This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3308579

Hetero-FedIoT: A Rule-based Interworking Architecture for Heterogeneous Federated IoT Networks

Anam Nawaz Khan[1], Atif Rizwan[1], Rashid Ahmad[1,2,*], Wenquan Jin[3], Qazi Waqas Khan[1], Sunhwan Lim[4], DoHyeun Kim[1,5,*]

***Abstract*—The rapid growth of the Federated Internet of Things ecosystem has introduced new challenges in achieving seamless connectivity and interoperability across heterogeneous IoT networks. The presence of heterogeneous platforms and protocols creates significant obstacles for effective communication among cross-silos federated IoT nodes. To tackle this challenge, we have developed Heterogeneous Federated Internet of Things (Hetero-FedIoT), an innovative rule-based interworking architecture enabling interoperability and seamless connectivity among heterogeneous federated IoT networks (oneM2M, OCF and EdgeX). Hetero-FedIoT offers a two-faceted solution to address these challenges. Firstly, it incorporates a rule-based interworking mechanism that fosters effective collaboration among Hetero-FedIoT networks. Additionally, it introduces a novel aggregation function capable of achieving accelerated convergence, effectively handling both system and statistical heterogeneity. By leveraging device proxies, Hetero-FedIoT enables interoperability among heterogeneous FedIoT networks by translating protocols from platform-native formats to a common format and vice versa. As a result, collaborative model training can be seamlessly conducted without the need to consider underlying frameworks. Additionally, the novel aggregation algorithm employed by Hetero-FedIoT empowers nodes to customize the complexity of local models according to their communication and computation capabilities. This is accomplished through the dynamic adjustment of hidden channel widths, ensuring that the overall performance of the global model remains unaffected. This groundbreaking Hetero-FedIoT architecture establishes a foundation for enhanced interoperability and optimal performance. Extensive evaluation of Hetero-FedIoT has demonstrated superior computational and communication efficiency over baseline schemes. The Hetero-FedIoT system revolutionizes decentralized training under heterogeneous conditions, fostering widespread adoption.**

***Index Terms*—Federated Learning, Statistical Heterogeneity, System Heterogeneity, Interworking, Federated Internet of Things.**

I. Introduction

Department of Computer Engineering, Jeju National University, Jejusi 63243, Jeju Special Self-Governing Province, Republic of Korea (e-mail: anamnawaz@jejunu.ac.kr)

Department of Computer Science, COMSATS University Islamabad, Attock Campus 43600, Pakistan (e-mail: rashid.ahmad@ciit-attock.edu.pk)

Department of Electronic and Communication Engineering, Yanbian University, Yanji 133002, China (e-mail: jinwq@ybu.edu.cn)

Autonomous IoT Research Section, Intelligent Convergence Research Laboratory, Electronics and Telecommunications Research Institute, Daejeon, 34129, Korea (e-mail: shlim@etri.re.kr)

Department of Computer Engineering (and Advanced Technology Research Institute), Jeju National University, Jeju 63243, Republic of Korea (e-mail: kimdh@jejunu.ac.kr)

FEDIOT, short for Federated Internet of Things, represents the concept of decentralized collaboration among multiple IoT systems. This procedure involves consolidating and synchronizing diverse IoT networks and platforms to facilitate collaborative model training. Typically, in a centralized training scheme, the data generated by IoT devices is transmitted to a central server, either by physical servers or cloud clusters for further processing [1]. However due to the implementation of Global Data Privacy Regulations (GDPR), there has been a rise in the adoption of Federated Learning (FL) as an alternative approach. FL introduces a fresh paradigm where IoT devices collaborate to train a machine learning model, overseen by a central server [2], [3]. Each device independently trains the model using its own set of training data, and only the local parameter updates are shared with the server. By keeping the actual data on the devices, it ensures the preservation of privacy for the edge nodes. This iterative process continues until an optimal global model is achieved.

During federation, the edge nodes only send the local model parameters, or gradients, instead of data for model aggregation, which further reduces the communication overhead [4]. The Google keyboard serves as a prominent example, as it effectively utilize local data on the device to maintain contextual relevance and track user engagement with each potential recommendation displayed. The information obtained from users is utilized to augment the intelligence of forthcoming G-board recommendations [5]. However, the device usage pattern of users varies, which leads to statistical heterogeneity and imbalanced data and label distributions.

The current state of the IoT market exhibits substantial fragmentation, wherein a multitude of independently developed solutions primarily emphasize vertical optimization rather than horizontal integration [6]. This fragmented landscape has led to a lack of interoperability and synergy among solution deployments, hindering the realization of a unified and cohesive ecosystem. According to [7] interoperability is required to realize 40% of the potential benefits of IoT. Interoperability, in the context of FedIoT systems, refers to the seamless connectivity of multiple devices. However, the heterogeneous nature of these systems, encompassing diverse protocols, platforms, technologies, and hardware, presents a significant challenge in achieving smooth and efficient connectivity among them [8]. This gap can be attributed to the adoption of different technological standards and protocols among edge networks. Recently, several solutions have been devised for achieving interoperability between standard frameworks for IoT platforms. The solutions presented are primarily centered on bridging techniques that make use of proxies, gateways, and middle-ware. However, in the context of data sharing and transfer across platforms, it is imperative to take into account communication protocols and data formats [9]. The primary objective of standardization initiatives in the this realm is to promote consistent service provision by means of standardized interfaces. The implementation of specialized platforms is crucial in order to guarantee the necessary performance efficiency to satisfy the computational and communication limitations of edge nodes.

For instance, the Machine-to-Machine (oneM2M) standard is a globally recognized to facilitate interoperability among various IoT systems. The provision of a shared service layer facilitates the smooth transmission and exchange of data across disparate platforms and devices. In order to enable effective communication among diverse devices and platforms, the oneM2M standard plays a crucial role by establishing a comprehensive set of protocols and APIs. These standardized mechanisms facilitate seamless interoperability, ensuring transparent interaction between different entities involved in the system. Despite the standardization efforts undertaken by oneM2M to facilitate communication among heterogeneous IoT platforms, the integration of this platform with other systems may still encounter interoperability challenges arising from differences in implementation and protocol preferences [10], [11]. In this regard, The Open Connectivity Foundation (OCF) provided an open-source platform that places a significant emphasis on fostering interoperability and connectivity among edge devices. However, the integration of OCF-IoTivity with non-OCF platforms poses challenges attributed to inherent protocol variations. These obstacles impede the communication with other platforms, consequently affecting the realization of comprehensive interoperability within the broader IoT landscape [12].

Despite their connection to the same middleware, the disparities in protocols between platforms such as oneM2M, Watson-IoT by IBM, FIWARE, an open-source platform, and OCF create interoperability challenges in FedIoT systems. Since each platform uses its own set of protocols and communication mechanisms, which results in incompatible data formats and communication methods. As a result, devices and systems operating on different platforms struggle to effectively exchange information and collaborate. Several researchers have suggested the use of rule-based interoperability solutions. The rules serve as a contextual structure that encompasses information and enables the functioning of service scenarios through the assessment of occurrences [13].

For rule and protocol translation via proxies, it is important to ensure that the knowledge of rules is easily understood by the heterogeneous Fed-IoT systems. The current situation requires the development of a system to provide interworking support for heterogeneous protocols, device identification, well-defined semantic management, and processing of data formats on heterogeneous platforms [14]. To comprehend the information included within a rule context, the interworking solution must have an interpreter that effectively sifts through occurrences in line with the defined rule conditions [15]. As a result, establishing a uniform rule context and interface that can be easily applied across the FedIoT platforms is critical. In order to comprehend the data encompassed by a set of rules, the utilization of an interpreter becomes essential for efficient event filtering based on the specified rule criteria. Hence, establishing a standardized rule context and interface that can be seamlessly implemented across the FedIoT networks becomes imperative.

Within the current IoT frameworks incorporating rule support, two pivotal issues emerge: the communication protocol and the format of the rule context. The rule context assumes a pivotal role in facilitating the execution of rule scenarios by the rule operator. The effective interpretation and utilization of this contextual information require the rule operator to possess a high level of skill in discerning the context and extracting pertinent details. This proficiency enables informed decision-making during the execution of rules, enhancing the overall effectiveness of the rule-based system [16]. Furthermore, in order to assure proper rule transmission and reception for deployment, the rule client and server must use a unified communication protocol. However, with heterogeneous FedIoT networks established under different communication protocols and rule context formats, existing frameworks struggle to maintain a consistent rule model [17]. To address these issue, the interworking proxy can be used to translate protocols across two disparate FedIoT networks. To this aim proxy serves as a crucial network element that plays a central role in promoting coherence among IoT networks [18].

This is achieved by eliminating discrepancies, ensuring uniformity in data formatting, and facilitating consistent service interfaces. Therefore, using a proxy with rule-enabled FedIoT platforms can help ensure that the same rules are applied uniformly across all connected devices. For rule-enabled FedIoT platforms, this connection also ensures the smooth operation of rules across spatially distant and cross-silos heterogeneous FedIoT networks.

Furthermore, there exists heterogeneity in terms of device resources, dynamic communication standards and device types [19]. The devices within a FedIoT network possess a broad spectrum of capabilities, which encompass diverse processing, communication, and storage abilities. These capabilities include factors such as computing power, memory size, network speeds (such as 4G, 5G, WiFi), and battery life [20]. Existing solutions face a significant limitation due to their assumption that all participating devices possess identical computational and communication capabilities [21]. However, in reality, the devices exhibit varying capabilities, leading to inefficient weight updates. This issue becomes particularly pronounced in synchronous federated learning, where the edge server must wait for updates from all learning nodes, before computing global updates [22]. Consequently, the presence of a slow learning node also known as stragglers acts as a bottleneck, causing delays in the overall decentralized training process. To mitigate this challenge, researchers have proposed asynchronous model training approaches [23]. However, the scalability of such solutions is limited due to communication overhead and non-deterministic convergence. As a result, it

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3308579

becomes crucial to optimize the local model complexity based individual computation and communication capabilities of edge nodes.

Our study focuses on improving interoperability in FedIoT systems, with a special emphasis on the resolving challenges posed by system and statistical heterogeneity. The objective of this study is to develop an innovative FedIoT system that can effectively handle statistical and system heterogeneity, while ensuring fairness in federation. As well as establishing seamless connectivity among cross-silo nodes. To achieve this, we proposed a framework of common rules facilitation framework to provide interworking among heterogeneous FedIoT networks. The proposed architecture for interoperability effectively manages the diverse communication protocols and rule models embedded within the frameworks, ensuring consistent rule operations. In addition, this paper proposed a fair aggregation approach that allows the server to dynamically alter the width of hidden channels in a deep learning model based on the processing and communication capabilities of cross-silo nodes. Our developed system aims to collaboratively train a thermal comfort prediction model. To establish a heterogeneous FedIoT network, EdgeX, OCF and oneM2M standard platforms networks are developed and established. The proposed framework represents a novel approach to addressing statistical and system heterogeneity in federated learning systems, and is currently the most advanced solution available. Our framework addresses the challenges arising from heterogeneous statistical and system variations, thereby facilitating improved collaboration and effective learning in federated settings. The noteworthy contributions of the proposed work are as follows:

1) Design and implementation of a common rule facilitation based interworking proxy for seamless connectivity and interoperability among heterogeneous FedIoT (oneM2M, OCF, EdgeX) platforms.
2) Hetero-FedIoT encompasses a protocol and rule translation framework, which incorporates a proxy layer between edge nodes, gateways, and federated server. This framework serves as a bridge to connect heterogeneous nodes, facilitating seamless communication.
3) The proposed framework allows participating nodes to adapt global models with heterogeneous complexity levels according to their computation and communication abilities.
4) The learning outcome of developed Hetero-FedIoT framework remains stable and effective even when the system heterogeneity undergoes dynamic changes.

The remaining sections of the paper are organized as follows: Section II discusses the related work. Section III presents the proposed methodology, and detailed working of the system as well as the operational and functional overview. Section IV describes the experimental setup and implementation details. Section V presents the findings of the proposed systems including comparative analysis with baseline schemes. Lastly, Section VI provides the conclusion and outlines future directions for research.

II. Background and Related Work

Despite the inherent heterogeneity of FedIoT networks, the existing solutions often assume a scenario where the edge nodes participating in collaborative training are homogeneous. In practical settings, decentralized training faces a range of challenges including diverse device types and resources, varying network quality, communication protocols, and underlying platforms. As a result, heterogeneity has proven to have consequences for the efficiency and performance of FedIoT networks. Therefore in order to make disparate FedIoT networks interoperable and seamlessly connected, it is imperative to develop solutions to tackle the heterogeneity of protocols and data transfer formats.

Generally, to send and receive data from disparate device networks such as OCF [24], AllJoyn [25] and oneM2M [26] various studies employed techniques such as service layer proxy. Another study supported the use of an interworking proxy to facilitate communication between OCF and non-OCF device networks. In this regard, the interworking proxy is used for bridging OCF clients and servers [27]. In similar attempts, an OCF enabled multi-protocol gateway is developed by [28] for semantic conversion and protocol translation for accessing non-OCF services. Like wise [29] proposed a proxy framework for communication between edge nodes to provide interoperability among heterogeneous communication protocols. In order to provide services, OCF devices have access to OCF resources. The OCF rule server is built to support condition-action based autonomous decision making, as per the optional specification. Similarly [30], has suggested the action triggering mechanism to automate the action command execution based on conditions for the oneM2M systems.

