Internet of Things 19 (2022) 100572

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Research article

FogAI: An AI-supported fog controller for Next Generation IoT

İbrahim Kök [a], Feyza Yıldırım Okay [b,*], Suat Özdemir [c]

[a] Department of Computer Engineering, Pamukkale University, Denizli, Turkey
[b] Department of Computer Engineering, Gazi University, Ankara, Turkey
[c] Department of Computer Engineering, Hacettepe University, Ankara, Turkey

A R T I C L E I N F O

Keywords:
FogAI
Fog computing
Artificial intelligence
Deep Q-learning
Task offloading
IoT
NGIoT

A B S T R A C T

In this paper, we present a novel artificial intelligence-based fog controller, called FogAI that provides a versatile control mechanism to the fog layer. FogAI not only abstracts the control mechanism from the fog environment but also offers potential solutions for the problems of fog-based Next Generation Internet of Things (NGIoT) systems. To this end, we first present a comprehensive examination of challenging issues in Fog Computing (FC). Then, we outline possible FogAI based solutions to these challenges from different perspectives. To illustrate the feasibility of our FogAI concept, we design a use case scenario for task offloading problem in FC. Then, we propose a Deep Q-Learning (DQL) algorithm that autonomously performs task offloading in delay-sensitive and computationally-intensive IoT applications and test it on FogAI. The results show that the proposed FogAI-assisted DQL algorithm is superior to existing offloading policies.

1. Introduction

The Internet of Things (IoT) is a prominent technology that consists of a great number of ubiquitous and heterogeneous smart devices providing real-time interconnection and intercommunication with each other. With the widespread usage of IoT applications, it is becoming a part of our lives and the number of smart devices increases sharply [1]. According to Cisco [2], this number is expected to reach up to 29.3 billion by 2023. On the other hand, this dramatic rise creates new problems regarding the storage and processing needs of the data collected by these devices.

Over the past decades, cloud computing has been accepted as an adequate solution for IoT requirements with its rich computational power and storage capacity. However, due to the rapid increase and diversification of IoT data, traditional cloud computing has started to fall behind the technology in supporting all the data communication, processing, and storage needs of IoT data [3]. In cloud-based IoT systems, transmitting a huge amount of data to a centralized server may lead to undesirable consequences such as bandwidth bottleneck, congestion, high latency, and QoS degradation [4]. Fog computing (FC) is introduced to provide innovative and promising solutions to data and quality of service (QoS) related problems of IoT systems. FC is a complementary technology for cloud computing in order to lighten the burden of cloud computing by moving the services closer to the end systems. Instead of sending whole data to long-distance cloud servers, FC distributes cloud services and allows local data processing and storage. Due to its geographically distributed nature and proximity to the IoT devices, FC provides significant efficiency by decreasing the amount and time for data transmission. Therefore, it ensures end-to-end latency requirements of IoT applications [5]. It also offers different capabilities to cloud computing such as high mobility, location-awareness, and temporal

* Corresponding author.
E-mail addresses: ikok@pau.edu.tr (İ. Kök), feyzaokay@gazi.edu.tr (F.Y. Okay), ozdemir@cs.hacettepe.edu.tr (S. Özdemir).

https://doi.org/10.1016/j.iot.2022.100572
Received 28 February 2022; Received in revised form 21 May 2022; Accepted 3 July 2022
Available online 8 July 2022
2542-6605/

storage. With the help of its key features, FC is highly recommended for delay-sensitive IoT applications such as augmented reality, vehicular networks, smart traffic lights [6].

Fog and cloud computing concepts are not substitutes but complements of each other. In this fog–cloud architecture, fog layer resides in the middle of physical and cloud layers. Fog layer is responsible for communicating physical layer and preprocessing the IoT data before transmitting to the cloud layer for further analysis. Depending on the number of IoT devices in the system, the amount of data produced and transmitted to the related servers may increase exponentially [7]. FC allows network structure to expand horizontally so that delay and mobility related QoS requirements can be satisfied.

Although FC provides additional advantages over cloud computing, several challenges arise from the rapid increase in network size and the distributed nature of FC. Fog servers become inadequate to manage the network that produces many types of IoT data at a tremendous pace [8]. Therefore, a more intelligent system is needed for better network interoperability, effective handling of heterogeneity, making fast and optimized offloading decisions, and elimination of possible security and privacy threats [9]. On the other hand, IoT is evolving towards the Next Generation IoT (NGIoT). Different from traditional IoT networks, NGIoT networks bring together hyperconnectivity, edge computing, Artificial Intelligence (AI), autonomous system, and blockchain technologies. These NGIoT systems with embedded intelligence, hyperconnectivity and AI capabilities promise smart solutions by shifting data processing and network management to distributed edge devices [10].

Managing fog servers are studied in the traditional IoT networks as well. The existing studies in the literature generally present control systems to manage underlying networks for different purposes, using virtual cloud controllers, virtual fog controllers, or external SDN controllers located in the cloud or fog layers. Abdelaal et al. [11] propose event based control systems as a service from cloud layer in the aim of decreasing resource utilization for large-scale systems. Guan et al. [12] focus on simplifying the uncertainties of cloud control systems by decomposing and proposing a new controller design for cloud control systems. Inaltekin et al. [13] investigate the optimal placement of virtual control services in the fog continuum. They aim to balance the latency and reliability aspects by determining the locations of virtual controllers. Yanuzzi et al. [14] propose virtualized control functions for smart cities in a smart cabinet, and Al et al. [15] propose their fog based control systems for smart home with a control panel placed in a home. On the other hand, SDN-based cloud/fog systems have been gained too much attraction due to the controlling and managing capabilities of SDN. Alomari [16] provides a comprehensive survey of SDN-based cloud and fog systems for efficient resource management. Task offloading [17], load balancing [18], and security [19] are some of the other aspects where SDN offers a manageable and controllable system for cloud and fog systems.

Apart from the above-mentioned studies, this study focuses on a more generalized controller system model covering the fog continuum and offers AI-based solutions to the most FC challenges. More specifically, this paper proposes a novel controller called FogAI for handling NGIoT specific issues in the fog layer. In FogAI, logical and operational decisions can be made dynamically. Thus, FogAI provides versatile control and managing, flexibility, and seamless integration among physical-fog–cloud layers. In the next section, we present a detailed explanation on why our proposed controller system is important and how it is differentiated from existing studies.

Although there are some studies in the literature covering fog-based challenges and solutions [20–24], we differentiate from the existing studies by offering AI-based solutions to the fog-based challenges in NGIoT applications. Overall contributions of this paper can be listed as follows:

- We highlight the importance of FC for NGIoT applications and comprehensively identify the research challenges that still need to be addressed in FC.
- We propose a novel AI-supported fog controller called FogAI to provide AI-based functionalities demanded by fog-based NGIoT applications
- We propose FogAI based potential solutions and mechanisms for possible FC challenges.
- To verify our FogAI concept, we also propose a Deep Q-Learning (DQL) based task offloading algorithm that maximizes the control and management of the system resources while also reducing task processing delay, task transmission delay, and total system delay.

The rest of the paper is organized as follows. Section 2 explains why we need FogAI. Section 3 summarizes FC challenges. In Section 4, we give the details of the proposed FogAI layer. Section 5 offers potential FogAI solutions for NGIoT challenges. In Section 6, we propose a FogAI based solution to the task offloading problem. Section 7 presents our simulation results and finally concluding remarks are given in Section 8.

2. Motivation

To facilitate the understanding of FogAI structure and to highlight its necessity, we provide explanations of the following critical questions.

