECOM 2020-2020 IEEE global communications conference*, pp 1–6, IEEE
18. Didic A, Nikolaidis P (2015) Real-time control in industrial iot
19. Yousefpour A, Fung C, Nguyen T, Kadiyala K, Jalali F, Niakanlahiji A, Kong J, Jue JP (2019) All one needs to know about fog computing and related edge computing paradigms: a complete survey. *J Syst Arch*
20. Khan WZ, Ahmed E, Hakak S, Yaqoob I, Ahmed A (2019) Edge computing: a survey. *Futur Gener Comput Syst* 97:219–235
21. Liu C-F, Bennis M, Debbah M, Poor HV (2019) Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing. *IEEE Trans Commun* 67(6):4132–4150

22. Jin W, Kim D (2018) Development of virtual resource based iot proxy for bridging heterogeneous web services in iot networks. *Sensors* 18(6):1721
23. Jin W, Kim DH (2017) Iot device management architecture based on proxy. In: 2017 6th international conference on computer science and network technology (ICCSNT), pp 84–87, IEEE
24. Wang C, Yu FR, Liang C, Chen Q, Tang L (2017) Joint computation offloading and interference management in wireless cellular networks with mobile edge computing. *IEEE Trans Veh Technol* 66(8):7432–7445
25. Li C, Xue Y, Wang J, Zhang W, Li T (2018) Edge-oriented computing paradigms: a survey on architecture design and system management. *ACM Comput Survays (CSUR)* 51(2):1–34
26. Chen C-H, Lin M-Y, Liu C-C (2018) Edge computing gateway of the industrial internet of things using multiple collaborative micro-controllers. *IEEE Netw* 32(1):24–32
27. Morabito R, Petrolo R, Loscrì V, Mitton N (2019) Reprint of: Legiot: a lightweight edge gateway for the internet of things. *Futur Gener Comput Syst* 92:1157–1171
28. Aghdai A, Huang M, Dai D, Xu Y, Chao J (2018) Transparent edge gateway for mobile networks. In: 2018 IEEE 26th international conference on network protocols (ICNP), IEEE, pp 412–417
29. Wang X, Leng S, Yang K (2017) Social-aware edge caching in fog radioaccess networks. *IEEE Access* 5:8492–8501
30. Chen X, Shi Q, Yang L, Xu J (2018) Thriftyedge: Resource-efficient edgecomputing for intelligent iot applications. *IEEE Netw* 32(1):61–65
31. Zhang K, Zhu Y, Leng S, He Y, Maharjan S, Zhang Y (2019) Deeplearning empowered task offloading for mobile edge computing in urbaninformatics. *IEEE Internet Things J* 6(5):7635–7647
32. Hao Y, Jiang Y, Chen T, Cao D, Chen M (2019) itaskoffloading: Intelligenttask offloading for a cloud-edge collaborative system. *IEEE Netw* 33(5):82–88
33. Marquez G, Johnson B, Jafari M, Gomez M (2019) Online machinelearning based predictor for biological systems. In: 2019 IEEE symposiumseries on computational intelligence (SSCI), pp 120–125, IEEE
34. Hoi SC, Wang J, Zhao P (2014) Libol: A library for online learningalgorithms. *J Mach Learn Res* 15(1):495–499
35. Jin W, Hong Y-G, Kim D-H (2018) Design and implementation of a wireless IoT Healthcare system based on OCF IoTivity. *Int. J. Grid Distrib Comput* 11(4):87–96
36. Jin W, Kim D (2019) Resource management based on OCF for device self-registration and status detection in IoT networks. *Electronics* 8(3):311
37. Jin W, Kim D (2020) Interworking proxy based on OCF for connecting web services and IoT networks. *J Commun* 15(2)
38. Cha H, Choi Y, Lee K (2019) Facilitating healthcare IoT standardization with open source: a case study on OCF and IoTivity. In: 2019 ITU Kaleidoscope: ICT for health: networks, standards and innovation (ITU K), pp 1–9, IEEE
39. Elsayed K, Ibrahim MAB, Hamza HS (2019) Service discovery in heterogeneous IoT environments based on OCF/IoTivity. In: 019 15th international wireless communications and mobile computing conference (IWCMC), pp 1160–1165 IEEE
40. Park S (2017) OCF: a new open IoT consortium. In: 2017 31st international conference on advanced information networking and applications workshops (WAINA), pp 356–359, IEEE
41. Datta, SK, Da Costa RPF, Bonnet C, Härri J (2016) oneM2M architecture based IoT framework for mobile crowd sensing in smart cities. In: 2016 European conference on networks and communications (EuCNC), pp 168–173, IEEE
42. Yun J, Ahn I-Y, Song JS, Kim J (2019) Implementation of sensing and actuation capabilities for IoT devices using oneM2M platforms. *Sensors* 19(20):4567
43. Chang WG, Lin FJ (2016) Challenges of incorporating OMA LWM2M gateway in M2M standard architecture. In: 2016 IEEE conference on standards for communications and networking (CSCN), pp 1–6, IEEE
44. Satyanarayanan M, Chen Z, Ha K, Hu W, Richter W, Pillai P (2014) Cloudlets: at the leading edge of mobile-cloud convergence. In: 6th international conference on mobile computing, applications and services, pp 1–9, IEEE
45. Abbas N, Zhang Y, Taherkordi A, Skeie T (2017) Mobile edge computing: a survey. *IEEE Internet Things J* 5(1):450–465
46. Mach P, Becvar Z (2017) Mobile edge computing: a survey on architecture and computation offloading. *IEEE Commun Survays Tutorials* 19(3):1628–1656
47. Sun X, Ansari N (2016) EdgeIoT: mobile edge computing for the Internet of Things. *IEEE Commun Mag* 54(12):22–29
48. Dolui K, Datta SK (2017) Comparison of edge computing implementations: fog computing, cloudlet and mobile edge computing. In: 2017 global internet of things summit (GIoTS), pp 1–6. IEEE
49. Kim JB, Kwon DH, Hong YG, Lim HK, Kim MS, Han YH (2019) Deep Q-network based rotary inverted pendulum system and its monitoring on the EdgeX platform. In: 2019 international conference on artificial intelligence in information and communication (ICAIIIC), pp 034–039, IEEE
50. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444
51. Yu S, Wang X, Langar R (2017) Computation offloading for mobile edge computing: a deep learning approach. In: 2017 IEEE 28th annual international symposium on personal, indoor, and mobile radio communications (PIMRC), IEEE
52. Zhang Y, Niyato D, Wang P (2015) Offloading in mobile cloudlet systems with intermittent connectivity. *IEEE Trans Mob Comput* 14(12):2516–2529
53. Qiao G et al (2018) Collaborative task offloading in vehicular edge multi-access networks. *IEEE Commun Mag* 56(8):48–54
54. Zhang K, Zhu Y, Leng S, He Y, Maharjan S, Zhang Y (2019) Deep learning empowered task offloading for mobile edge computing in urban informatics. *IEEE Internet Things J* 6(5):7635–7647
55. Crutcher A, Koch C, Coleman K, Patman J, Esposito F, Callyam P (2017) Hyperprofile-based computation offloading for mobile edge networks. In: 2017 IEEE 14th international conference on mobile ad hoc and sensor systems (MASS), pp 525–529, IEEE
56. Xu J, Chen L, Ren S (2017) Online learning for offloading and autoscaling in energy harvesting mobile edge computing. *IEEE Trans Cognit Commun Netw* 3(3):361–373
57. Zhang R, Cheng P, Chen Z, Liu S, Li Y, Vucetic B (2020) Online learning enabled task offloading for vehicular edge computing. *IEEE Wirel Commun Lett* 9(7):928–932
58. Zhang F, Ge J, Wong C, Li C, Chen X, Zhang S, Luo B, Zhang H, Chang V (2019) Online learning offloading framework for heterogeneous mobile edge computing system. *J Parallel Distrib Comput* 128:167–183
59. Liu C, Cao Yu, Luo Y, Chen G, Vokkarane V, Yunsheng M, Chen S, Hou P (2017) A new deep learning-based food recognition system for dietary assessment on an edge computing service infrastructure. *IEEE Trans Serv Comput* 11(2):249–261
60. Zhang Q, Lin M, Yang LT, Chen Z, Li P (2017) Energy-efficient scheduling for real-time systems based on deep Q-learning model. *IEEE Trans Sustain Comput* 4(1):132–141
61. Alsheikh MA, Niyato D, Lin S, Tan HP, Han Z (2016) Mobile big data analytics using deep learning and apache spark. *IEEE Netw* 30(3):22–29
62. Biswas A, Chandrakasan AP (2018) Conv-RAM: an energy-efficient SRAM with embedded convolution computation for low-power CNN-based machine learning applications. In: 2018 IEEE international solid-state circuits conference-(ISSCC), pp 488–490, IEEE