Rules engines are often deployed in IoT edge systems to support dynamic and autonomous operations. By automatically performing actions in response to triggered conditions based on a predefined set of rules, they promote autonomous decision-making. In a similar manner, FedIoT systems may more efficiently employ rule and protocol translation to reliably operate collaborative model training service [31]–[33]. Multiple rule-based systems have been proposed to provide self-governing and adaptable service scenarios in IoT networks. For instance, [34] presented a semantic rule-based approach for the automated administration of IoT devices in built environments. This method identifies and executes events, conditions, and actions to facilitate device management. In another study, [35] introduced a rule-based event processing mechanism that is suitable for heterogeneous sensing devices using the Drools framework. The authors of [36] developed a rule-based event processing mechanism for monitoring and controlling nodes in real-time. The framework under consideration utilize the functionalities of Drools for effective rule-oriented processing, thereby ensuring optimal monitoring and control operations in a real-time. The authors of [37] introduced the Representational State Transfer (RESTful) framework for rule administration. The framework offers a variety of customizable and scalable options to support the provision of IoT services. Luo et al. [38] have developed another scalable Drools rule-engine for the edge gateway, focused on complex event processing using triggered conditions and actions. To address real-time events at the edge of networks, Choochotkaew et al. [39] developed a complex event processing engine. Additionally, Chen et al. [40] designed a real-time data-stream filtering engine using Drools framework. These efforts highlight the significance of handling real-time data and events efficiently at the edge of networks. Moreover, within federated IoT systems, the rule and protocol translator assumes a pivotal role, facilitating rule-based interoperability across heterogeneous platforms.

EdgeX is an open-source platform that leverages microservices to effectively manage edge nodes and their data. With its micro-services architecture, it enables seamless management, security, and accessibility of data across a diverse range of edge systems [41]. Through the adoption of a micro-services approach, the deployment of multiple device proxies on an EdgeX-based gateway platform becomes feasible, thereby enabling efficient protocol and rule translation within FedIoT networks [42]. Furthermore to enable flexible event processing at the edge gateway, Kuiper, an open-source rules engine for EdgeX, is utilized. Kuiper utilizes SQL-based rules and employs rule profiles written in the JSON format to provide a flexible and configurable framework for event processing and action execution at the network edge.

The existence of heterogeneity poses challenges not only in the development of software that operates on various platforms with complex libraries and frameworks but also in facilitating efficient communication among distinct protocols within FedIoT networks. These aforementioned challenges can be addressed by the adoption of transparent computing [43], which guarantees that edge nodes can conveniently access services through uniform interfaces, irrespective of the protocols employed for communication [44]. This approach enhances connectivity and fosters seamless interactions among diverse components in FedIoT networks, playing a crucial role in establishing cohesive and interoperable communication across the network while overcoming the barriers imposed by heterogeneous environments. The adoption of transparent computing [45] enables edge nodes to make service requests independent of the underlying protocol standards, data pipelines, and formats, reducing complexity in service provisioning. This concept is widely used by various edge applications to facilitate data sharing across connected devices, ensuring uninterrupted service delivery. Consequently, to support communication between several protocols, an interworking proxy must be deployed, which can be situated either at the network's edge or built into the device itself [46].

In the context of heterogeneous protocol integration, the authors of [47] utilized a proxy, and their proposed approach empowers FedIoT nodes to gain transparent access to edge services. This seamless integration of protocols allows for enhanced connectivity and streamlined communication, enabling efficient service provision in FedIoT networks.

The interworking proxy is a critical component for facilitating interoperability between OCF, and oneM2M device networks. By integrating the capabilities of HTTP client and server, as well as OCF and oneM2M client and server, allowing for smooth communication via request and response exchanges [48], [49]. The bridging specification for the OCF platform defines an architecture that enables the delivery of messages to disparate network environments by adapting dynamically based on the destination protocols of the client, server, and translator components. This adaptable architecture guarantees efficient message routing, enabling better communication across heterogeneous network environments. Notably, the aforementioned studies confirm that the proxy or gateway allows transparent access to various network components. Therefore, to enable communication between heterogeneous FedIoT systems, the provision of a protocol and data translator using a proxy becomes imperative. This translator offers a unified development environment for applications, enabling different frameworks to seamlessly communicate with each other and share data. The interworking proxy serves as a crucial component in bridging the gap across FedIoT networks, ensuring the consistent application of collaborative training.

The suggested rule-based interworking approach is thoroughly tested by implementing a transparent rule deployment and operation with EdgeX, oneM2M, and OCF. This comprehensive testing validates the effectiveness and reliability of the proposed solution, providing seamless rule execution and interoperability among the diverse frameworks within the FedIoT system.

Federated IoT networks have limited applicability due to heterogeneity challenges. Unfortunately, stragglers are not taken into account in the design of early FL techniques [50] nor are system and statistical heterogeneity. To guarantee performance and a fair contribution from all nodes, a key assumption is uniform involvement from all FedIoT nodes. In heterogeneous network environments, existing solutions often encounter challenges arising from unrealistic assumptions, leading to performance degradation. To mitigate the communication bottlenecks in the aforementioned networks, [51] proposed an adaptive averaging strategy. Additionally, [52] tackled the resource optimization issue by performing decentralized training at the network's edge instead of relying solely on a centralized server architecture.

Various studies [53]–[55] have attempted to find algorithmic solutions for the heterogeneity problem. Traditional approaches, however, are inadequate for large-scale IoT networks for collaborative training. Moreover, heterogeneity in FL systems is not solely limited to device resources. Additional system artifacts, such as label, class, and data distribution, node sampling, and end user behavior, introduce statistical heterogeneity into the network [56]. To resolve this issue, personalized local models for non-IID data are becoming increasingly popular in the scientific literature [57]. By incorporating techniques such as assisted learning [58], meta learning [59], multi-task learning [60], transfer learning [61] and knowledge distillation [62], the performance of collaborative training can be enhanced. However, the aforementioned techniques frequently incur unneeded processing and communication complexity. To the best of our knowledge, this study represents the pioneering effort in devising a solution to tackle potential heterogeneity issues in multiple FedIoT networks. This is accomplished by implementing a rule-based interworking architecture, which enhances communication and interoperability among the diverse frameworks within the FedIoT system. In this study an interworking rule proxy is developed for protocol translation at the edge gateway to achieve seamless server and node communication during federation. To address the impact of slow learners on overall global model performance, the proposed system includes a mechanism that allows edge nodes to train their respective local models according to their system capacities. This method seeks to mitigate the negative effects of slow learners on overall system's performance. Specifically, edge nodes with limited system resources can train less complex model architectures by altering the width of hidden channels. This adaptive approach ensures optimal performance and resource utilization in the context of diverse system capabilities within the FedIoT networks.

III. Rule-Based Interoperability for Heter-FedIoT Networks

This section introduces the methodology for the proposed heterogeneity-aware rule and protocol translation. The collaborative model training platform consists of three main components: a network of nodes, an edge gateway, and a server. To organize the edge nodes effectively, they are grouped based on two widely adopted IoT platform standards: OCF and oneM2M. The developed system comprise of a cluster of nCube and OCF nodes on the device layer each forming a FedIoT node network established upon their respective protocol standards. The edge nodes comprise of an AI engine and a Knowledge Agent (KA). The former is responsible for important system tasks such as model training, management, and data processing. It trains local models and facilitates the exchange of model updates via the KA, which serves as a means of communication for transmitting and receiving updated models. It has has two sub-modules that perform protocol and rule translation for making the heterogeneous federated systems interoperate. The local privacy sensitive data is stored in each node's data repository, while the model updates are stored in the model repository. The federated server, which comprises an EdgeX platform, establishes a gateway between the server and edge nodes. This gateway facilitates the implementation of the EdgeX micro-services platform to facilitate interworking among heterogeneous FedIoT networks. This aims to enable the devices to communicate and exchange data using the EdgeX framework, allowing them to participate in the federation. The basic architecture of the developed system is presented in Fig.1.

The proposed Hetero-FedIoT framework develops a rule assisted interworking proxy mechanism. EdgeX Common Rule Engine (CRE) Proxy is implemented as a micro-service within the EdgeX architecture to provide APIs for managing and executing rules. Rules indicate the specific conditions in which a particular course of action is to be pursued. With the EdgeX framework, rules specify the behaviour of the FedIoT system in response to certain events. While the CRE Proxy serves the purpose of regulating and coordinating the transmission of both local and global parameter updates among edge nodes and server according to predetermined criteria. The server

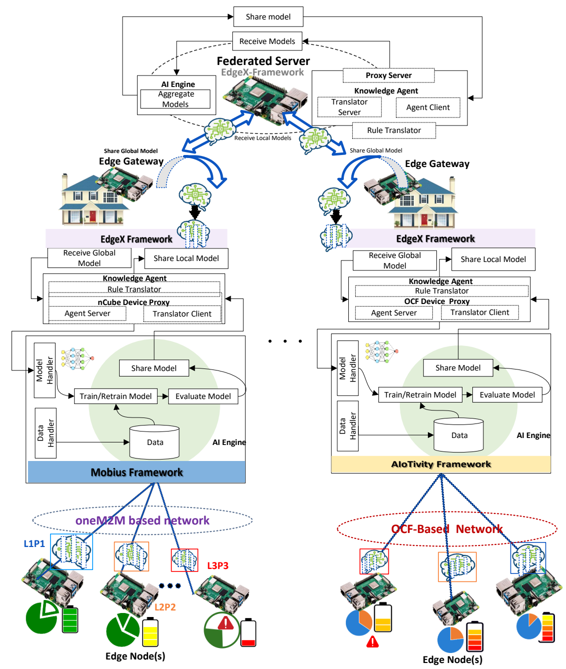

Fig. 1. Architecture of proposed rule-based interworking mechanism for collaborative training across heterogeneous FedIoT systems

transmits the global model to the FedIoT node networks during the initial phase. To mitigate the communication overhead and performance degradation caused by weak nodes in terms of computation power, the server adjusts the size of the global model based on node's capacity. This is accomplished by reducing the width of hidden channels within the deep learning model. Following receipt of the initial global model, the data handler is responsible for managing the data intended for local training. The nodes then use privacy-sensitive data to train the model for collaborative thermal comfort prediction. The local model parameters are transmitted to the server for aggregation after the training process. The local model parameters are sent from the KA to the server via the edge gateway in this transmission. Through rule and protocol translation, the KA consists of a translator client and an agent server that facilitate the transmission of local updates to the server and the reception of global updates from the EdgeX server platform. Protocol translator include proxy clients and proxy servers. The operational overview of the proposed system is shown in Fig. 2. The roles of agent server and client are switched while sending local updates and receiving global updates.

The rule-based federated knowledge deployment architecture in Hetero-FedIoT edge computing environment. Within the realm of Hetero-FedIoT, the rule service encompasses various components that play integral roles in its functioning. The fundamental constituents of the system are the rule handler, event handler, action handler, rule proxy, and rule repository. Together, they enable the automated implementation of FL. The rule handler serves as a platform for rule clients (OCF and oneM2M) to deploy rule contexts, which are subsequently stored in the rule repository. Furthermore, the rule handler

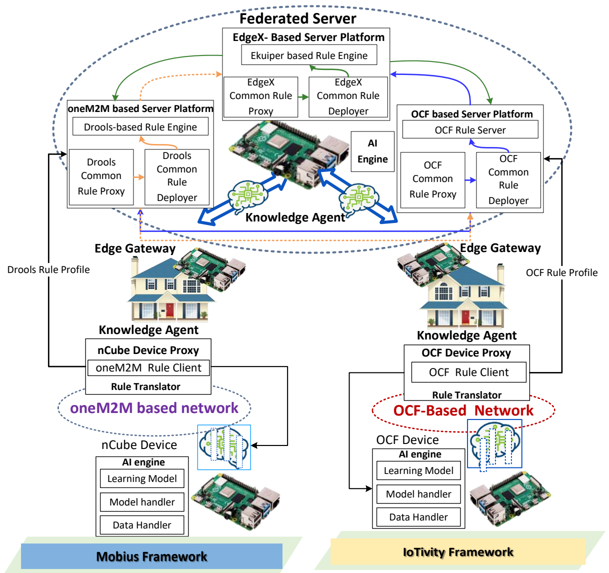

Fig. 2. Operational overview of rule-based interworking scheme for Hetero-FedIoT

offers functionalities for the collaborative training. The event handler evaluates local parameter updates from participating nodes to determine if the criteria for triggering the corresponding actions have been met.