(1) Why a New Fog Controller Concept? IoT technology and its applications have evolved quickly in recent years and the underlying infrastructure needs to evolve with it, in order to support the emerging needs such as rapid scaling, heterogeneity, interoperability, agility, security, and privacy issues. A new layer has required for alleviating the management challenges of fog servers and preventing a possible bottleneck of service provisioning. FogAI is a novel controller concept that extends FC to help in controlling and managing the network. FogAI concept is similar to SDN, but it extends the SDN capabilities. Although SDN is an emerging technology, it still suffers from centralized management of the decentralized structure of FC, for example, a single point of failure, reconfiguration cost in a dynamic network environment, scalability and latency [25–27]. However, FogAI can eliminate the disadvantages of SDN controller with its distributed fog architecture and advanced AI support. In addition, it offers promising solutions to many FC problems such as resource management, heterogeneity, mobility, federation and interoperability. In these aspects, it offers a broader concept than the SDN controller concept.

(2) Why an AI-Supported Controller? Communication, Computing, Caching, and Control (4Cs) problems are the main obstacles to fully enabling collaboration in Fog-based IoT networks. In particular, solving the communication problems brought by the heterogeneous and complex IoT network, and ensuring the flexible processing on edge/fog servers by dividing the computational tasks into sub-tasks are challenging. Moreover, content and interest-based data caching are essential for context-aware and delay-sensitive IoT applications [28]. At this point, there is a need for advanced infrastructure components that serve to create interest and context-aware systems. Furthermore, it is hard to achieve autonomous control of the IoT network by making real-time decisions by adapting to a dynamic network environment. To solve these problems, different methods based on heuristic algorithms, traditional machine learning, and deep learning are proposed in the literature. Based on these methods, it is seen that advanced AI algorithms provide superior success in solving current IoT network problems [29]. For this reason, we introduce an AI-supported controller (FogAI) as a promising novel structure in solving 4Cs problems in NGIoT applications.

(3) What is FogAI Good For? FogAI is an extension of fog layer as a new AI-supported controller. It helps FC manage heterogeneous and pervasive IoT devices by offering additional capabilities with its agent, modules, algorithms, and tools. FogAI provides potential solutions for the different challenges explained in Section 5.

3. Fog computing and challenges

FC is an innovative approach that extends the processing, storage, and computation capabilities of cloud computing. Since FC is a complementary solution to cloud computing, they cooperate with each other and this cooperation creates fog–cloud collaboration [30]. FC is generally three-layer where fog layer acts as a bridge and provides a seamless connection between physical and cloud layers [30]. During the transmission of data, each layer takes over different responsibilities that are explained in detail below:

- Physical Layer: This layer is located closest to the end-users. It consists of various types of static and mobile IoT devices such as mobile phones, smart vehicles, sensors, and actuators. They sense the environment and produce very heterogeneous and ubiquitous data due to the distributed deployment through the area.
- Fog Layer: This layer is located at the edge of the network and connects the physical and cloud layers. It consists of servers, routers, gateways, switches, and base stations. It provides quick and distributed processing over a wireless connection like WiFi, 3G, 4G. It also ensures temporal storage in such cases where instant responses are demanded like real-time disaster management [31] or traffic management [32]. Hence, the queries can be processed immediately at local servers without sending them to a remote server.
- Cloud Layer: This layer supports extensive data analysis which requires more powerful computational and storage capacities. After preprocessing the data in fog layer, the data is sent to the cloud layer over the Internet. Cloud layer enables permanent storage for further analysis of the huge amount of data.

FC is a decentralized structure in which distributed fog servers aim to overcome the design problems caused by the centralized structure of cloud computing. Although FC technology is still in its early stages, it has gained too much attention from academics and practitioners for different purposes [33,34]. As interests in FC increases, the application areas of fog-based systems diversify, thereby increasing the fog-related problems. While most of these problems are more fundamental, some of them may be domain-specific like smart grids and vehicular networks, or problem-specific like scheduling and offloading. The fundamental challenges [35–42] can be listed as follows:

- **Federation and Interoperability:** Since fog servers are geographically distributed, these servers can be operated by different cloud or fog service providers. In the fog–cloud collaboration, there is a need for different technologies, modules, and algorithms are required to operate the system by interconnecting with each other [35].
- **Heterogeneity:** Since fog layer poses a decentralized structure, each fog node residing in this layer may differ in their storage and computing capacities by considering the demands of the area they are deployed or the resource limits of those fog nodes. Heterogeneity should be taken into consideration in the task assigned to each fog server.
- **Scalability:** Fog–cloud architecture is a highly dynamic structure in nature where several IoT and fog nodes can join or leave the network. As the number of IoT devices increases, fog servers should scale up and cover these large number of IoT devices to remain operational. On the contrary, some IoT nodes can fail due to limited resources, and accordingly, the network structure should scale down itself [36].
- **Load balancing and Resource Allocation:** Fog servers may have limited memory and low computational resource to compute the tremendous amount of data received from physical layer. When the capacity of a specific fog server becomes high, it has to distribute its load to an available fog server so as not to fail. A mechanism is required to control the current status of active servers. When the capacity of a server exceeds a critical threshold, this mechanism should determine the optimal neighbor server to offload its tasks. In addition, when deploying the resource-limited fog nodes in a large-scale environment, resource fragmentation can be a serious problem leading to QoS degradation [37]. By considering the user requirements, a pool of resources should be provisioned and released to a pool of tasks with minimum power consumption.

- **Node Placement:** Fog servers are expected to cover a specific area to manage and control the underlying layer effectively. Therefore, they should be placed where the resources are in order to take advantage of FC's proximity. However, it is not always possible in situations where they are exposed to some physical threats and/or vulnerabilities. Considering the requirements of the application, determining the optimal number of servers and their locations forms an optimization problem to be solved efficiently.
- **Mobility:** FC supports mobility. Hence, in addition to IoT devices, fog servers can be static or mobile as well. When a mobile device in the IoT layer, such as a mobile phone, moves out of range, it must transmit its data to the related fog server. It should be decided how to make the transmission by taking into account critical metrics such as path length, bandwidth, and security [38].
- **Data, Task, and Computation Offloading:** Since IoT nodes consist of resource-poor devices, they tend to fail in coping with such a huge workload and need to offload their tasks to more resourceful devices like fog servers [39,43]. The most appropriate fog server should be selected by evaluating the current information of all fog servers in the fog layer. It should also be decided on how to transmit the data or task to the desired server.
- **Policy Agreement:** Fog servers are deployed in a large-scale area and customized for different services such as healthcare or environmental monitoring. These different tasks can be managed by different service providers. Besides, the cloud operator can be different as well. All operators in the fog–cloud architecture should collaborate to provide seamless integration of layers. Proper Service level agreement (SLA) management techniques should be utilized to provide acceptable QoS in a highly dynamic fog system [40].
- **Collaboration with Emerging Technologies:** Software-Defined Networking (SDN), Network Function Virtualization (NFV), Information-Centric Networking (ICN), and Tactile Internet are some of the auxiliary technologies for FC in facilitating network management and control; however, new challenges are expected to arise during the integration of these technologies. The pros and cons of these technologies should be considered carefully, and smooth integration should be provided.
- **Security and Privacy:** Security and privacy challenges are commonly studied for FC under different topics such as authentication, trust, and malicious attacks [41] Fog nodes are generally physically open and vulnerable to security and privacy threats. Since they are deployed in wide-range unreliable regions, usually they cannot be protected by strict surveillance or protection mechanisms [41,44]. Although authentication provides an initial set of relations among IoT and fog nodes, it is not generally enough to secure the system. IoT and fog nodes can be compromised after joining the network. These compromised nodes use detailed information for their own benefit and break the privacy of the system. An effective privacy-preserving model should be used to avoid compromising and revealing detailed information of users. Besides, fog and cloud nodes can be operated by different operators who may not be fully trusted. Therefore, a robust management model should be employed to verify that all nodes participating in the network provide a certain level of trust [45]. In addition, FC usually deploys its service to delay-sensitive applications where real-time processing is important. This makes the fog system much more fragile against Denial of Service (DoS) attacks [42].