63. Zhang W, Zhao D, Xu L, Li Z, Gong W, Zhou J (2016) Distributed embedded deep learning based real-time video processing. In: 2016 IEEE international conference on systems, man, and cybernetics (SMC), pp 001945–001950, IEEE
64. Blanco-Filgueira B, García-Lesta D, Fernández-Sanjurjo M, Brea VM, López P (2019) Deep learning-based multiple object visual tracking on embedded system for iot and mobile edge computing applications. *IEEE Internet Things J* 6(3):5423–5431
65. Zhang S, Fine JP, Touchie MF, O'Brien W (2020) A simulation framework for predicting occupant thermal sensation in perimeter zones of buildings considering direct solar radiation and ankle draft. *Build Environ* 183:107096
66. Molina-Solana M, Ros M, Ruiz MD, Gómez-Romero J, Martín-Bautista M (2017) Data science for building energy management: a review. *Renew Sustain Energy Rev* 70:598–609
67. Aste N, Manfren M, Marenzi G (2017) Building automation and control systems and performance optimization: a framework for analysis. *Renew Sustain Energy Rev* 75:313–330
68. Manic M, Wijayasekara D, Amarasinghe K, Rodriguez-Andina JJ (2016) Building energy management systems: the age of intelligent and adaptive buildings. *IEEE Ind Electron Mag* 10:25–39
69. Jabłoński K, Grychowski T (2017) Fuzzy inference system for the assessment of indoor environmental quality in a room. *Indoor Built Environ* 27:1415–1430
70. Jin W, Ullah I, Ahmad S, Kim D (2019) Occupant comfort management based on energy optimization using an environment prediction model in smart homes. *Sustainability* 11(4):997
71. ORNL Long Term Energy Environment. https://trynthink.github.io/buildingsdatasets/show.html?title_id=long-term-energy-environment-data-for-ornl-research-house-3. Accessed May 2020

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.