The rule executor is responsible for assessing predetermined conditions and events. The aforementioned evaluations prompt the execution of corresponding actions, such as updating local models through aggregation and initiating the next round of training. The objecrive is to promote collaborative learning through the facilitation of communication and coordination among the distributed network. Subsequently, the action command is conveyed from the action handler to the AI engine, leading to model aggregation, broadcasting global updates to FedIoT nodes, and management of communication and synchronization between server and nodes. When translating from one protocol to another, a protocol translator employs a proxy client and proxy server to offer an additional layer of abstraction between the communicating parties. The proxy-client mediates between the nodes and the server, and then converts node requests into a language the server can comprehend. In response, the proxy server acts as a translator, by translating the server response into a format the nodes can comprehend. In cases where the nodes and server are using heterogeneous protocols, this method facilitates communication and provides interworking among disparate frameworks. The developed proxy components for Hetero-FedIoT do the protocol conversions and mediate conversations between the nodes and server for federated learning.

A Drools rule proxy is an interface that enables the definition and execution of rule-based on the Drools rule engine that govern the collaborative training process, specifying how data is partitioned and how and when to aggregate the model. After model aggregation, the updated global model is then inserted into Drools working memory and sent back to the nodes. It provides a collection of methods for creating and modifying rules as well as executing them against a set of data. The proposed interworking architecture also consists of an OCF rule proxy to specify rules for handling communication and data exchange between OCF and non-OCF compliant systems in a secure and controlled manner. The rule translators are deployed on each platform to convert the incoming common rule profile to a target network-specific format.

A. *Operational overview of rules-based interworking mechanism among Hetero-FedIoT networks*

This section presents the operational flow of the developed rule proxy for dealing with system heterogeneity issue that arise during collaborative learning of a global model in cross-silo manner. Fig. 3 illustrates the rule-based interoperability mechanism for collaborative training. The developed architecture consists of three layers: the device layer, an edge gateway in the middle, and the server layer. The EdgeX-based server platform incorporates oneM2M and OCF server rule engines, which trigger rule actions based on specified conditions. Additionally, a Mobius server is deployed for the oneM2M platform, housing an AI engine, a Drools rule engine, a Drools common-rule deploying agent, and a Drools common-rule proxy. To implement a common service entity, Mobius, an open-source IoT service platform for oneM2M, is utilized. Mobius acts as middleware, catering to various IoT application service domains, including registration, data management, subscription/notification, and security. In oneM2M systems, all services are invoked from the Mobius platform. Algorithm 1 and 2 present the process of the rule interworking scheme.

The device proxies are used to deliver local model updates to the server in the form of an event. The nCube device has a HTTP/MQTT client, while the OCF node has an OCF client for publishing the events to the server platforms. The nCube begins model training using local privacy sensitive data and gets local model parameters afterwards. The nCube device then takes on the role of a MQTT Publisher and transfers data to the Mobius server platform residing inside the server through the HTTP Protocol. Subsequently, the information is stored in a database, and a notification is transmitted to the rule processing system. Based on the registered local updates (model parametera) according to the rule profile, an action related to collaborative model training is triggered. One of the core contributions of the present study is that, previously in oneM2M specifications, Drools is not invoked by oneM2M specifically for the rule engine. The propose system develops a Drools-based rule engine and a Drools rule proxy for oneM2M node network. The Drools client communicates with the common proxy for translation and conversion of Drool rules to common format through the HTTP post handler, while in the case of receiving common rules, they are translated to Drools format and sent to the Drools rule engine for further action.

For the oneM2M platform, there was previously no implementation of Drools engine. Furthermore, the proposed work is the first attempt to implement the Drools rule engine on the

Algorithm 1: Hetero-FedIoT rule translation mechanism for translating platform-native format rules to common format

Data: Hetero-FedIoT Platform native format rules: (OCF, oneM2M, EdgeX)
Result: Rule profile

initialization;
Hetero-FedIoT platform ← The Current Heterogeneous platform ID;
Hetero-FedIoT profile ← Node initialization with a JSON object;
Hetero-FedIoT conditions ← JSON-Array Node Initialization;
Hetero-FedIoT actions ← JSON-Array Node Initialization;
Hetero-FedIoT ruleConditionList ← parse platform native format condition expression to ruleConditionList;
while *ruleConditionList.hasNext()* **do**
    Hetero-FedIoT condition ← JSON-Array Node Initialization; Hetero-FedIoTruleCondition ← ruleConditionList.next ();
    Hetero-FedIoTcondition.put("parameter", ruleCondition.getParameter());
    Hetero-FedIoTcondition.put("operator", ruleCondition.getOperator());
    Hetero-FedIoTcondition.put("value", ruleCondition.getValue());
    Hetero-FedIoTcondition.put("option", ruleCondition.getOption());
    Hetero-FedIoTconditions.add(condition);
Hetero-FedIoT ruleActionList ← parse platform native format actions to RuleActionList;
while *criterion for stopping not met* **do**
    action ← initialize JSON-object node;
    Hetero Fed-IoT ruleAction ← ruleActionList.next () ;
    while *ruleActionList. hasNext()* **do**
        action ← JSON-Array Node Initialization;
        Hetero-FedIoT ruleAction ← ruleActionList.next () ;
        Hetero-FedIoTaction.put("uri", ruleAction.getUri());
        Hetero-FedIoTaction.put("command", ruleAction.getCommand());
        Hetero-FedIoTactions.add(action);
    Hetero-FedIoTaction.put("uri", ruleAction.getUri());
    Hetero-FedIoTaction.put("command", ruleAction.getCommand());
    Hetero-FedIoTactions.add(action);
Hetero-FedIoTprofile.set("conditions", conditions);
Hetero-FedIoTprofile.set("actions", actions);
Hetero-FedIoTprofile.set("platform", platform);
Hetero-FedIoTprofile.set platform-specific properties with values;
return profile;

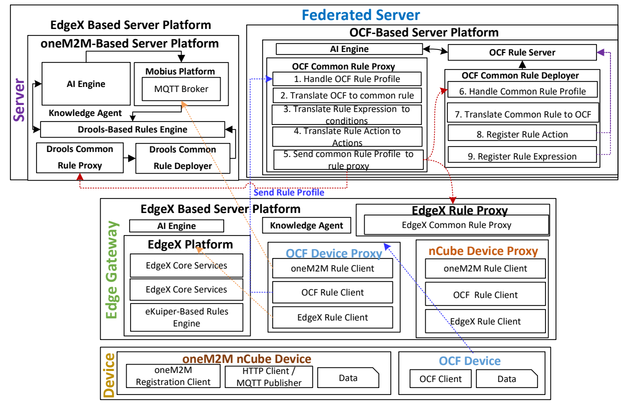

Fig. 3. Layered architecture of rule-based interworking approach for Heter-FedIoT networks

Mobius platform. The Drools based rule engine comprises an MQTT subscriber client to get data from the Mobius platform. For instance, if local model updates are received by the Mobius platform, they are subscribed to by the MQTT subscriber client. Drool rule-proxy uploads Drools rule-profile to the rule controller.

The OCF server platform offers resource-based services. Specifically, the developed OCF rule proxy comprises two main components: the OCF rule server and the OCF common Rule proxy. To register a rule profile on the OCF platform, the register function is utilized. This ensures proper integration and management of rule profiles within the OCF server platform. For deployment of rules on the OCF platform, the rule action resource defines the actions to be performed. OCF-IoTivity is configured on both the devices and the gateway for receiving and sending data from and to other platforms. On the server side, the Kuiper-based rule engine is deployed along with data and device management functionalities, as shown in Fig. 4. The Kuiper rule engine utilizes a straightforward declarative syntax to define rules, which are then executed against a given dataset to trigger specific actions. The server platform incorporates a device configurator that facilitates the registration process for participating nodes. Each node registers itself and communicates its capabilities for both communication and computation to the server. Based on the node resources, the server decides the model complexity level to be assigned. In the case of an EdgeX-based network, the EdgeX Common Rule Proxy receives Kuiper-based rule profiles from EdgeX clients, translates them into common format, and sends them to OCF and oneM2M nodes. The Common rule deployer performs translation to their own format and then registers the rule profile. In the case of EdgeX, the common rules are translated into Kuiper rules and then registered to the Kuiper rules engine for performing further actions.

OCF device proxy receives the data using IoTivity, while nCube device proxy receives the data through MQTT/HTTP. The respective device proxies, after receiving the local parameters, parse the received local updates and send them to the EdgeX data module, OCF, and oneM2M server platform.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3308579

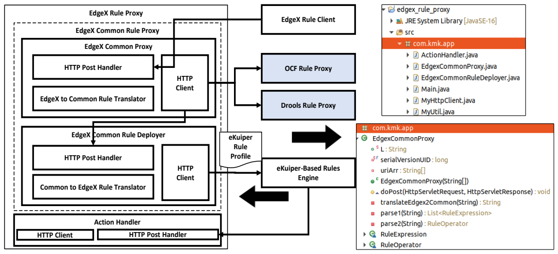

Fig. 4. EdgeX rule translation operation for collaborative knowledge sharing

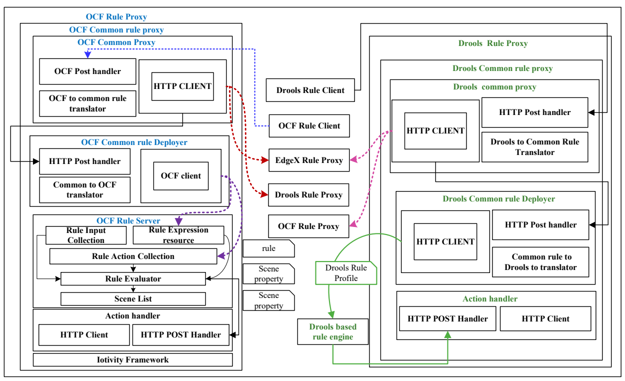

Fig. 5. Proposed Rule-Agent architecture to bridge FedIoT nodes and server interactions for interoperable Hetero-FedIoT system

After the server receives the data, the rule-engine in the server becomes active, and eventually rules events are triggered. Based on the contents of the rule profile, the AI Engine performs aggregation.

Fig. 5 presents the details of the interworking mechanism among heterogeneous networks of devices and federated servers. EdgeX, OCF, and Drools Rule Proxy all have two main functional blocks, namely Common Rule Proxy and Common Rule Deployer. The HTTP Post handler receives the OCF, Drools, and EdgeX-specific rules and translates them into common rules for interoperability. Subsequently the rule is sent to the common rule translation module for translation into a common format. Later, the HTTP client sends the data to other platforms using OCF and Drools Rule Proxies. For instance, for delivering messages to the OCF platform, the receiving body is the OCF rule proxy, and in the case of oneM2M, the Drools rule engine receives the data. Upon receiving common rules, the rules are translated into OCF and oneM2M. After conversion from common rule to OCF format, the rules are deployed to OCF rule server. One of the main contributions of the proposed interworking scheme is the development of an OCF rule server using Java, which was only partially implemented in C language previously.

Once the common rule profile has been transmitted to the associated device proxy, the device proxy converts the rule profile to the platform-native format to execute the federation process. The rule conditions are initially processed to assign condition values in line with the platform's native format via a common rule profile. Then, the rule actions are performed according to the platform-native format using the common rule profile's action list. The following actions are performed: local model updates and fine tuning, weight collection from nodes, model aggregation, global model distribution and broadcast, model deployment, and data transfer. The conditions and actions for FedIoT nodes and server are specified in the following format: $Whennode/server < condition >; then < action >; end$. The rules are specified using a number of operators. These operators are used to describe the criteria that should activate a rule and are combined to construct more complicated rules. The operators specify the conditions under which the local model is trained and when to perform aggregation after receiving local updates from the nodes.

Algorithm 2: Hetero-FedIoT rule translation mechanism for translating common format format rules to platform-native

Data: Rule Profile: (OCF, oneM2M, EdgeX)
Result: platform specific data entity
initialization;
Hetero-FedIoT ruleConditionList ← parse profile condition expression to RuleConditionList;
while *ruleConditionList.hasNext()* **do**
  Hetero-FedIoT ruleCondition ← ruleConditionList.next ();
  handle platform specific actions;
Hetero-FedIoT ruleActionList ← parse profile actions to RuleActionList;
while *ruleConditionList.hasNext()* **do**
  handle platform specific actions;
  if *required properties are null* **then**
    required properties← deafault values;
  else
    required properties ← profile values;
if *required properties are null* **then**
  required properties← deafault values;
else
  required properties ← profile values;
return platform specific data

B. Proposed model consolidation scheme under system and statistical heterogeneity

The primary objective of federation is to facilitate the collaborative training of a global model by leveraging the locally available privacy-sensitive data denoted by $\{D_1 \ldots D_n\}$ owned by n number of nodes. The model parameters denoted by $\{P_1 \ldots\ldots P_n\}$ serve as the parameterization for the local models. The server receives local model updates and aggregates them to acquire a global model after consolidation. The process continues till multiple communication rounds until optimal global model is formed that performs well across all the nodes. The iterative process can be formulated as $M_g^k = 1/n \sum_{i=1}^{n} M_i^k$ at k^{th} iteration. Afterwards updated global model M_g^t is shared with local nodes for retraining as follows, $M_i^{k+1} = M_g^k$. The objective of the proposed model consolidation scheme is to reduce the communication and computational complexity by assuming that the local models possess an identical architecture to that of the global model. However, the node can optimize its model complexity according to its computational and communication abilities. In this scenario, the local parameters are subset of global parameters such that the following property holds true; $M_i^{k+1} \subseteq M_g^k$. Our proposed aggregation scheme draws inspiration from the following studies [63], [64] which demonstrated that the model complexity can be adjusted by modifying the width and depth of hidden channels in deep learning frameworks, all while maintaining performance. We adopted this property of deep neural networks to improve communication efficiency and achieve stable model aggregation in the presence of nodes with heterogeneous resources.