4. FogAI: AI-supported fog controller

We present the graphical representation of the proposed FogAI concept and the positions of FogAI components in the system architecture in Fig. 1. Our three-tier IoT system architecture consists of a physical device layer, fog layer controlled by FogAI, and cloud layer. There are heterogeneous IoT devices in the physical layer as the data producers and data consumers. Data producers naturally generate a wide variety of data types such as text feeds, audio feeds, video streams, sensory data, and so on. Data consumers can be static or mobile device applications that request the processed data by fog servers or cloud servers. The processing and analysis of generated data require different hardware requirements [46].

For this reason, fog servers are responsible for processing a wide variety of data with different specifications in the fog layer. On the other hand, these fog servers require more intelligent mechanisms for management, scalability, interoperability, and security issues. Herein, we propose a new virtual controller called ***FogAI*** to manage, control and monitor the highly heterogeneous and dynamic fog network in terms of data and resource management, functionality, and system performance. FogAI controller has various AI-based components such as agent, Deep Learning (DL) models, engines, modules, algorithms, and other tools, which offer promising solutions to the challenging issues discussed previously. To better understand these components and their structures, we briefly clarify each of them below.

- *Agents:* Agents are some of the basic elements of Reinforcement Learning (RL). They observe the IoT network environment and produce policies that allow the designed system to be adaptively and dynamically managed in line with certain goals.
- *AI Models and Algorithms:* These components refer to DL models and algorithms developed to perform one or more of the system functions. DL models can be evolved for a wide range of IoT applications from simple sensor data analysis applications to comprehensive big data analysis applications.
- *Modules/Interfaces:* These components refer to software elements that can perform one or more tasks in a network environment. They can ensure that applications run in an effective way according to the system topology.
- *Engines and Tools:* These components provide flexibility to the system by scaling up/down the system resources. Also, they simplify the management of fog servers and make system decisions more agile.

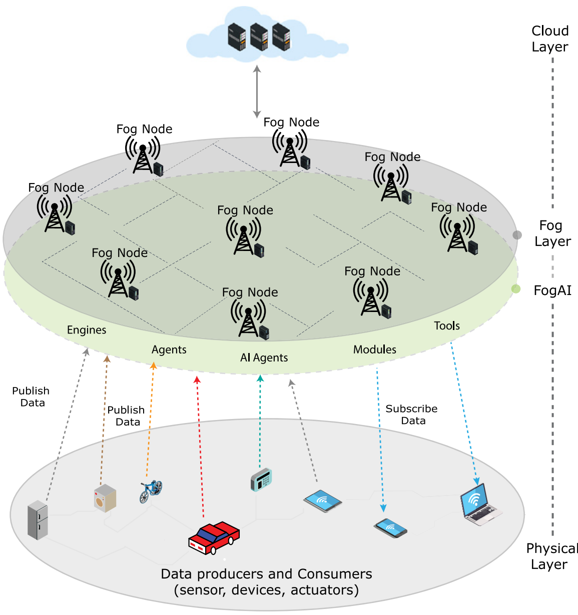

Fig. 1. System infrastructure and graphical representation of FogAI.

4.1. Network model and formulation

As shown in Fig. 1, we present the potential network model of FogAI structure over the IoT architecture. We also formulate the network in accordance with the task offloading problem addressed in the study. In network model, set of $F = \{f_1, f_2, \dots ..., f_m\}$ and $S = \{s_1, s_2, \dots ..., s_m\}$ represent the fog devices and smart devices/sensors, respectively. m denotes device count.

$L_{sf} = \{l_{s_1,f_1}, l_{s_2,f_1}, l_{s_3,f_1}, \dots ..., l_{s_i,f_j}\}$ and $L_{fc} = \{l_{f_1,c}, l_{f_2,c}, \dots ..., l_{f_i,c}\}$ are communication link sets between the IoT device (e), fog (f) and cloud (c) layers. Based on the notations above, we formulate task processing, task offloading and total system delays to evaluate proposed FogAI controller.

Here, S_{m_t} is the total amount of task from smart devices at time t.

$$S_{m_t} = \sum_{i=1}^{m} s_{i_t} \qquad (0 < s_i) \tag{1}$$

Here, we define task processing delay of FogAI component as follows.

$$T^{proc}_{FogAI\,Agent} = \frac{S_{m_t}}{f^{cc}} \tag{2}$$

where f^{cc} is the computing capability of FogAI component.

In addition, B^{up} and B^{down} denote the total uplink bandwidth and downlink bandwidth respectively. S^{up} and s^{down} refer to the number of tasks transmitted up and down according to the link direction. Herein, uplink and downlink task transmission delays can be given as shown in Eqs. (3) and (4).

$$T^{up}_{l_{ef}/l_{fc}} = \frac{S^{up}}{B^{up}} \tag{3}$$

$$T^{down}_{l_{cf}/l_{fe}} = \frac{S^{down}}{B^{down}} \tag{4}$$

Thus, we obtain the total end-to-end task transmission delay (T^{trans}_{e2e}) between the task sender and receiver by summing all the uplink and downlink transmission delays.

$$T^{trans}_{e2e} = T^{up}_{l_{e,f}} + T^{up}_{l_{f,c}} + T^{down}_{l_{e,f}} + T^{down}_{l_{f,c}} \tag{5}$$

Finally, the total system delay for N tasks transmitted to the network can be given as follows.

$$T^{system}_{total} = \sum_{n=1}^{N} (T^{proc}_{FogAI\,Agent} + T^{trans}_{e2e})$$

5. Potential FogAI based solutions for NGIoT challenges

In this section, we provide potential FogAI based solutions to the FC challenges through illustrative scenario cases. A summary of these potential solutions are presented in Table 1.

5.0.1. FogAI based solutions for federation and interoperability

FogAI based agents and DL models can cooperate with the different providers to ensure the coordination between application components. For example, Actor–Critic based federated RL agents can be used to control and manage multiple fog servers or edge devices [47]. RL-based multi-agents can be used to interpret certain ontology concepts and provide an exact meaning to a message sent by system components. At the same time, these agents can ensure the interoperability of protocols in an environment where heterogeneous communication protocols such as MQTT, COAP, HTTP, SMTP, and AMQP exist [48]. Besides, Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) based DL models can be deployed in FogAI to detect anomalous behavior of IoT devices by classifying malicious events [49].

5.0.2. FogAI based solutions for heterogeneity

The network heterogeneity raises problems such as mutual interference, differentiated QoS provisioning, and resource allocation [29].

To solve heterogeneity problems, FogAI agents and DL models can be employed. For example, in Vehicular Ad hoc NETworks (VANETs) vehicles can make various types of requests such as multimedia requests, internet requests, emergency notifications, and content caching [50]. An RL agent can be developed that can effectively respond to these requests according to the status of the network.

In other potential solutions, RNN based DL model and modules can be used to identify and track objects such as cars, buses, pedestrians, and cyclists in an application of autonomous vehicle driving [51]. Also, LSTM based model can proactively predict mobility of vehicles for content delivery and caching. Therefore, the requested contents can be transferred, cached to the RSU according to the direction of the vehicles, and stored [52].

5.0.3. FogAI based solutions for scalability

In a network system, scalability can have many dimensions such as operational environment, capacity, performance, reliability, and security [53]. From an architectural point of view, two types of scalability can be mentioned. Horizontal scalability is related to increased devices and software expansion of components, while vertical scalability is related to their ability to increase efficiency [54]. From a scalability perspective, FogAI based agents, models, and algorithms can offer effective solutions. For example, in smart transportation applications, vehicle mobility and speed patterns can be predicted via DL models and algorithms in FogAI.

5.0.4. FogAI based solutions for load balancing and resource allocation

Due to the dynamicity, heterogeneity, and uncertainty of fog environment, load balancing, and resource management systems are essential to make the workloads distributed in a balanced manner [55]. Load balancer and resource allocator agents can be used together in highly dynamic, heterogeneous, and uncertain areas like smart healthcare. In this application, patient data and server loads in the network are monitored continuously, and this data can be distributed equally among the existing servers by responsible agents [56]. On the other hand, Deep Belief Network (DBN) based DL model can be directed to suitable servers by analyzing a large amount of user data and network load [57].

5.0.5. FogAI based solutions for node placement

FogAI based agents or models can be employed. For instance, an efficient FogAI based server, service, and task placement models/agents can be used to optimize service decentralization on FC landscape [58]. In this optimization process, agents can take into account parameters such as location, computing, and storage capacities of fog devices.

5.0.6. FogAI based solutions for mobility

In IoT system design, devices and fog nodes can be mobile [36]. Thus, engines and modules that handle this mobility are needed to ensure the continuity of services [59]. Herein, FogAI based mobility engines can be a solution to this challenge. In addition, an SDN controller can be located to FogAI, and make the fog and physical layers are more manageable and flexible by decoupling the network as data and control planes. Thus, in applications with high mobility, even if a mobile IoT device moves out its coverage range, it can easily transmit its data to its own server efficiently due to the global point of view of the SDN controller in FogAI.

5.0.7. FogAI based solutions for data, task and computation offloading

With limited computation, memory, and power capacities, IoT devices cannot handle such a huge workload [60]. Therefore, there is a need for components to meet current challenges in terms of data, task, and computing. Herein, system needs can be met by RL agents, DL models, and algorithms that take into account one or more of these challenges. For instance, an RL agent can develop a task offloading policy to minimize energy consumption, task delay, and task loss. In this way, it can be decided that the task will be run in the cloud or locally [61].

Table 1
FC challenges and potential FogAI based solutions.

FC challenges	Possible FogAI components	Potential FogAI based solutions	NGIoT application domain
Federation and Interoperability	RL agent	Federated agents can be used to control, manage and coordinate multiple fog or edge devices.	Network optimization
	RL agent	Agent can interpret specific ontology concepts and provide an exact meaning to messages sent by system components. Also, they can ensure interoperability between the protocols such as HTTP, MQTT and AMQP.	Image recognition
	RNN	RNNs can classify data traffic with high accuracy in detection malicious behavior.	Intrusion detection
Heterogeneity	RL agent	Agent can provide allocation policy by considering networking, caching, and computing in VANETs.	VANETs
	CNN, Module	DL models and modules can be used to identify and track objects such as cars, buses, pedestrians, and cyclists.	Autonomous vehicle driving
	LSTM	LSTM module can predict the mobility movement and select and appropriate RSU in fog layer to cache contents.	VANETs
Scalability	RNN, LSTM	FogAI based DL models and algorithms can predict the mobility and speed patterns of vehicles. Also, traffic congestion can be reduced by regulating the traffic flow according to the predicted values.	Intelligent transportation system
Load balancing and Resource Allocation	RL agent	Agent can allocate resources in VANETs by taking into account the computing, caching and networking capabilities of fog servers.	VANETs
	RL agent	Agent can distribute data traffic in a balanced manner by simultaneously considering patient data and server loads.	Smart healthcare
Node Placement	RL agent, DL models	Models/agents can be used to optimize service decentralization by considering parameters such as location, computing and storage capacities of fog servers.	Network optimization
Mobility	Engines	Engines can make the fog and physical layers are more manageable and flexible by handling the mobility of vehicles.	VANETs
Data, Task, and Computation Offloading	RL agent	Agent can take an optimal offloading action decision to minimize the energy consumption, processing delay, and task loss probability.	Network optimization
Policy Agreement	DL models	DL models can proactively identify SLA violations and change the network state to avoid that violation.	Network management
Emerging Technologies	Interface, Engine	Tactile–human interfaces/Engines can interact human senses with machines and technologies. Enabling tactile interaction can make it possible to detect, manipulate, and position objects in a virtual environment.	Telesurgery, Vehicle platoons, and Augmented reality
	RL agent	FogAI agent can cooperate with SDN controller to control the network more effectively.	Network management
Security and Privacy	DL models	DL models can be used for security issues such as threat identification, false data injection, and intrusion detection.	Security systems
	RL agent	Agent can maintain connectivity with dynamic decisions by taking into account position, speed, and connectivity of vehicles on the air/ground.	Multi-Robot systems

5.0.8. FogAI based solutions for policy agreement

FogAI based SLA management systems can be a potential solution for the fog–cloud architecture. FogAI controller may consist of different providers that offer options such as different billing and metering methods, different scalability, and elasticity procedures.

5.0.9. FogAI based solutions for emerging technologies

SDN technology can be employed for content-based forwarding, but it suffers from a single point of failure. In addition, an SDN controller can have disadvantages such as centralizing the decentralized structure of the fog computation, reconfiguration cost in the dynamic network environment, and latency. A FogAI controller can cooperate with SDN controller and provide backup solutions for SDN in case of any controller failure. Moreover, the FogAI can eliminate the disadvantages of the SDN controller with its distributed fog architecture and advanced AI support. Also, ICN models can be implemented by FogAI components for data management at the gateway level. Given that data delivery requires the new ICN paradigm, an agent can decide where to process the data by considering incoming content and fog server status. From a content delivery perspective, Tactile Internet can offer skill sets over the network in addition to the content delivery in potential applications such as telesurgery, telerehabilitation, toolkits, and augmented reality.

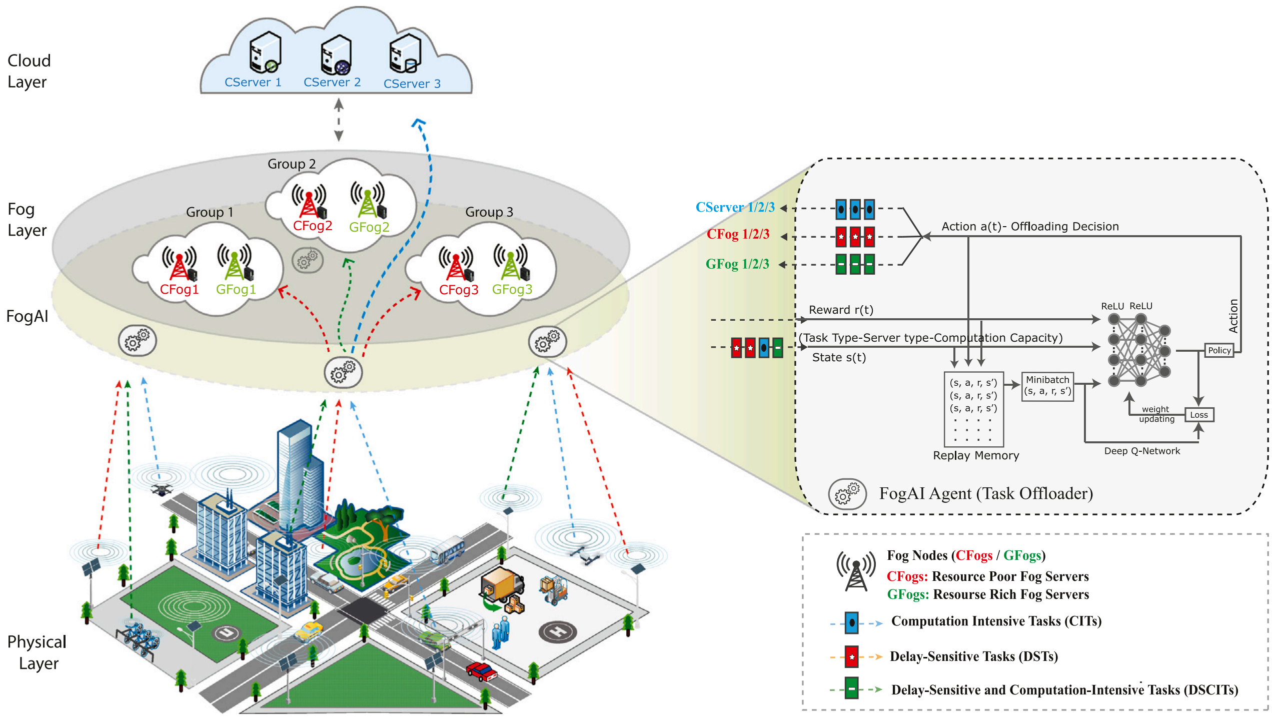

Fig. 2. Proposed FogAI based system model and illustration of the case study.