Firstly, the procedure for selecting subsets of global model parameters is detailed as follows: The AI Engine selects a subset of global model parameters, referred to as $M_h l$, specifically for a single hidden layer, for which the parameters are determined by $M_g \in S^{oc_g \times ic_g}$. Where oc_g and ic_g are the output and input channel sizes, respectively. We considered three possible levels of varied model complexity denoted by p within the Hetero-FedIoT system defined as $M_{hl}^p \subset M_{hl}^{p-1} \ldots \subset M_{hl}^1$. Considering the hidden channel's shrink ratio as sr, satisfying the condition $oc_{hl}^p = sr^{p-1} oc_g$ and $ic_{hl}^p = sr^{p-1} ic_g$. Given the aforementioned conditions, the size of the local model parameters can be expressed as $|M_{hl}^p| = sr^{2(p-1)} |M_g|$ while the shrink ratio of the model is quantified as $SR = \frac{M_{hl}^p}{M_g} = sr^{2(p-1)}$. Subsequent to the adaptive composition, the proposed technique effectively allocates subsets of global model parameters to nodes according to their heterogeneous resource capacities. For instance, if the number of nodes according to computational complexity are symbolized as $\{n_1 \ldots n_p\}$ the server aggregates the parameters based on the following scheme:

$$M_{hl}^p = 1/n \sum_{i=1}^{n} M_i^p, \frac{M_{hl}^{p-1}}{M_{hl}^p} = \frac{1}{n - n_p} \sum_{i=1}^{n-n_p} \frac{M_{hl}^{p-1}}{M_{hl}^p} \ldots \quad (1)$$

In the above equations the notation M_i^P represents the tensor. Taking into account the upper left sub-matrix $M_g^k[: oc_n, ic_n]$, which is a fixed size oc_n and ic_n. Further, elements set $M_g^{p-1,k+1} \backslash M_g^{p,k+1}$ refers to the collection of elements that are contained within the set $M_g^{p-1,k+1}$ however omitted from $M_g^{p,k+1}$. While (1) demonstrates the aggregation of model parameters from all local nodes at the lowest computational complexity level, denoted as $L3$. In the latter part of the equation, model parameters are collected from nodes with lower computational complexity levels. Specifically, the difference between the orange subset $(p-1)$ and the red subset (p) is considered.

$$\frac{M_{hl}^1}{M_{hl}^2} = \frac{1}{n - n_{2:p}} \sum_{i=1}^{n-n_{2:p}} \frac{M_i^1}{M_i^2} \quad (2)$$

Likewise, in (2), the model parameters corresponding to the blue complexity level can be aggregated from a total of $n - n_{2;p} = n_1$ FedIoT nodes

$$M_g = M_{hl}^1 = M_{hl}^p \cup \left[\frac{M_{hl}^{p-1}}{M_{hl}^p} \right] \cup \ldots \frac{M_{hl}^1}{M_{hl}^2} \quad (3)$$

In (3), the global parameters are obtained by combining disjoint sets through their union operation. The aforementioned equations are illustrated using Fig. 1. The detailed working of these equations is depicted in Algorithm 3.

Algorithm 3: Hetero-FedIoT: Heterogeneity-Aware Federated Learning Mechanism

---

Input: Local Data D_i distributed on N edge nodes, Number of local epochs E, batch-size BS, node resources Nr, Learning-rate η global model parameters M_g, shrink ratio S, Model complexity level P, Server round SC

Function *system executes***:**
- Initialize global model parameters based on node resources
- **for** *each communication round k=0,1...* **do**
  - $\mathrm{W}_k(ActivenodesN, 1)$
  - $S_k \leftarrow randomlyselectafractionofactivenodes\mathrm{N}_k$
  - **for** *each node* $N \in S_k$ *in parallel* **do**
    - Determine model Complexity L:P
    - $Sr_L \leftarrow Sr^{(p-1)}, oc_L \leftarrow sr_L oc_g, ic_n \leftarrow Sr_n oc_g$
    - $M_L^k \leftarrow M_g^k[: oc_n, : ic_n]$
    - $M_L^{k+1} \leftarrow$ NodeUpdate $\left(L, sr_n, M_L^k\right)$
  - **for** *each complexity level* ***LP*** **do**
    - $\mathrm{M}_g^{p-1,k+1} \backslash M_g^{p,k+1} \leftarrow \frac{1}{N_k - N_{p:P,k}} \sum_{i=1}^{N_k - N_{p:P,k}} M_i^{p-1,k+1} \backslash M_i^{p,k+1}$
  - update $Nr_{1:Z,\eta}$
- Query representation statistics from local FedIoT nodes (Optional)

Function *NodeUpdate(L, Sr_L, M_L):***:**
- **BS** $\leftarrow Splitlocaldata\mathbf{D}_L$ **in batch-size BS**
- **for** *each epoch (E) from 1-n* **do**
  - **for** *batch* $b_L \in B_L$ **do**
    - $\mathrm{M}_L \leftarrow M_L - \eta \nabla \ell (M_L, Sr_L; b_L)$
- Return local model parameters M_L to server

The primary objective of the process is to perform parameter averaging by considering specific edge nodes that possess the corresponding parameters in their allocated parameter tensors. As a consequence, the parameters of a model with medium complexity are aggregated with both large models in their entirety and partially with small models. To achieve this, a fixed sub-network is assigned to each model complexity level, ensuring a consistent and stable global model. This approach facilitates the concept of global aggregation across all local models belonging to the same sub-network. Moreover, a constant sub-network is allocated for each model complexity level to mitigate node drift during the aggregation process, effectively handling statistical heterogeneity.

Accordingly, nodes with less complex local models make efficient use of resources by aggregating with models that match their complexity level. Following the distribution of global updates to the nodes, the parameters of local models are optimised using their local private data. The proposed approach chooses sub-networks directly from global model parameter subsets. A scaling module is used for scaling representations during training time, so a global model can be deployed for dynamic edge inference without requiring to be scaled. The proposed architecture incorporates a typical linear hidden layer in the following form:

$$y = f_{\text{act}} \left(\gamma * \left(BN \left(sm \left(D_n M_n^p + b_n^p - \text{ mean } \right) / std + \beta \right) \right) \tag{4}$$

where f_{act} denotes the RELU activation function, sm denotes a scaler module. The Parametric Layer and the sm are introduced before the scaled Batch Normalization BN and activation layers are applied. Whereas weights and biases for the local model are represented by $D_n M_n^p$ and b_n^p respectively. While γ and β are learn-able parameters, and y is the output. The computation and communication abilities of the edge nodes are defined as L_n. When the server receives this information, it can determine the proper model complexity to assign to the FedIoT node.

IV. Experiment Setup

In this section, the specifics of the development environment are detailed, the experimental setup is described, and the research findings are presented.

A. Data Description

The proposed framework is evaluated using open-source thermal comfort and digit classification data-sets namely ASHRAE RP-884, Scales, Medium US and MNIST respectively. The employed thermal comfort data has been widely used in numerous studies such as [65], [66] making it one of the most widely used public databases for examining human thermal comfort. The ASHRAE RP-884 data set consists of more than 25000 observations from 52 studies and 26 cities in various climate zones around the world. Further, the data contains 70 features among which 10 are common to three of them. Firstly, the data-set is pre-processed to remove abnormalities, missing values are removed, and the 7-point thermal sensation scale is re-classified to a 5-point scale (-2, -1, 0, +1, +2) by merging the cold and very cold states. The data distribution on the nodes is highly non-independent and non-identically distributed, fostering statistical heterogeneity. The common features selected for collaborative model training are presented in Fig. 6. The features are classified in terms of indoor and outdoor environments and occupant features.

For model training, Long Short-Term Memory networks and Convolutional Neural networks (LSTM-CNN) are used, The data is encoded into numerical values, and the missing

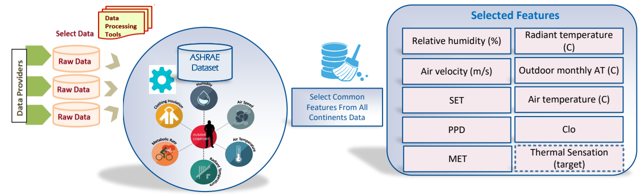

Fig. 6. Selected features for thermal comfort prediction

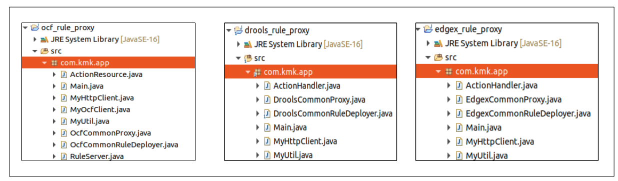

Fig. 7. Implementation details of rule-based interworking architecture

values are filled using the datawig library. Afterwards, the data is scaled and normalized in a specified range for improved performance using Min-max scaler normalization. Table I shows a comprehensive breakdown of the selected features.

B. Implementation Setup

Table II, IV and III provides overview of development environment for interoperable FL. The development of the framework is motivated by the heterogeneity observed in statistical, system, platform, and device resources. For the purpose of testing REST APIs, we make use of Talend's API tester as the web client. The rule registry and the device proxies are both built on Jetty-based micro-services, facilitating communication between the various parts of the rule platform.

The Spring-Boot Framework is utilized for constructing graphical user interfaces that display the registered rule schema and profiles. Additionally, the Jackson library, dedicated to processing JSON format, is employed. The EdgeX-based rules engine is responsible for rule enforcement in an EdgeX-based edge network. For configuration purposes, the Hanoi edition of the EdgeX framework is applied, while EMQ Kuiper is leveraged for rule management on the EdgeX platform. Detailed implementation specifics of the rule agent platform are presented in Fig. 7. To facilitate the deployment of OCF services, IoTivity 2.2.2 has been integrated into the OCF network. Within this network, the developed rule server allows for the automatic processing of rules. The EdgeX platform, featuring rule functionality, is hosted on a Raspberry Pi running Ubuntu server. For acquiring paramter updates and data from FedIoT node networks, a Windows system is configured as a proxy server running an OCF-server platform.

To verify the viability of the proposed Hetero-FedIoT architecture and collect rule and protocol translated data, three separate rule scenarios are set up. Firstly, EdgeX, oneM2M, and OCF-based rule servers, along with an OCF device proxy, are implemented. The OCF device network sends local model updates via the OCF device proxy to the EdgeX, oneM2M,

TABLE I
EXPERIMENTAL DATA DESCRIPTION

Attribute Types	Attribute Name	Attribute Description	Measuring Units
Indoor variable	I_{AT}	Indoor temperature	$^{\circ}C$
	I_{RH}	Indoor relative humidity	%
	I_{AV}	Indoor air velocity	m/s
	I_{IRT}	Indoor radiant temperature	C
Personal variable	CL	Clothing insulation	CLO
	MR	Metabolic rate	Met
	SET	Standard Effective Temperature	SET index
	PPD	predicted percentage of dissatisfied	%
Outdoor variable	O_{AT}	Outdoor temperature	$^{\circ}C$
Target variable	TS	Thermal Sensation	5-point thermal scale

TABLE II
DEVELOPMENT AND IMPLEMENTATION ENVIRONMENT OF OCF DEVICE PLATFORM FOR RULE-BASED INTERWORKING

Entity	**Library and Framework**
OCF Rule Server	Iotivity-lite 2.2.2, Jackson 2.11.4, Jetty-9.4.40v20210413, HTTP client-4.5.13, javax.servlet-api-2.11.4
OCF Rule Client	Iotivity-lite 2.2.2, Jackson 2.11.4
OCF Rule Proxy (Java)	Jetty (HTTP Server), Apache HTTP Client, Iotivity (OCF Server/Client)
OCF Device Proxy (Java)	Apache HTTP Client, Iotivity (OCF Server/Client)
OCF Device (Java)	IoTivity (OCF Client)
Hardware, OS	Raspberry Pi 3, 4, Model B, Ubuntu 20.10 Arch 64
AI Engine (Python)	TensorFlow (Deep Learning)

TABLE III
DEVELOPMENT AND IMPLEMENTATION ENVIRONMENT OF EDGEX GATEWAY PLATFORM FOR RULE-BASED INTERWORKING

Platform	**Entity**	**Library and Framework**
EdgeX Gateway Platform	EdgeX Rules Engine Deployer	Iotivity-lite 2.2.2, Jackson 2.11.4, jetty-9.4.40v20210413, HTTP client-4.5.13, javax.servlet-api-2.11.4
	EdgeX Rules Engine	EMQX Kuiper for EdgeX Hanoi Framework
	EdgeX Core Services	EdgeX Framework Hanoi
	Hardware, OS	Raspberry Pi 3, 4, Model B, Ubuntu 20.10 Arch 64

and OCF-based rule servers, depending on the target platform. Subsequently, rules are translated, and actions are performed based on the triggered rules. In the case of the oneM2M network, a oneM2M device proxy is developed to transmit local model parameters from the oneM2M device network to the EdgeX, OCF, and oneM2M rule servers.