5.0.10. *FogAI based solutions for security and privacy*

RL agents and DL models can be used to ensure network security and privacy issues. For example, multi-robot systems such as UAVs, drones, and autonomous cars often face connectivity preservation problems due to the dynamic and complex environment. To overcome this problem, DRL agents can maintain the connection with dynamic decisions by considering their position, speed, and connectivity. On the other hand, DBN-based DL models can be used for security threat identification and false data injection attacks. Autoencoders can be used for intrusion detection and machine fault diagnosis.

6. Case study: Deep reinforcement learning based autonomous task offloading in FogAI

In this section, we describe a case study to demonstrate the application of task offloading via DRL agent in FogAI.

6.1. *Proposed FogAI based system model*

As shown in Fig. 2, the proposed system model consists of three layers: physical layer, fog layer with FogAI, and cloud layer. The physical layer includes different devices that generate heterogeneous data and tasks. As seen in studies [62,63], tasks can be grouped as Delay-Sensitive (DST), Computation-Intensive (CIT), and both Delay-Sensitive and Computational-Intensive (DSCIT) in real network environments. Therefore, in this paper, the tasks are divided into these three groups.

All devices request the nearest fog servers to handle their tasks to meet QoS and user requirements effectively. Fog layer has a dynamic and heterogeneous structural design. Therefore, it consists of multiple resource-poor and resource-rich servers. Here, resource-poor and resource-rich servers are named CPU-assisted fog (CFog) and GPU-assisted fog (GFog), respectively.

FogAI contains the DQL based agent that offloads the tasks to the appropriate servers by considering the task type, server type, and computing capacities of the servers. The agent periodically makes a task offloading policy at the beginning of each task offloading round. The task offloading policy includes both task offloading decisions (where to offload) and appropriate fog or cloud server allocation. The cloud layer consists of data centers with multiple servers that are capable of sufficient storage and computational resources.

6.2. *Deep Q-learning based task offloading algorithm*

In this case study, we consider carrying out a balanced task offloading by distributing tasks to appropriate servers. In this way, we aim to increase QoS by reducing task transmission delay, task processing delay, and total system delay. For this purpose, we propose the DQL based task offloading algorithm that can offload a task to the appropriate fog server or a nearby cloud server by considering the task type, server type, and server computing capabilities. In our algorithm, we use a fully connected Deep Neural Network (DNN) that has two hidden layers (108 × 108 neurons) and Rectified Linear Unit (ReLU) activation function. To get these values, we tested the DNN algorithm under the different numbers of neurons (32, 54, 64, 108, 128), and batch sizes (27, 45, 81) in the training process. We also set up experience replay memory size 5.10^5, the maximum episode is set 50, and the maximum number of steps in each episode is set 405, which is the number of total tasks used in the simulation. The learning rate α is fixed 10^{-3}, discount factor $\gamma = 0.95$. The e-greedy algorithm is used with random $\epsilon = 0.1$.

To make the proposed algorithm feasible in our system, we define the core system elements as follows:

- *FogAI Agent:* It refers to the task offloader that offloads the tasks according to the proposed DQL algorithm.
- *Environment:* It refers to our system model where FogAI agent interacts with.
- *State:* It is represented as a vector that includes the task types, computation capacities of the fog, and cloud servers at time slot t.
- *Action:* In our setting, a task can be offloaded to either the fog or cloud servers; thus, we define the action as a $1 \times n$ vector where n represents the total number of servers in the system. In the vector, the server to which the task will be transferred is set to 1, while the others to 0.
- *Reward:* In each time slot t, the FogAI agent receives an immediate reward from the environment after its action. If the agent offloads a task successfully to the appropriate fog or cloud server, it gets a positive reward of +1, otherwise a negative reward of −1.

7. Simulation results and evaluation

7.1. Simulation settings

We conduct our simulations using Tensorflow and Python 3.6 on a PC equipped with Intel i7-6850K CPU, 16 GB RAM, and NVIDIA GeForce GTX 1080Ti GPU. We assume that devices have three task types (DST, CIT, and DSCIT) requests to be processed on cloud or fog servers. In determining these three task types, we have considered the task types in papers [62,63]. Since all task types require different processing capabilities, we define three fog groups that contain both GPU-assisted (colored Green) and CPU-assisted (colored Red) fog servers. We also define one cloud server group with three servers (Server 1/2/3 - colored Green). As DSTs are delay-sensitive, these tasks need to be processed on CFog servers. Thus, these tasks can be performed without waiting for offloading, and congestion delay. Besides, it is assumed that DSCITs can be processed on resource-rich GFog servers. Processing these tasks in the fog layer is required in terms of reducing latency, while resource-rich servers are computationally required. Note that when a task is offloaded to the cloud server, it increases delays. Therefore, it is assumed that only CIT tasks are transferred to cloud servers.

For experiments, we generated two different balanced task datasets, each containing 405 tasks by using the discrete uniform distribution. In this way, we ensured that the data sets have an equal number of samples (135 tasks) from each task type. We used one of the data sets in training and the other in the testing phase.

7.2. Performance evaluation

We evaluate the performance of our DQL based algorithm in terms of total system reward, task workload, task computation delay, task transmission delay, and total system delay. For this purpose, we compare the proposed algorithm with the three commonly used offloading algorithms in the literature [64–66]. These algorithms offload the tasks according to different policies. These policies are explained as follows.

(1) Proposed FogAI Agent Offloading Policy (FAOP): This represents the task offloading policy of the proposed DQL based agent located in FogAI. FAOP jointly considers the incoming task type, server type, and the computation capacity of the servers in task offloading.

(2) Task-centric Offloading Policy (TOP): This policy does not consider the computation capacity of the servers while considering the task type and server type in task offloading. It randomly sends the task to one of the servers of its type.

(3) Random Offloading Policy (ROP): This policy randomly offloads the incoming tasks to one of the fog or cloud servers. It does not consider the task type, server type, and computation capacities.

(4) CoMputation-centric Offloading Policy (CMOP): This policy only considers available computation capacity of the servers in task offloading. Note that the available computation capacity is determined by taking into account the current load of the servers. On the other hand, this policy does not consider task and server types.

Fig. 4 illustrates the average total rewards obtained by the offloading policies learned in each training episode. From Fig. 4, we can see that FAOP increased the system reward from −152 to +384 thanks to its learning capacity.

On the other hand, ROP, TOP and CMOP policies collected the system rewards in the (min/max) range of $-356/-294$, $-202/-94$, and $-166/-94$, respectively. We obtained the average system rewards of FAOP, ROP, TOP and CMOP policies as $+322, -317, -136, -134$, respectively. It is clearly seen from the results that FAOP is significantly more successful than the other three policies. This is because FAOP constantly updates its offloading policy based on the feedback (rewards) received from the environment. It is also the ability to use past experiences in policy-making.

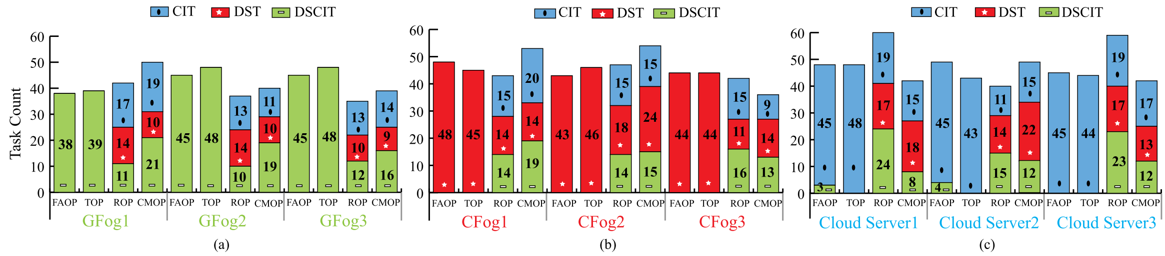

Fig. 3. Average task loads of the servers (a) GFog servers, (b) CFog servers, (c) Cloud servers.

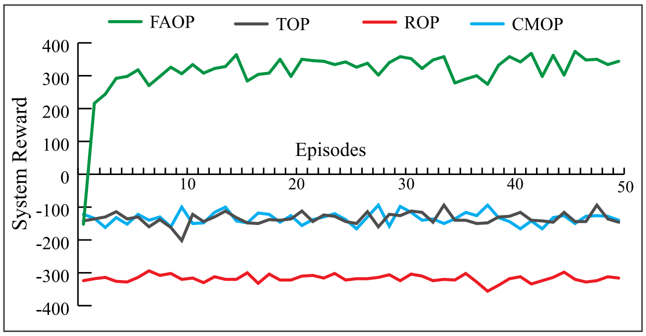

Fig. 4. Average total system reward.

Fig. 3 depicts the average tasks loads offloaded to the servers. Here, it is seen that FAOP and TOP deliver the tasks to the servers more suitable and balanced according to task types. However, ROP and CMOP offload the tasks to servers highly unbalanced. Task offloading results in Fig. 3 indicate that FAOP distributes tasks to suitable servers with an approximately 95% success rate.

We also investigate task processing delay, task transmission delay, and total system delay of all policies. Here, an average task completion time is calculated as the sum of the task processing time and the task transmission time. In order to provide more comprehensive results about delays, we set the transmission and protocol processing delays in study [67] as task offloading and task processing delays, respectively.

Based on Tables VII, VIII, and IX in [67], we set the transmission delay to fog and cloud servers to 15 ms and 120 ms, respectively. Besides, we set that DST, DSCIT, and CIT tasks are processed in CFog/GFog/Cloud Server at 4/8/16 ms, 20/40/80 ms, and 40/80/160 ms, respectively.

Fig. 5 shows the average task offloading and task processing delays separately. When the average processing delays are examined, it is seen that the proposed FOAP policy has the lowest task processing delay. This is because the FOAP policy makes server allocation for each task type more accurate than other policies. More precisely, FOAP policy produced approximately 34%, 35%, and 36% lower task processing delay than TOP, ROP, and CMOP policies, respectively. Other policies are listed as TOP, ROP and CMOP, from least to most. In terms of task transmission delays, we see that FOAP and TOP policies have lower delays than ROP and CMOP. Here, TOP has an advantage over other policies since it distributes the tasks more appropriately to the server type.

On the other hand, we calculate the average total system delays created by all policies and give them in Fig. 6. Considering the average total system delay, the FOAP policy revealed approximately 11%, 14%, and 15% lower latency than TOP, CMOP, and ROP policies, respectively. Overall results show that the proposed FAOP generates a lower delay time for all delay types. In this way, it ensures a positive effect on the overall system delays.

Based on the numerical results presented above, we can say that the FogAI supported DQL agent provides significant improvements in vital performance metrics of the IoT network such as workload distribution, task processing latency, data transfer latency and end-to-end latency. The results have shown that the proposed solutions have the potential to meet the needs of IoT architecture, especially in task-oriented communication. On the other hand, in the future cyber–physical world, we introduce a conceptual architecture that makes the challenges of the fog-based IoT network smarter, autonomous and scalable with AI-powered mechanisms. We believe that the proposed conceptual architecture and AI-powered fog components will provide a wide range of potential solutions to the covered challenges in this paper. Moreover, we hope that this study will motivate researchers to apply a new approach to solving NGIoT network problems.

8. Conclusion

In this paper, we comprehensively examine the challenges in FC and introduce AI-supported solutions that can overcome the problems of fog-based NGIoT systems. From the architectural point of view, we propose a new AI-supported fog controller concept

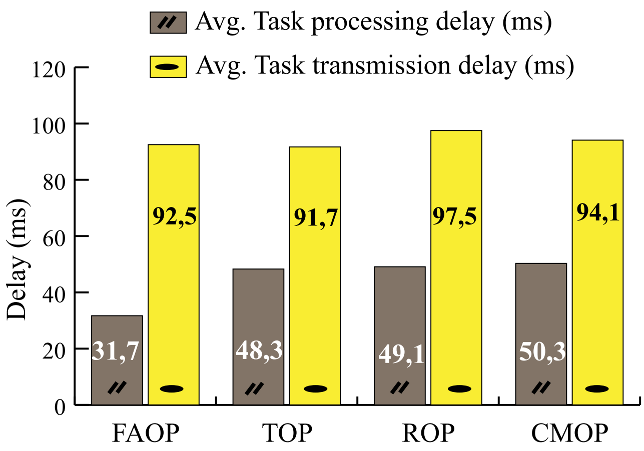

Fig. 5. Average task processing and task transmission delay.

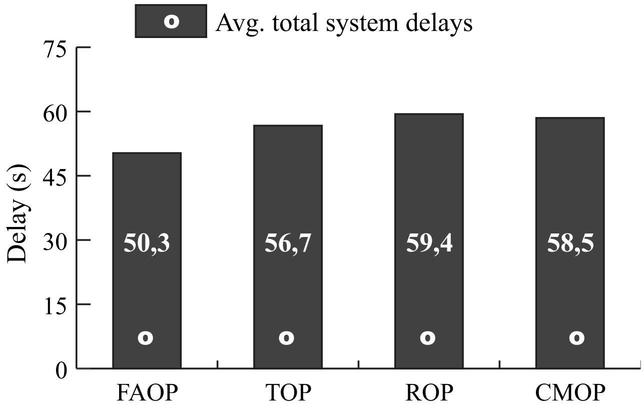

Fig. 6. Average total system delay.

called FogAI that makes the network management and control dynamic by abstracting the fog layer from the physical and cloud layer. From the algorithmic point of view, we present deep learning agents, models, and algorithms that can work in FogAI. Moreover, to fully understand the proposed FogAI concept, we present a case study on DQL-based task offloading to minimize task processing and response time, and balance the task workload of the system. In conclusion, we expect that, in addition to solving FC layer problems, FogAI concept can make the whole IoT network smarter and more dynamic by providing coordination between the physical and cloud layers.

This study identifies the FC-related issues and emphasizes the importance of the FogAI and its potential solutions; however, how to overcome each challenge precisely is yet unclear and incomplete. We sought to provide potential FogAI components and FogAI-based solutions as suggestions for each challenge; however, each solution can be enhanced using different methods. In this paper, we present an example of how the proposed concept is applied to the task offloading problem. As part of future research, we want to show how the FogAI concept may be applied to other issues including mobility, security, and heterogeneity.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under the grant number 118E212.