C. Implementation of proposed rule-based interworking proxy for Hetero-FedIoT networks

We developed a platform for rule translation that enables the conversion of rules between platforms. The platform is comprised of multiple components, including a device proxy, a rule registry, rule repositories, and an interface for users. The device proxy, located in the gateway node, receives schema and rule profiles using format-specific protocols (OCF and oneM2M). It translates the rules from the specific format to a common format and performs format conversion on the rule profiles before forwarding them to the nodes using the platform's native protocol. The device proxy also includes a server to receive platform-native rule profiles and rule schema. Rules are deployed to the FedIoT platform using the rule schema, which specifies the conditions and procedures for training collaborative models. The standardized rule profiles and schema are stored in a central repository accessible through rule registration. When the registry receives a rule schema, it uses a schema parser and validator to extract and verify the information before saving it to the repository using a schema repository writer. The common rule profile includes platform information, the rule profile, rule conditions, and actions. The rule profile JSON data serves as the source for this information. The conditions and actions of collaborative model training are defined as JSON arrays, which are then serialized into lists of rule conditions and actions.

To conduct the collaborative model training, the device proxies for rule and protocol translation for OCF and oneM2M devices are deployed on Raspberry Pi 3, 4, and Model B devices. The OCF device services developed on the gateway act as a proxy for OCF-based rule servers, translating rule profiles from a common format into the platform-native format required by OCF-based rule servers. The same is the case with

TABLE IV
DEVELOPMENT AND IMPLEMENTATION ENVIRONMENT OF ONEM2M DEVICE PLATFORM FOR RULE-BASED INTERWORKING

Entity	Library and Framework
Drools-Based Rules Engine (Java)	Spring Boot (HTTP Server), Apache HTTP Client, Drools (Rules Engine), KETI nCube Client
Drools Rule Proxy (Java)	Jetty (HTTP Server), Apache HTTP Client
nCube Device Proxy (Java)	Jetty (HTTP Server), Apache HTTP Client, Iotivity-OCF Client
nCube Device	KETI nCube (nCube Registration Client), Apache HTTP Client
AI Engine (Python)	TensorFlow (Deep Learning)
Hardware, OS	Raspberry Pi 3, 4, Model B, Ubuntu 20.10 Arch 64

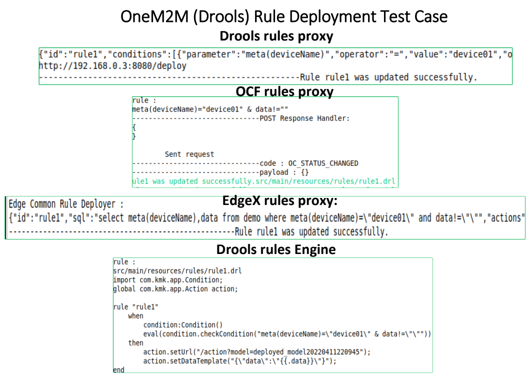

Fig. 8. Implementation of Drools rule deployment

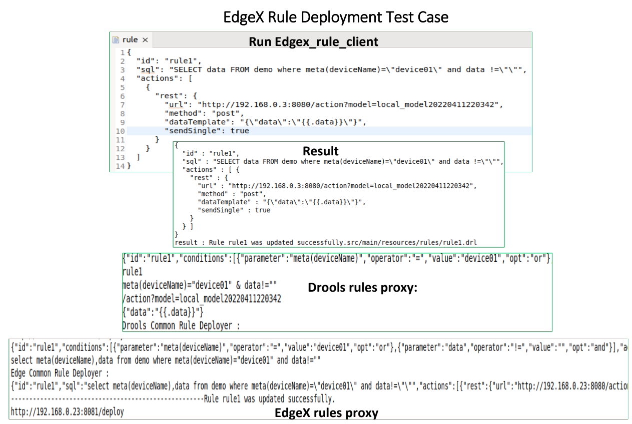

Fig. 9. Implementation of EdgeX rule deployment

oneM2M device networks. For OCF and oneM2M networks, the PC acts as federated server. Data from the device proxy is passed on to the rules registry. Based on the registered rule schema, the rule registry sends the common rule profile to the OCF and oneM2M platforms, allowing them to distribute it to the heterogeneous FedIoT networks. The implementation results of the oneM2M, EdgeX, and OCF rule deployments are presented in Fig. 8, 9 and 10.

By enforcing a condition-response logic structure, rules serve as resources that can make decisions on their own. In each case, if the rule expression evaluates to true, it triggers the associated Rule Action. The participating devices transmit their model updates to the device proxy, which then combines them and transmits them to the server. The server then utilizes these updates to update the global model and returns the revised model to the proxy, which distributes it to the participating devices. This procedure is performed numerous times to train a more precise global model.

Fig. 8 shows the test result analysis of oneM2M (Drools) Rule Deployment. Using Drools to deploy a rule in OneM2M involves the creation of a rule file in the Drools language, followed by the integration of the Drools rule engine into the OneM2M platform. oneM2M client sends Drools rule-profile to Drools Common Rule Proxy for translation. After translating the rule profile into common format, the common rule profile is sent to the rule proxy.

Afterwards It will be sent to oneM2M, EdgeX, and OCF platform from rule proxy. Each of them handles and translate it to their native format and then registers the rule in the rule repository, when a given event or condition occurs within the OneM2M system, the rule is triggered and action is performed. This process is repeated for each platform. This is accomplished by utilizing the Services Capability Layer (SCL) of oneM2M, which provides a means to register, discover, and access services.

The results of EdgeX based rule deployment are presented in Fig. 9, where EdgeX Client sends Kuiper rule profiles to the EdgeX Common Rule Proxy. EdgeX Common Rule Proxy handles and translates the common rule. After translation, the common rule profile is sent to rule proxy. From there it is then sent to EdgeX, OCF, oneM2M platform. The Fig. 10 shows the test result analysis of OCF Rule Deployment.

Fig. 11 depicts the rule agent platform, wherein each process is distinctly identified by a process ID associated with individual FedIoT platforms; PID 2786156 presents the rule registry; and PID 45398 shows the running EdgeX-based services. The figure shows the execution results of operating and distributing the local model via the EdgeX-based IoT edge computing platform. In the EdgeX-based platform, core-data, core-metadata, and core-command are modules that provide EdgeX services; the Java module provides the protocol switching function between HTTP and IoTivity; and the Python module provides

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3308579

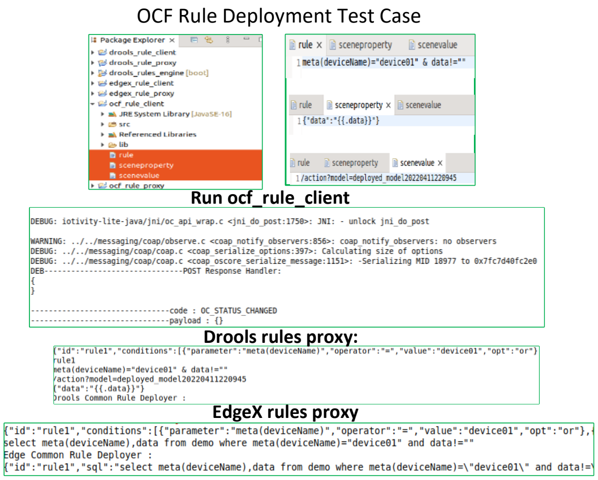

Fig. 10. Implementation of OCF rule deployment

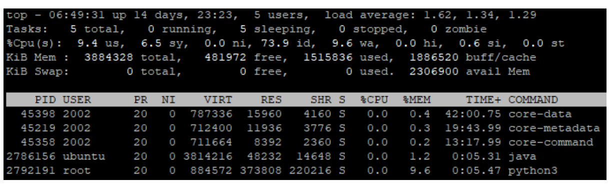

Fig. 11. Process monitoring of EdgeX-based FedIoT edge computing platform

training and model transmission functions. Total memory utilization is 1,515,836 KB, split as follows: core-data (15,960 KB), core-metadata (11,936 KB), and core-command (8,392 KB) for executing the EdgeX service; protocol switching rule proxy (48,232 KB); and Python (373,808 KB).

Fig. 12 shows the execution results for deploying rules in the oneM2M Mobius-based IoT edge computing platform . In the Mobius-based platform, the Node JS-based module is responsible for providing Mobius services, and the Python module provides learning and fine-tuning operations. To run the Mobius service, we run five Node JS-based modules, occupying 80,788 KB, 77,976 KB, 72,388 KB, 56,932 KB, and 77,560 KB, respectively. The Python module occupies 383,144 KB, and the total memory usage is 703,488 KB.

Lastly Fig. 13 shows the result stats for deploying rules. OCF resources are developed using the IoTivity Framework; rule agents based on Java modules provide protocol translation and conversion functions, and Python modules provide learning and model transmission functions. The module for

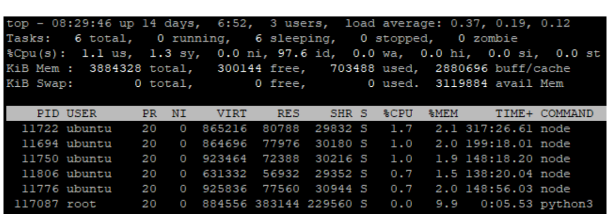

Fig. 12. Process monitoring of Mobius-based FedIoT network

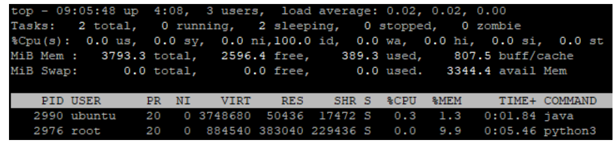

Fig. 13. Process monitoring of IoTivity-based FedIoT network

running the IoTivity service is excluded from the monitoring; the protocol switching proxy occupies 50,436 KB, the Python module occupies 383,040 KB, and the total memory usage is 389.3 MB.

V. RESULTS AND PERFORMANCE ANALYSIS OF THE PROPOSED HETERO-FEDIOT SYSTEM

The developed system involves 12 edge nodes, with six nodes located on each network. To evaluate the proposed aggregation scheme, we constructed models with three different complexity levels and a predefined shrink ratio for hidden channels. Each node is initially assigned a complexity level by the server depending upon the resources and capabilities of the nodes. The server assigns the optimal level of model complexity to each node depending on the reported communication and processing capabilities. To deal with nodes having varied resource capabilities, we uniformly sampled nodes from different combinations of computation complexity levels. For instance, each network assigns three different levels of model complexity to each of its nodes. Model L1:P1 has all the model parameters, whereas models L2:P2 and L3:P3 have a fixed shrink ratio of 0.5 and 0.25, respectively. Hetero-FedIoT edge platforms with rule support filter events and take action in response to rule triggers in real time. The architecture and hyper-parameters of the model are comprehensively presented in Table V. The hidden layers of the model are trained using the Rectified Linear Unit (Relu) activation function, while the output layer employs the Soft-Max function. During the training process, the categorical cross-entropy loss function is utilized. Furthermore, the optimizer employed is Adam, with a learning rate set to 0.001. A predetermined random seed is chosen for the purposes of training and shuffling the dataset.

During experiments, we assigned an initial level of computational complexity to each node. The word static is utilized to denote the situation where the complexity assignment remains constant throughout the experiments, whereas the term "Dynamic" is employed to refer to the scenario where edge nodes uniformly sample computation complexity levels during each communication round. Extensive experiments are carried out for both fixed and dynamic assignments in order to comprehensively examine their effects. The results shown in the figures are derived from the fixed complexity scenario, which involves the evaluation of models with a predetermined size for each node. Conversely, the tables offer valuable perspectives on the dynamic scenario, wherein the distribution of a node's model complexity is randomly altered while upholding a constant ratio of 50% for the quantity of weak learners in each FedIoT network. Through an examination of both fixed and dynamic assignment scenarios, our objective is

TABLE V
DETAILS OF THE EXPERIMENT AND MODEL HYPER-PARAMETER SETTINGS

Optimal Model Parameters	**Values**
Data	ASHRAE, Scales, Medium US
Model Name	LSTM-CNN
Activation Function	Relu, SoftMax
Optimizer	Adam
Learning rate	0.001
Filters (Conv layer)	128
Kernel size (Conv layer)	5
Server rounds	20
Momentum	0.9
Local Epochs	5
Local batch-size	128

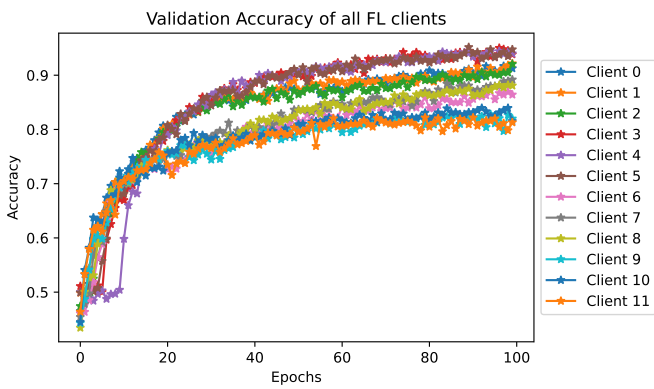

Fig. 14. Learning curves of FL nodes based on proposed Hetero-FedIoT architecture

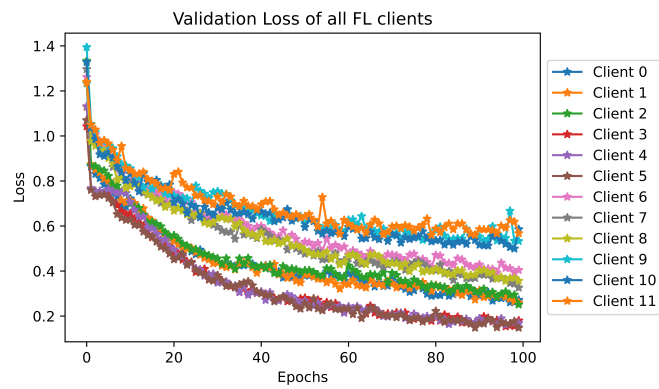

Fig. 15. Loss based performance analysis of edge nodes based on proposed Hetero-FedIoT architecture

to conduct a comprehensive evaluation of the impact on the system's overall performance.