References

[1] P. Bellavista, J. Berrocal, A. Corradi, S.K. Das, L. Foschini, A. Zanni, A survey on fog computing for the internet of things, Pervasive Mob. Comput. 52 (2019) 71–99.
[2] Cisco, Cisco Annual Internet Report (2018–2023) White Paper, Tech. Rep., 2020.
[3] F. Saeik, M. Avgeris, D. Spatharakis, N. Santi, D. Dechouniotis, J. Violos, A. Leivadeas, N. Athanasopoulos, N. Mitton, S. Papavassiliou, Task offloading in edge and cloud computing: A survey on mathematical, artificial intelligence and control theory solutions, Comput. Netw. 195 (2021) 108177.
[4] H.F. Atlam, R.J. Walters, G.B. Wills, Fog computing and the internet of things: a review, Big Data Cognit. Comput. 2 (2) (2018) 10.
[5] O.-K. Shahryari, H. Pedram, V. Khajehvand, M.D. TakhtFooladi, Energy-efficient and delay-guaranteed computation offloading for fog-based IoT networks, Comput. Netw. 182 (2020) 107511.
[6] M. Mukherjee, L. Shu, D. Wang, Survey of fog computing: Fundamental, network applications, and research challenges, IEEE Commun. Surv. Tutor. 20 (3) (2018) 1826–1857.
[7] K.S. Desikan, V.J. Kotagi, C.S.R. Murthy, Topology control in fog computing enabled IoT networks for smart cities, Comput. Netw. 176 (2020) 107270.
[8] Y. Zhan, Y. Xia, A.V. Vasilakos, Future directions of networked control systems: A combination of cloud control and fog control approach, Comput. Netw. 161 (2019) 235–248.
[9] O. Salman, I. Elhajj, A. Chehab, A. Kayssi, IoT survey: An SDN and fog computing perspective, Comput. Netw. 143 (2018) 221–246.
[10] O. Vermesan, M. EisenHauer, M. Serrano, P. Guillemin, H. Sundmaeker, E.Z. Tragos, J. Valino, B. Copigneaux, M. Presser, A. Aagaard, et al., The next generation internet of things–hyperconnectivity and embedded intelligence at the edge, in: Next Generation Internet of Things. Distributed Intelligence at the Edge and Human Machine-to-Machine Cooperation, 2018.
[11] A.E. Abdelaal, T. Hegazy, M. Hefeeda, Event-based control as a cloud service, in: 2017 American Control Conference (ACC), IEEE, 2017, pp. 1017–1023.
[12] S. Guan, S. Niu, Stability-based controller design of cloud control system with uncertainties, IEEE Access 9 (2021) 29056–29070.
[13] H. Inaltekin, M. Gorlatova, M. Chiang, Virtualized control over fog: Interplay between reliability and latency, IEEE Internet Things J. 5 (6) (2018) 5030–5045.
[14] M. Yannuzzi, F. van Lingen, A. Jain, O.L. Parellada, M.M. Flores, D. Carrera, J.L. Pérez, D. Montero, P. Chacin, A. Corsaro, et al., A new era for cities with fog computing, IEEE Internet Comput. 21 (2) (2017) 54–67.
[15] M.A. Al Faruque, K. Vatanparvar, Energy management-as-a-service over fog computing platform, IEEE Internet Things J. 3 (2) (2015) 161–169.
[16] A. Alomari, S.K. Subramaniam, N. Samian, R. Latip, Z. Zukarnain, Resource management in SDN-based cloud and SDN-based fog computing: taxonomy study, Symmetry 13 (5) (2021) 734.
[17] J. Ren, J. Li, H. Liu, T. Qin, Task offloading strategy with emergency handling and blockchain security in SDN-empowered and fog-assisted healthcare IoT, Tsinghua Sci. Technol. 27 (4) (2021) 760–776.
[18] A.J. Kadhim, J.I. Naser, Proactive load balancing mechanism for fog computing supported by parked vehicles in IoV-SDN, China Commun. 18 (2) (2021) 271–289.
[19] A. Alamer, Security and privacy-awareness in a software-defined fog computing network for the internet of things, Opt. Switch. Netw. 41 (2021) 100616.
[20] K. Velasquez, D.P. Abreu, M.R. Assis, C. Senna, D.F. Aranha, L.F. Bittencourt, N. Laranjeiro, M. Curado, M. Vieira, E. Monteiro, et al., Fog orchestration for the internet of everything: state-of-the-art and research challenges, J. Internet Serv. Appl. 9 (1) (2018) 1–23.
[21] J. Liu, T. Zhao, S. Zhou, Y. Cheng, Z. Niu, CONCERT: a cloud-based architecture for next-generation cellular systems, IEEE Wirel. Commun. 21 (6) (2014) 14–22, http://dx.doi.org/10.1109/MWC.2014.7000967.
[22] Z. Wen, R. Yang, P. Garraghan, T. Lin, J. Xu, M. Rovatsos, Fog orchestration for internet of things services, IEEE Internet Comput. 21 (2) (2017) 16–24, http://dx.doi.org/10.1109/MIC.2017.36.
[23] D. Santoro, D. Zozin, D. Pizzolli, F. De Pellegrini, S. Cretti, Foggy: A platform for workload orchestration in a fog computing environment, in: 2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), 2017, pp. 231–234, http://dx.doi.org/10.1109/CloudCom.2017.62.
[24] R. Yang, Z. Wen, D. McKee, T. Lin, J. Xu, P. Garraghan, Software-defined fog orchestration for IoT services, in: Fog and Fogonomics: Challenges and Practices of Fog Computing, Communication, Networking, Strategy, and Economics, Wiley Online Library, 2020, pp. 179–212.
[25] Y.E. Oktian, S. Lee, H. Lee, J. Lam, Distributed SDN controller system: A survey on design choice, Comput. Netw. 121 (2017) 100–111.
[26] L. Sidki, Y. Ben-Shimol, A. Sadovski, Fault tolerant mechanisms for SDN controllers, in: 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), IEEE, 2016, pp. 173–178.
[27] U. Ghosh, X. Dong, R. Tan, Z. Kalbarczyk, D.K. Yau, R.K. Iyer, A simulation study on smart grid resilience under software-defined networking controller failures, in: Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security, 2016, pp. 52–58.
[28] W. Chen, X. Qiu, T. Cai, H.-N. Dai, Z. Zheng, Y. Zhang, Deep reinforcement learning for internet of things: A comprehensive survey, IEEE Commun. Surv. Tutor. (2021).
[29] N.C. Luong, D.T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, D.I. Kim, Applications of deep reinforcement learning in communications and networking: A survey, IEEE Commun. Surv. Tutor. 21 (4) (2019) 3133–3174, http://dx.doi.org/10.1109/COMST.2019.2916583.
[30] C. Puliafito, E. Mingozzi, F. Longo, A. Puliafito, O. Rana, Fog computing for the internet of things: A survey, ACM Trans. Internet Technol. (TOIT) 19 (2) (2019) 1–41.
[31] T.A. Butt, Context-aware cognitive disaster management using fog-based internet of things, Trans. Emerg. Telecommun. Technol. (2019) e3646.
[32] Z. Ning, J. Huang, X. Wang, Vehicular fog computing: Enabling real-time traffic management for smart cities, IEEE Wirel. Commun. 26 (1) (2019) 87–93.
[33] Y. Zhou, L. Tian, L. Liu, Y. Qi, Fog computing enabled future mobile communication networks: A convergence of communication and computing, IEEE Commun. Mag. 57 (5) (2019) 20–27.
[34] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, J.P. Jue, All one needs to know about fog computing and related edge computing paradigms: A complete survey, J. Syst. Archit. 98 (2019) 289–330.
[35] J. Dizdarević, F. Carpio, A. Jukan, X. Masip-Bruin, A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration, ACM Comput. Surv. 51 (6) (2019) 1–29.
[36] C. Mouradian, D. Naboulsi, S. Yangui, R.H. Glitho, M.J. Morrow, P.A. Polakos, A comprehensive survey on fog computing: State-of-the-art and research challenges, IEEE Commun. Surv. Tutor. 20 (1) (2017) 416–464.
[37] L. Wan, L. Sun, X. Kong, Y. Yuan, K. Sun, F. Xia, Task-driven resource assignment in mobile edge computing exploiting evolutionary computation, IEEE Wirel. Commun. 26 (6) (2019) 94–101.
[38] F.Y. Okay, S. Ozdemir, Routing in fog-enabled IoT platforms: A survey and an SDN-based solution, IEEE Internet Things J. 5 (6) (2018) 4871–4889.
[39] X. He, H. Lu, H. Huang, Y. Mao, K. Wang, S. Guo, QoE-based cooperative task offloading with deep reinforcement learning in mobile edge networks, IEEE Wirel. Commun. 27 (3) (2020) 111–117.
[40] P. Maiti, J. Shukla, B. Sahoo, A.K. Turuk, Mathematical modeling of qos-aware fog computing architecture for iot services, in: Emerging Technologies in Data Mining and Information Security, Springer, 2019, pp. 13–21.
[41] J. Ni, K. Zhang, X. Lin, X.S. Shen, Securing fog computing for internet of things applications: Challenges and solutions, IEEE Commun. Surv. Tutor. 20 (1) (2017) 601–628.

[42] A. Alrawais, A. Alhothaily, C. Hu, X. Cheng, Fog computing for the internet of things: Security and privacy issues, IEEE Internet Comput. 21 (2) (2017) 34–42.
[43] Y. Hao, Y. Miao, L. Hu, M.S. Hossain, G. Muhammad, S.U. Amin, Smart-edge-cocaco: AI-enabled smart edge with joint computation, caching, and communication in heterogeneous IoT, IEEE Netw. 33 (2) (2019) 58–64.
[44] P. Zhang, M. Zhou, G. Fortino, Security and trust issues in fog computing: A survey, Future Gener. Comput. Syst. 88 (2018) 16–27.
[45] M. Mukherjee, R. Matam, L. Shu, L. Maglaras, M.A. Ferrag, N. Choudhury, V. Kumar, Security and privacy in fog computing: Challenges, IEEE Access 5 (2017) 19293–19304.
[46] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F.R. Yu, Z. Han, Computing resource allocation in three-tier IoT fog networks: A joint optimization approach combining stackelberg game and matching, IEEE Internet Things J. 4 (5) (2017) 1204–1215.
[47] H.-K. Lim, J.-B. Kim, J.-S. Heo, Y.-H. Han, Federated reinforcement learning for training control policies on multiple IoT devices, Sensors 20 (5) (2020) 1359.
[48] N. El Abid Amrani, M. Youssfi, O.E.K. Abra, Semantic interoperability between heterogeneous multi-agent systems based on deep learning, in: 2018 6th International Conference on Multimedia Computing and Systems (ICMCS), 2018, pp. 1–6.
[49] T.D. Nguyen, P. Rieger, M. Miettinen, A.-R. Sadeghi, Poisoning attacks on federated learning-based IoT intrusion detection system, 2020.
[50] Y. He, N. Zhao, H. Yin, Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach, IEEE Trans. Veh. Technol. 67 (1) (2017) 44–55.
[51] J. Dequaire, D. Rao, P. Ondruska, D. Wang, I. Posner, Deep tracking on the move: Learning to track the world from a moving vehicle using recurrent neural networks, 2016, arXiv preprint arXiv:1609.09365.
[52] H. Khelifi, S. Luo, B. Nour, A. Sellami, H. Moungla, F. Naït-Abdesselam, An optimized proactive caching scheme based on mobility prediction for vehicular networks, in: 2018 IEEE Global Communications Conference (GLOBECOM), 2018, pp. 1–6.
[53] C.C. Byers, Architectural imperatives for fog computing: Use cases, requirements, and architectural techniques for fog-enabled IoT networks, IEEE Commun. Mag. 55 (8) (2017) 14–20.
[54] A. Gupta, R. Christie, P. Manjula, Scalability in internet of things: features, techniques and research challenges, Int. J. Comput. Intell. Res. 13 (7) (2017) 1617–1627.
[55] X. Xu, S. Fu, Q. Cai, W. Tian, W. Liu, W. Dou, X. Sun, A.X. Liu, Dynamic resource allocation for load balancing in fog environment, Wirel. Commun. Mobile Comput. 2018 (2018).
[56] F.M. Talaat, M.S. Saraya, A.I. Saleh, H.A. Ali, S.H. Ali, A load balancing and optimization strategy (LBOS) using reinforcement learning in fog computing environment, J. Ambient Intell. Humaniz. Comput. (2020) 1–16.
[57] H.-Y. Kim, J.-M. Kim, A load balancing scheme based on deep-learning in IoT, Cluster Comput. 20 (1) (2017) 873–878.
[58] C. Guerrero, I. Lera, C. Juiz, Evaluation and efficiency comparison of evolutionary algorithms for service placement optimization in fog architectures, Future Gener. Comput. Syst. 97 (2019) 131–144.
[59] J. Pereira, L. Ricardo, M. Luís, C. Senna, S. Sargento, Assessing the reliability of fog computing for smart mobility applications in VANETs, Future Gener. Comput. Syst. 94 (2019) 317–332.
[60] I. Kök, B.H. Çorak, U. Yavanoğlu, S. Özdemir, Deep learning based delay and bandwidth efficient data transmission in IoT, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 2327–2333.
[61] D. Van Le, C. Tham, A deep reinforcement learning based offloading scheme in ad-hoc mobile clouds, in: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2018, pp. 760–765.
[62] T.X. Tran, D. Pompili, Joint task offloading and resource allocation for multi-server mobile-edge computing networks, IEEE Trans. Veh. Technol. 68 (1) (2019) 856–868, http://dx.doi.org/10.1109/TVT.2018.2881191.
[63] T. Zhang, Y.-H. Chiang, C. Borcea, Y. Ji, Learning-based offloading of tasks with diverse delay sensitivities for mobile edge computing, in: 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, 2019, pp. 1–6.
[64] M. Huang, W. Liu, T. Wang, A. Liu, S. Zhang, A cloud–MEC collaborative task offloading scheme with service orchestration, IEEE Internet Things J. 7 (7) (2019) 5792–5805.
[65] D. Wang, X. Tian, H. Cui, Z. Liu, Reinforcement learning-based joint task offloading and migration schemes optimization in mobility-aware MEC network, China Commun. 17 (8) (2020) 31–44.
[66] S. Yang, A joint optimization scheme for task offloading and resource allocation based on edge computing in 5G communication networks, Comput. Commun. 160 (2020) 759–768.
[67] I. Kök, S. Özdemir, DeepMDP: A novel deep learning based missing data prediction protocol for IoT, IEEE Internet Things J. (2020) 1.

Ibrahim Kök received his M.Sc. and Ph.D. degrees in Computer Science from Gazi University in 2015 and 2020, respectively. He is currently an Assistant Professor with the Department of Computer Engineering, Pamukkale University, Denizli. His current research interests include Internet of Things (IoT), Deep Learning, AI-enabled IoT, and IoT Data Analytics.

Feyza Yildirim Okay received the M.Sc. and Ph.D. degrees in Computer Engineering from Gazi University Graduate School of Natural and Applied Sciences in 2013 and 2019, respectively. She is currently working as a research assistant at the same institute. Her research interests include Internet of Things, Fog Computing, Software Defined Networking, and Network Security.

Suat Ozdemir is with the Department of Computer Engineering at Hacettepe University, Ankara, Turkey. Before joining Hacettepe University, he worked at Gazi University between 2007 and 2020. He received his M.Sc. degree in Computer Science from Syracuse University (August 2001) and Ph.D. degree in Computer Science from Arizona State University (December 2006). His current research interests include Internet of Things, Data Analytics, Artificial Intelligence and Network Security