A comparative analysis is performed, incorporating benchmark techniques such as local training, Fed-Avg, and LG-FedAvg [57]. The term local training refers to a situation where there is no interaction between the nodes and the server. Our experimental research encompasses more complex models (LSTM-CNN) compared to those employed in previous studies, particularly the LG-FedAvg approach, which utilized Multilayer Perceptron (MLP) architecture on the MNIST dataset. The global model results are derived by assessing the global model's performance on the validation data. Conversely, the computation of the local results involved the cumulative averaging of the performance results of individual data instances over all nodes.

From the performance score analysis, we can verify that, in comparison to the results achieved by baseline techniques, Hetero-FedIoT performs well despite statistical heterogeneity. The experimental results of the proposed system are depicted in Fig. 14 and 15. The graphs illustrate the validation accuracy and loss of edge nodes with fixed complexity assignments against the thermal comfort data set. While the global accuracy and loss are depicted in Fig. 16 and 17.

Upon closer inspection, it becomes clear that the edge nodes converged well without getting stuck in the local optima. The convergence is slow during the first few rounds; however, after obtaining an updated model from the server in subsequent iterations, their cumulative loss has been seen decreasing. The

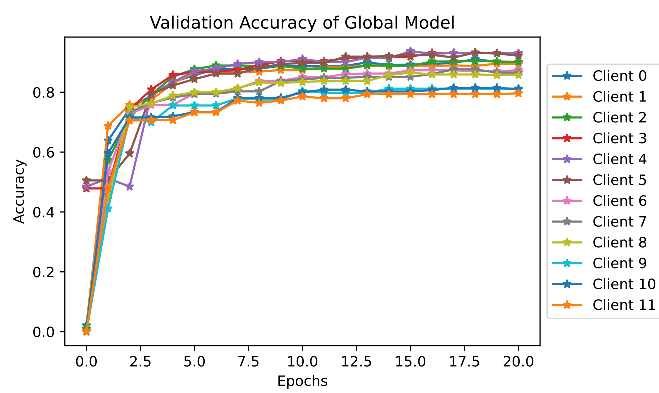

Fig. 16. Accuracy based performance analysis of global model based on proposed Hetero-FedIoT architecture

results also suggest that the performance of the CNN-LSTM model is better at capturing the spatial-temporal features and relations in the thermal comfort data. In Table VI we present the results of our investigation using Non-IID data with three complexity levels. In order to demonstrate the effects of heterogeneous computation and communication capabilities, we utilize uniform sampling across various combinations of computational complexity levels. The notation "L1-L2-L3" denotes the uniform sampling of all available levels of complexity for each node during each communication round. In order to depict the computation and communication requirements of our proposed system, we furnish details regarding the quantity

TABLE VI
SUMMARY OF RESULTS OBTAINED UNDER VARYING LEVELS OF COMPUTATIONAL COMPLEXITY FOR THE THERMAL COMFORT DATASET

Model Complexity	Model Ratio	No. of Parameters	Floating Point Ops.	Model Space Req.(MB)	Local Accuracy	Global Accuracy
L1	1.00	100K	5.2M	0.36	88.50	86.45
L1:L3	0.54	54K	2.8M	0.19	88.79	86.27
L1:L2:L3	0.43	42K	2.5M	0.19	88.76	86.85
L2	1.00	26K	1.39M	0.10	88.86	86.86
L2:L3	0.56	17K	910K	0.05	88.63	86.70
L3	1.00	8K	401K	0.05	88.07	86.84
Local Training	1.00	635K	1.31M	2.42	85.22	NA
FL-FedAvg	1.00	635K	1.31M	2.41	80.39	77.24
FL-LG-FedAvg	1.00	635K	1.31M	2.41	81.26	79.68

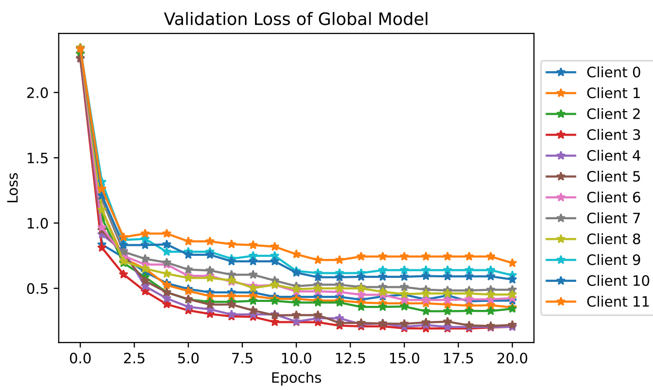

Fig. 17. Loss based performance analysis of global model based on proposed Hetero-FedIoT architecture

of model parameters, FLOPs (Floating Point Operations), and Model Space (MB). When complexity levels are uniformly sampled, the computational metrics of models L1-L3 are computed through the averaging of the metrics of models L1 and L3. The ratio is determined through the comparison of the parameter count of a specific model with that of the complete global model, expressed as a percentage. The outcomes demonstrate that the proposed scheme outperforms baseline Federated Averaging (Fed-Avg), LG-FedAvg as well as independently trained local models. Accuracy comparisons between the proposed Hetero-FedIoT and baseline methods show the superiority of our suggested method. The proposed scheme achieved an accuracy of 86.85%. Also, when comparing all performance indicators, Hetero-FedIoT based collaborative thermal comfort model produces the best results. According to the findings, our approach is just as effective as the ones that use the same local model complexity for all nodes. Our approach can be easily adapted to new contexts and has little computational overhead.

For further experimental evaluation of the proposed system, we conducted tests using the CIFAR-10 dataset to assess its performance. The results, presented in Table VII, showcase the effectiveness of our proposed system for image classification tasks using a CNN model. The experimental findings reveal that nodes with limited learning resources, which can only train the lowest complexity model (L3), achieve a global accuracy of 54.12% on CIFAR-10. However, by implementing the L2-L3 and L1-L3 configurations, where half of the FedIoT nodes are trained with larger models (L2:P2 and L1:P1) and the other half with model L3:P3, the weaker learners significantly improve their performance. The accuracies obtained for these configurations are 56.17% and 63.35%, respectively. These results are remarkably close to the hypothetical situation where all learners possess high proficiency levels, yielding global accuracies of 61.21% (L2) and 55.11% (L1). These findings illustrate the potential for significant performance improvements among nodes with low learning capabilities when the suggested method is used. Even weaker learners can obtain accuracies comparable to competent learners by using strategically designed model setups.

The obtained results demonstrate that our proposed strategy attains comparable performance to strategies that employ personalized local models. Our approach offers a high level of customization, incurs minimal computational overhead, and leverages a single global model for evaluating both local and global performance.

For evaluating the performance of the developed rule-enabled interworking proxy for Hetero-FedIoT networks, we provided the average training time, waiting time, model transfer time, model size, and latency comparisons to demonstrate the performance results. The Fig. 18 presents the average training and waiting times for edge nodes. Each node's average training time over all server rounds is shown in the graph. During the first round of collaborative training, the recorded time quantifies the time spent on model training, whereas the rest of the time accounts for local retraining. In the initial round, the server transmits the initial weights to the nodes, while in the subsequent rounds, the server disseminates aggregated weights. The duration of the training process is susceptible to variations in hardware capabilities and model sizes. Prior to generating the updated global model, the aggregator server platform awaits the arrival of local models from the participating nodes. The duration of this waiting period is influenced by factors such as the size of the model, the available resources on each node, the number of participating nodes, the frequency and magnitude of updates, the volume of data samples, and the latency within the network. Fig. 19 shows the sample size and average waiting time of each node. In order to decrease waiting time and improve the global model's performance, the suggested method adjusts the size of the local model based on the resources of FedIoT nodes.

From Fig. 18 it can be analyzed that nodes with shorter training times also have longer waiting times due to the fact

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3308579

TABLE VII
A DETAILED SUMMARY OF THE RESULTS ACHIEVED USING MNIST DATASET UNDER VARYING LEVELS OF COMPUTATIONAL COMPLEXITY.

Model Complexity	Model Ratio	No. of Parameters	Floating Point Ops.	Model Space Req.(MB)	Local Accuracy	Global Accuracy
L1	1.00	610 K	5.2 M	2.30	91.08	55.11
L1:L3	0.54	322 K	2.8 M	1.22	91.81	63.45
L1:L2: L3	0.43	261 K	2.5 M	1.01	91.50	55.39
L2	1.00	150 K	1.39 M	0.58	90.76	61.21
L2:L3	0.56	97 K	910 K	0.35	90.91	56.17
L3	1.00	36 K	401 K	0.15	89.61	54.12
Local Training	1.00	1.81M	3.5M	6.79	87.90	NA
FL-FedAvg	1.00	1.81M	3.5M	6.79	59.01	58.12
FL-LG-FedAvg	1.00	1.81M	3.5M	6.79	91.48	60.67

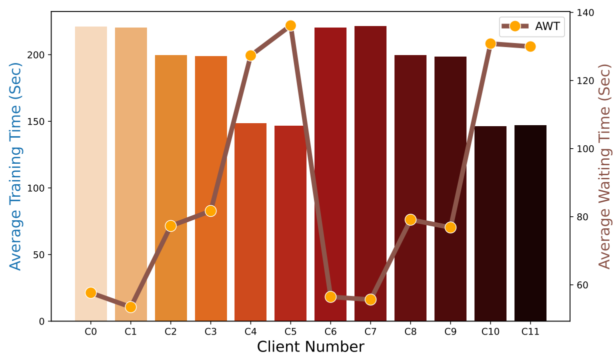

Fig. 18. Average training time of node with heterogeneous local models for 20 server rounds.

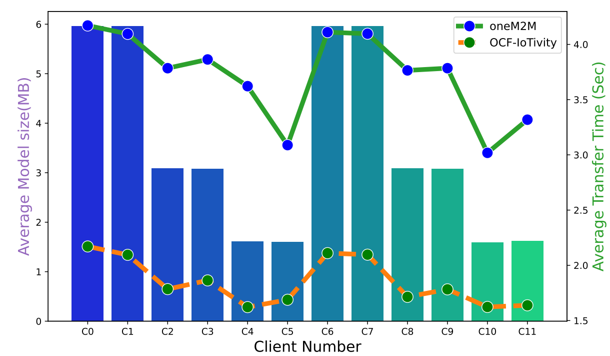

Fig. 20. Average model transfer time comparison between OCF-IoTivity and oneM2M FedIoT networks

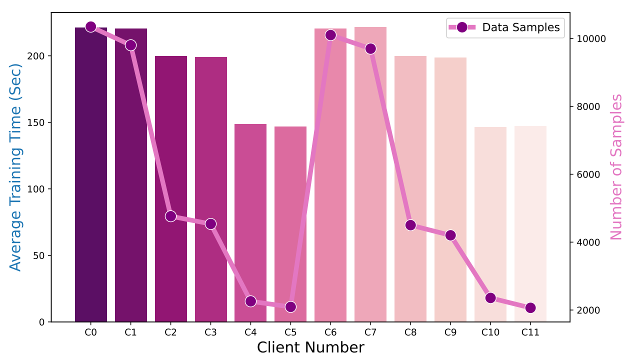

Fig. 19. Average training time and sample size of participating edge nodes

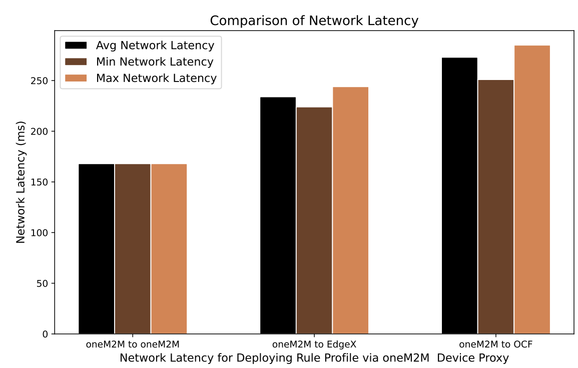

Fig. 21. Network latency for deploying oneM2M rules to Hetero-FedIoT networks

that the server needs to wait for all nodes to complete the training process before model aggregation. So the nodes that finished the training earlier have to wait longer for the updated global model.

Model transfer time and model size are also compared among FedIoT networks to demonstrate the efficiency of the OCF-IoTivity platform and the oneM2M platform. The average model size for each node along with the model transfer time are displayed in Fig. 20. From the graphical analysis, it is verified that OCF IoTivity has a significantly faster model transfer time compared to oneM2M because the latter uses the MQTT/HTTP protocol for communication. If several nodes complete training at the same time, the models can be transferred in parallel because they are all bound to a separate server port. In addition, after the aggregation procedure is complete, the server will send out a parallel broadcast across the dedicated ports, sharing the aggregated weights with each node.

Network latency for deploying a rule profile from an OCF device proxy to multiple FedIoT platforms is depicted in Fig. 21. Deploying a rule profile to multiple IoT platforms through an OCF device proxy uses a significant amount of memory. Here, we employed an OCF device proxy to deploy rules to OCF, EdgeX, and oneM2M platforms.

The Round Trip Time (RTT) of an OCF device proxy

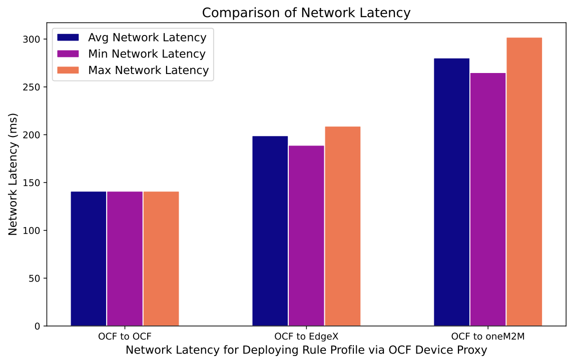

Fig. 22. Network latency for deploying OCF rules to Hetero-FedIoT networks

for cross-platform deployment has also been analyzed. The RTT analysis suggests that the average delay of each rule deployment increases as the number of heterogeneous platforms increases, as shown in Fig. 21 and 22 respectively. Requests are carried out asynchronously. The maximum network latency is hence the total delay time for deploying rule profiles from the OCF network to the OneM2M and EdgeX networks using device proxy. For deploying rules profiles from an OCF client to an OCF server, the minimum, maximum, and average dealay times are 141 MS.

Fig. 22 shows that as the number of platforms increases, network latency also increases. The experimental analysis demonstrates how the performance of the device and underlying platform influence network latency. Due to total memory usage and network resource consumption, node performance is affected, which in turn causes higher network latency. The empirical investigations suggest that the rule deployment delay of oneM2M proxy is higher than the OCF. On the other hand oneM2M-oneM2M node networks have an average RTT of 168 milliseconds, which is marginally higher than OCF node networks.

VI. Conclusion

The emerging FedIoT systems face significant challenges, primarily concerning heterogeneity and a lack of interoperability. These issues have become the leading causes of fragmentation within FL ecosystems. The continuous technological advancement necessitates cross-silo FL nodes to establish seamless connections and communication. However, achieving interoperability between heterogeneous FedIoT systems poses an even more complex and challenging task. To solve this issue, the proposed system develops a rule-based interworking architecture to provide interoperability among heterogeneous FedIoT platforms. Hetero-FedIoT is a comprehensive and unified solution to resolve the potential challenges of system and statistical heterogeneity in collaborative training. The developed solution is predicated on a rule-based framework that facilitates the cooperative training of heterogeneous local models and provides smooth connectivity between silo-spanning FL platforms. In our proposed architecture, the edge gateway plays a pivotal role in providing translation capabilities, effectively bridging the gap between disparate communication protocols and data formats. This capability reduces vendor lock-in and significantly boosts interoperability. The proposed rule-based strategy leverages device proxies to convert rules between platform-specific and standard formats, thereby enabling seamless collaboration between diverse platforms in heterogeneous FedIoT networks. The obtained results confirm the effectiveness of our method, as it demonstrates improved global model performance, reduced communication cycles, and enhanced connectivity. This efficient utilization of computational and communication resources further enhances the overall system performance. In our future work, we aim to enhance the interworking proxy by decoupling it from the IoT platform, transforming it into an independent entity capable of serving multiple IoT frameworks. This separation will empower the interworking proxy to efficiently manage rules and translation operations across diverse IoT networks, ensuring consistency in rule application. Additionally, we plan to incorporate a general rule context into the translation process, facilitating the seamless transfer of comprehensive rules to the target FedIoT networks. By considering framework-specific aspects, we can optimize the translation process and enhance rule integration within various IoT settings.

VII. Acknowledgment

This work was supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (2021-0-00188, Open source development and standardization for AI enabled IoT platforms and interworking), and this work is supported by the Korea Agency for Infrastructure Technology Advancement(KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant 21DCRU-B158151-02). Any correspondence related to this paper should be addressed to Dohyeun Kim. This research was also supported by Brain Pool program funded by the Ministry of Science and ICT through the National Research Foundation of Korea (2021H1D3A2A02082991)

Authors' contributions

Anam N K. and Wenquan J. conceived and designed the experiments. Analysis and interpretation of the data is performed by Atif R., Anam N K. and Rashid A. The experiments are conducted by Anam N K., Atif R. and Wenquan J. The paper is written by Anam N K. and reviewed by Sunhwan L. The final draft is reviewed by Qazi W K. Do-Hyeun K. contributed materials, analysis tools, data and supervision. All authors commented on previous versions of manuscript and approved the final version.

The authors declare no competing interests.

References

[1] A. Taïk and S. Cherkaoui, "Electrical load forecasting using edge computing and federated learning," in *ICC 2020-2020 IEEE international conference on communications (ICC)*. IEEE, 2020, pp. 1–6.

[2] Q. Luo, J. Lin, Y. Zhuo, and X. Qian, "Hop: Heterogeneity-Aware Decentralized Training," in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, Apr. 2019, pp. 893–907, arXiv:1902.01064 [cs]. [Online]. Available: http://arxiv.org/abs/1902.01064

[3] N. Truong, K. Sun, S. Wang, F. Guitton, and Y. Guo, "Privacy preservation in federated learning: An insightful survey from the gdpr perspective," *Computers & Security*, vol. 110, p. 102402, 2021.

[4] W. Jin, S. Lim, Y.-H. Suh, C. Park, and D. Kim, "Transparent Access to Heterogeneous IoT Based on Virtual Resources," *Computers, Materials & Continua*, vol. 74, pp. 4983–4999, Jan. 2023.

[5] C. Yang, M. Xu, Q. Wang, Z. Chen, K. Huang, Y. Ma, K. Bian, G. Huang, Y. Liu, X. Jin, and X. Liu, "FLASH: Heterogeneity-Aware Federated Learning at Scale," *IEEE Transactions on Mobile Computing*, pp. 1–18, 2022, conference Name: IEEE Transactions on Mobile Computing.

[6] R. Du, P. Santi, M. Xiao, A. V. Vasilakos, and C. Fischione, "The Sensable City: A Survey on the Deployment and Management for Smart City Monitoring," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1533–1560, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8533352/

[7] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated Learning for Internet of Things: Recent Advances, Taxonomy, and Open Challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1759–1799, 2021, conference Name: IEEE Communications Surveys & Tutorials.

[8] D. Yacchirema, A. BelsaPellicer, C. Palau, and M. Esteve, "OneM2M Based-Interworking Architecture for Heterogeneous Devices Interoperability in IoT," in *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, Oct. 2018, pp. 1–6.

[9] R. Xu, W. Jin, and D. H. Kim, "Knowledge-based edge computing framework based on coap and http for enabling heterogeneous connectivity," *Personal and Ubiquitous Computing*, pp. 1–16, 2022.

[10] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the internet of things," *IEEE access*, vol. 6, pp. 6900–6919, 2017.

[11] A. N. Khan, A. Rizwan, R. Ahmad, and D. H. Kim, "An ocf-iotivity enabled smart-home optimal indoor environment control system for energy and comfort optimization," *Internet of Things*, vol. 22, p. 100712, 2023.

[12] A. Rizwan, A. N. Khan, R. Ahmad, and D. H. Kim, "Optimal environment control mechanism based on ocf connectivity for efficient energy consumption in greenhouse," *IEEE Internet of Things Journal*, 2022.

[13] J. Zhang, J. Yang, and J. Li, "When rule engine meets big data: design and implementation of a distributed rule engine using spark," in *2017 IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService)*. IEEE, 2017, pp. 41–49.

[14] J. Hwang, J. An, A. Aziz, J. Kim, S. Jeong, and J. Song, "Interworking models of smart city with heterogeneous internet of things standards," *IEEE Communications Magazine*, vol. 57, no. 6, pp. 74–79, 2019.

[15] J. Sun and H. Zhao, "A novel cep model and its applications in internet of things big data processing," *International Journal of Machine Learning and Computing*, vol. 9, no. 6, p. 721733, 2019.

[16] S. Bansal and D. Kumar, "Iot ecosystem: A survey on devices, gateways, operating systems, middleware and communication," *International Journal of Wireless Information Networks*, vol. 27, pp. 340–364, 2020.

[17] W. Jin and D. Kim, "Distributed rule-enabled interworking architecture based on the transparent rule proxy in heterogeneous iot networks," *Sensors*, vol. 23, no. 4, p. 1893, 2023.

[18] W. Jin, R. Xu, S. Lim, D.-H. Park, C. Park, and D. Kim, "Integrated service composition approach based on transparent access to heterogeneous iot networks using multiple service providers," *Mobile Information Systems*, vol. 2021, pp. 1–19, 2021.

[19] J. Cao, Z. Lian, W. Liu, Z. Zhu, and C. Ji, "HADFL: Heterogeneity-aware Decentralized Federated Learning Framework," Nov. 2021, arXiv:2111.08274 [cs]. [Online]. Available: http://arxiv.org/abs/2111.08274

[20] A. M. Abdelmoniem and M. Canini, "Towards Mitigating Device Heterogeneity in Federated Learning via Adaptive Model Quantization," in *Proceedings of the 1st Workshop on Machine Learning and Systems*. Online United Kingdom: ACM, Apr. 2021, pp. 96–103. [Online]. Available: https://dl.acm.org/doi/10.1145/3437984.3458839

[21] J. Chen, X. Pan, R. Monga, S. Bengio, and R. Jozefowicz, "Revisiting Distributed Synchronous SGD," *arXiv:1604.00981 [cs]*, Mar. 2017, arXiv: 1604.00981. [Online]. Available: http://arxiv.org/abs/1604.00981

[22] Q. W. Khan, A. N. Khan, A. Rizwan, R. Ahmad, S. Khan, and D. H. Kim, "Decentralized machine learning training: a survey on synchronization, consolidation, and topologies," *IEEE Access*, 2023.

[23] C. Xie, O. Koyejo, and I. Gupta, "Asynchronous Federated Optimization," p. 11.

[24] K. Elsayed, M. A. B. Ibrahim, and H. S. Hamza, "Service discovery in heterogeneous iot environments based on ocf/iotivity," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2019, pp. 1160–1165.

[25] C.-W. Wu, F. J. Lin, C.-H. Wang, and N. Chang, "Onem2m-based iot protocol integration," in *2017 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2017, pp. 252–257.

[26] J. Swetina, G. Lu, P. Jacobs, F. Ennesser, and J. Song, "Toward a standardized common m2m service layer platform: Introduction to onem2m," *IEEE Wireless Communications*, vol. 21, no. 3, pp. 20–26, 2014.

[27] Bigdata Research Center, Jeju National University, Jeju, South Korea, W. Jin, and D. Kim, "Interworking Proxy Based on OCF for Connecting Web Services and IoT Networks," *Journal of Communications*, pp. 192–197, 2020. [Online]. Available: http://www.jocm.us/show-236-1506-1.html

[28] Y. Li, X. Huang, and S. Wang, "Multiple protocols interworking with open connectivity foundation in fog networks," *IEEE Access*, vol. 7, pp. 60 764–60 773, 2019.

[29] J. Esquiagola, L. Costa, P. Calcina, and M. Zuffo, "Enabling CoAP into the swarm: A transparent interception CoAP-HTTP proxy for the Internet of Things," in *2017 Global Internet of Things Summit (GIoTS)*, Jun. 2017, pp. 1–6.

[30] N. A. Tuan, R. Xu, and D. Kim, "Enhanced interoperating mechanism between onem2m and ocf platform based on rules engine and interworking proxy in heterogeneous iot networks," *IEEE Access*, 2023.

[31] B. Mazon-Olivo, D. Hernández-Rojas, J. Maza-Salinas, and A. Pan, "Rules engine and complex event processor in the context of internet of things for precision agriculture," *Computers and Electronics in Agriculture*, vol. 154, pp. 347–360, Nov. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169918302436

[32] C. E. Kaed, I. Khan, A. Van Den Berg, H. Hossayni, and C. Saint-Marcel, "SRE: Semantic Rules Engine for the Industrial Internet-Of-Things Gateways," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 715–724, Feb. 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8091285/

[33] W. Jin, R. Xu, S. Lim, D.-H. Park, C. Park, and D. Kim, "Dynamic inference approach based on rules engine in intelligent edge computing for building environment control," *Sensors*, vol. 21, no. 2, p. 630, 2021.

[34] L. Mainetti, V. Mighali, L. Patrono, and P. Rametta, "A novel rule-based semantic architecture for iot building automation systems," in *2015 23rd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, 2015, pp. 124–131.

[35] L. Lan, R. Shi, B. Wang, L. Zhang, and N. Jiang, "A universal complex event processing mechanism based on edge computing for internet of things real-time monitoring," *IEEE Access*, vol. 7, pp. 101 865–101 878, 2019.

[36] U. Kulshrestha and S. Durbha, "Edge analytics and complex event processing for real time air pollution monitoring and control," in *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2020, pp. 893–896.

[37] F. Paganelli, G. Mylonas, and G. Cuffaro, "A restful rule management framework for internet of things applications," *IEEE Access*, vol. 8, pp. 217 987–218 001, 2020.

[38] X. Luo, Y. Fu, L. Yin, H. Xun, and Y. Li, "A scalable rule engine system for trigger-action application in large-scale iot environment," *Computer Communications*, vol. 177, pp. 220–229, 2021.

[39] S. Choochotkaew, H. Yamaguchi, T. Higashino, M. Shibuya, and T. Hasegawa, "EdgeCEP: Fully-Distributed Complex Event Processing on IoT Edges," in *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, Jun. 2017, pp. 121–129, iSSN: 2325-2944.

[40] Y. Chen and B. Bordbar, "DRESS: A Rule Engine on Spark for Event Stream Processing," in *2016 IEEE/ACM 3rd International Conference on Big Data Computing Applications and Technologies (BDCAT)*, Dec. 2016, pp. 46–51.

[41] R. Xu, W. Jin, A. N. Khan, S. Lim, and D.-H. Kim, "Cooperative swarm learning for distributed cyclic edge intelligent computing," *Internet of Things*, vol. 22, p. 100783, 2023.

[42] N. Gupta, K. Anantharaj, and K. Subramani, "Containerized Architecture for Edge Computing in Smart Home : A consistent architecture for model deployment," in *2020 International Conference on Computer*

Communication and Informatics (ICCCI), Jan. 2020, pp. 1–8, iSSN: 2329-7190.

[43] L. Lan, R. Shi, B. Wang, L. Zhang, and N. Jiang, "A Universal Complex Event Processing Mechanism Based on Edge Computing for Internet of Things Real-Time Monitoring," *IEEE Access*, vol. 7, pp. 101 865–101 878, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8768299/

[44] Y. Zhang and Y. Zhou, "Transparent computing: Spatio-temporal extension on von neumann architecture for cloud services," *Tsinghua Science and Technology*, vol. 18, no. 1, pp. 10–21, 2013.

[45] J. Ren, H. Guo, C. Xu, and Y. Zhang, "Serving at the edge: A scalable iot architecture based on transparent computing," *IEEE Network*, vol. 31, no. 5, pp. 96–105, 2017.

[46] J. Hwang, J. An, A. Aziz, J. Kim, S. Jeong, and J. Song, "Interworking Models of Smart City with Heterogeneous Internet of Things Standards," *IEEE Communications Magazine*, vol. 57, no. 6, pp. 74–79, Jun. 2019, conference Name: IEEE Communications Magazine.

[47] W. Jin and D. Kim, "Development of Virtual Resource Based IoT Proxy for Bridging Heterogeneous Web Services in IoT Networks," *Sensors*, vol. 18, no. 6, p. 1721, Jun. 2018, number: 6 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/18/6/1721

[48] S. Cavalieri, M. G. Salafia, and M. S. Scroppo, "Realising interoperability between opc ua and ocf," *IEEE Access*, vol. 6, pp. 69 342–69 357, 2018.

[49] D. Yacchirema, A. BelsaPellicer, C. Palau, and M. Esteve, "Onem2m based-interworking architecture for heterogeneous devices interoperability in iot," in *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2018, pp. 1–6.

[50] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. PMLR, Apr. 2017, pp. 1273–1282, iSSN: 2640-3498. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html

[51] D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau, "Federated Learning for Keyword Spotting," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 6341–6345, iSSN: 2379-190X.

[52] L. U. Khan, S. R. Pandey, N. H. Tran, W. Saad, Z. Han, M. N. H. Nguyen, and C. S. Hong, "Federated Learning for Edge Networks: Resource Optimization and Incentive Mechanism," *IEEE Communications Magazine*, vol. 58, no. 10, pp. 88–93, Oct. 2020, conference Name: IEEE Communications Magazine.

[53] A. Rizwan, A. N. Khan, R. Ahmad, and D. H. Kim, "Optimal Environment Control Mechanism Based on OCF Connectivity for Efficient Energy Consumption in Greenhouse," *IEEE Internet of Things Journal*, pp. 1–1, 2022, conference Name: IEEE Internet of Things Journal.

[54] D. Schäfer, J. Edinger, S. VanSyckel, J. M. Paluska, and C. Becker, "Tasklets: Overcoming Heterogeneity in Distributed Computing Systems," in *2016 IEEE 36th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, Jun. 2016, pp. 156–161, iSSN: 2332-5666.

[55] A. Rizwan, R. Ahmad, A. N. Khan, R. Xu, and D. H. Kim, "Intelligent digital twin for federated learning in aiot networks," *Internet of Things*, p. 100698, 2023.

[56] S. Zawad, A. Ali, P.-Y. Chen, A. Anwar, Y. Zhou, N. Baracaldo, Y. Tian, and F. Yan, "Curse or Redemption? How Data Heterogeneity Affects the Robustness of Federated Learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, pp. 10 807–10 814, May 2021, number: 12. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17291

[57] P. P. Liang, T. Liu, L. Ziyin, N. B. Allen, R. P. Auerbach, D. Brent, R. Salakhutdinov, and L.-P. Morency, "Think locally, act globally: Federated learning with local and global representations," *arXiv preprint arXiv:2001.01523*, 2020.

[58] X. Xian, X. Wang, J. Ding, and R. Ghanadan, "Assisted learning: A framework for multi-organization learning," *Advances in neural information processing systems*, vol. 33, pp. 14 580–14 591, 2020.

[59] M. Khodak, M.-F. F. Balcan, and A. S. Talwalkar, "Adaptive gradient-based meta-learning methods," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[60] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," *Advances in neural information processing systems*, vol. 30, 2017.

[61] Y. Mansour, M. Mohri, J. Ro, and A. T. Suresh, "Three approaches for personalization with applications to federated learning," *arXiv preprint arXiv:2002.10619*, 2020.

[62] D. Li and J. Wang, "Fedmd: Heterogenous federated learning via model distillation," *arXiv preprint arXiv:1910.03581*, 2019.

[63] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *arXiv:1905.11946 [cs, stat]*, Sep. 2020, arXiv: 1905.11946. [Online]. Available: http://arxiv.org/abs/1905.11946

[64] E. Diao, J. Ding, and V. Tarokh, "HeteroFL: Computation and Communication Efficient Federated Learning for Heterogeneous Clients," Dec. 2021, arXiv:2010.01264 [cs, stat]. [Online]. Available: http://arxiv.org/abs/2010.01264

[65] Y. Bai, K. Liu, and Y. Wang, "Comparative analysis of thermal preference prediction performance in different conditions using ensemble learning models based on ashrae comfort database ii," *Building and Environment*, vol. 223, p. 109462, 2022.

[66] R. F. Rupp, T. Parkinson, J. Kim, J. Toftum, and R. de Dear, "The impact of occupant's thermal sensitivity on adaptive thermal comfort model," *Building and Environment*, vol. 207, p. 108517, 2022.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3308579

Anam Nawaz Khan received the B.S. and M.S. degrees in computer science from COMSATS University Islamabad, Pakistan, in 2016 and 2019, respectively. In the meantime, she joined various reputable academic institutions and later decided to expand her knowledge by continuing her academic career. In 2020, she moved to the Republic of Korea and started working as a Ph.D. Research Fellow with the Department of Computer Engineering, Jeju National University. Her research interests include development and optimization of Artificial Intelligence and machine learning based applications in smart spaces. Furthermore, she actively engaged in numerous projects funded by national institutions and organizations, including ITCR, IITP, NRF and ETRI. She is currently working on integrating the AI enabled IoT Platforms and edge computing to provide autonomous decentralized intelligent services for smart environments.

Atif Rizwan is an accomplished computer scientist and researcher currently pursuing his Ph.D. at Jeju National University, Republic of Korea. He holds a Bachelor's degree from the University of the Punjab, Lahore, and Master's degrees in computer science from COMSATS University Islamabad, Attock Campus. With a strong foundation in mobile and web application development, Atif has seamlessly blended academic excellence with practical industry experience. His research focuses on applied machine learning, data mining, algorithm optimization, Federated Learning, and IoT-based applications, showcasing his commitment to advancing technology's frontiers for the betterment of society.

Rashid Ahmad Dr. Rashid Ahmad is a distinguished scholar who embarked on an academic journey that has taken him across continents. He completed his B.S. degree at the esteemed University of Malakand, Pakistan, followed by his M.S. degree in computer science from the National University of Computer and Emerging Sciences (FAST-NUCES), Islamabad, Pakistan. In 2015, he achieved the pinnacle of his academic pursuits by earning a Ph.D. degree in computer engineering from Jeju National University in South Korea. Presently, Dr. Ahmad holds the esteemed position of a Brainpool Research Fellow at the Bigdata Research Center, Jeju National University, where he contributes his expertise to the cutting-edge field of data intelligence. Prior to this role, he served as an Assistant Professor at COMSATS University Islamabad, Attock Campus.Dr. Ahmad's research is at the forefront of applied machine learning, with a specific focus on developing prediction and optimization algorithms tailored for IoT-based edge intelligence solutions. His extensive expertise extends to areas such as machine learning, distributed machine learning, continual learning, and optimizing deep learning models. Through his dedication and groundbreaking contributions to the field, Dr. Rashid Ahmad is helping to shape the future of data-driven intelligence and machine learning applications.

Wenquan Jin received the B.S. degree in computer science from the Yanbian University of Science and Technology, China, in 2013, and the M.S. and Ph.D. degrees in computer engineering from the Jeju National University, Korea, in 2015 and 2019, respectively. He is currently a Professor with Yanbian University, Republic of China since Sep 2022. His research work focused on integrating the IoT and Edge Computing with intelligent services for smart spaces. He studies standard specifications and communication protocols such as OCF, OMA DM, LWM2M, IoT-A, oneM2M, CoAP, HTTP and MQTT. Also, he studies in the area of optimization and Machine Learning for the smart homes and energy optimization based on Particle Swarm Optimization, Deep Neural Network and Recurrent Neural Network. Moreover, he participated in multiple projects from national fund and institutes such as ITCR, IITP and ETRI.

Qazi Waqas Khan is currently Pursing PhD degree in Computer Engineering at Jeju National University, South Korea. He received his MS in Computer Science and MCS (2 years) from COMSATS University Islamabad, Attock Campus, Punjab, Pakistan, in 2022 and 2019, respectively. In addition, he received a Bachelor of Science (BSC) degree in Statistics and Computer Science from the University of the Punjab, Lahore, Pakistan, in 2017. He was awarded Merit scholarship in MS(CS). He was also awarded PEEF scholarship in BSC and MCS degree. His research focused on applied Machine Learning, Federated Learning, Data Mining, Natural Language Processing, Computer Vision, and Internet of Things.

Sunhwan Lim is an esteemed researcher associated with the Autonomous IoT Research Section at the Electronics and Telecommunications Research Institute (ETRI). With a profound passion for cutting-edge technology, their research portfolio encompasses a diverse range of fields, making significant contributions in the realms of Artificial Intelligence, data privacy, electrocardiography, groupware, signal processing, blockchain technology, collaborative filtering, smart contracts, convolution algorithms, efficient data handling, and the preservation of data integrity. His work is not only shaping the present landscape but also paving the way for a future enriched by innovative solutions that merge the realms of technology and human experience

DoHyeun Kim received the B.S. degree in electronics engineering from Kyungpook National University, South Korea, in 1988, and the M.S.and Ph.D. degrees in information telecommunication from Kyungpook National University, South Korea, in 1990 and 2000, respectively. He was with the Agency of Defense Development (ADD),from 1990 to 1995. Since 2004, he has been with Jeju National University, South Korea, where he is currently a Professor with the Department of Computer Engineering. From 2008 to 2009, he was a Visiting Researcher with the Queensland University of Technology, Australia. His research interests include sensor networks, M2M/IOT, energy optimization and prediction, intelligent service, and mobile computing.