Evaluating the performance of a multimodal speaker tracking system at the edge-to-cloud continuum

Alessio Orsino, Riccardo Cantini, Fabrizio Marozzo

Abstract The edge-to-cloud compute continuum has become increasingly popular in recent years for effectively collecting and analyzing data generated by Internet of Things (IoT) devices at the network edge, ensuring low latency, high scalability, and privacy preservation. This continuum of computing resources, features, and services, which spans from the edge to the cloud, can be effectively leveraged in various application domains like smart cities, industrial IoT, and smart healthcare. However, many unexplored scenarios still exist where this technology can be successfully applied. This chapter investigates how the compute continuum can support speaker tracking in smart spaces, such as smart homes, offices, and public venues, especially focusing on multimodal systems that leverage both audio and visual data. The effectiveness of the edge-to-cloud continuum in supporting such systems was assessed through a simulation-based experimental evaluation performed with the iFogSim toolkit. Our findings reveal that edge-cloud integration improves application performance in terms of network usage and latency, compared to a centralized solution that solely relies on cloud computing.

Alessio Orsino
Department of Informatics, Modeling, Electronics and Systems (DIMES), University of Calabria, Italy e-mail: aorsino@dimes.unical.it

Riccardo Cantini
Department of Informatics, Modeling, Electronics and Systems (DIMES), University of Calabria, Italy e-mail: rcantini@dimes.unical.it

Fabrizio Marozzo
Department of Informatics, Modeling, Electronics and Systems (DIMES), University of Calabria, Italy e-mail: fmarozzo@dimes.unical.it

1.1 Introduction

In recent years, the rise of the Internet of Things (IoT) has led to the generation of massive amounts of high-velocity and heterogeneous data at the network edge [6, 2]. To effectively collect, process, and analyze these data, edge-to-cloud continuum solutions have emerged, which integrate the features and services provided by both edge and cloud computing, enabling real-time and data-driven decision-making in various domains [14]. In fact, current applications for processing IoT data primarily rely on cloud computing, posing challenges related to network traffic management and response time. To address these challenges, the edge computing paradigm has been introduced, allowing for data processing closer to the data source, thus offering benefits such as low latency, privacy preservation, and scalability. Nevertheless, due to the limited resources of edge devices, there is a need to combine their capabilities with cloud computing, which allows for the persistent aggregation and resource-intensive analysis of big data.

While the edge-to-cloud compute continuum has garnered considerable attention in cutting-edge application domains such as smart cities, industrial IoT, and smart healthcare, there are still many unexplored scenarios that can benefit from it. In particular, this study focuses on speaker tracking in smart spaces, such as smart homes, offices, and public venues. The localization and tracking of speakers in these environments have become increasingly important due to the widespread use of voice assistants, security systems, and smart meeting rooms [8]. Specifically, multimodal speaker tracking, which combines audio and visual data, has been proposed as a technique to ensure accurate and robust tracking, overcoming limitations of video-only or audio-only methods. In fact, video-only monitoring is limited by the camera's coverage area and faces challenges like occlusion and lighting variations, while audio-only solutions can be affected by noise and reverberations.

In this chapter, we specifically investigate the effectiveness of the edge-to-cloud continuum for multimodal speaker tracking in smart spaces, which leverages edge computing for real-time sensor data processing and cloud computing for higher-level processing and analysis. To evaluate the effectiveness of the edge-cloud integration, we followed a simulation-based approach, due to the large scale, heterogeneity, and complexity of such an IoT system, which poses significant challenges in system performance, scalability, and resource utilization. Indeed, modeling and simulation are essential to support the design and development of IoT applications, allowing a detailed evaluation before the real deployment. In particular, different design choices were investigated to understand their impact on the application performance in terms of bandwidth consumption and network latency, also comparing the performance of the edge-to-cloud continuum approach with a centralized cloud-based solution. The findings of our evaluation demonstrate the benefits of edge-cloud integration in improving application performance.

The remainder of this chapter is as follows. Section 1.2 provides the main concepts around multimodal speaker tracking in smart spaces and the compute continuum. Section 1.3 discusses related work. Section 1.4 describes how the multimodal speaker tracking system was modeled for simulation purposes. Section 1.5 presents the performance evaluation of deploying such an application at the edge-to-cloud continuum, by following a simulation-based approach. Finally, Section 1.6 concludes the chapter.

1.2 Background

This section provides the preliminaries to the rest of this chapter. In particular, we first discuss the task of multimodal speaker tracking in smart spaces and state-of-the-art techniques. Later, we introduce the edge-to-cloud compute continuum and its multi-tier structure.

Multimodal speaker tracking in smart spaces. Multimodal speaker tracking is the process of detecting and locating speakers in audio-visual contexts such as video conferences and public venues. This involves analyzing both audio and video streams to identify speakers' location and track their movements over time, which makes the task challenging due to issues like video occlusion, background noise, and changes in lighting conditions.

Generally, the audio information is used to determine the direction of arrival (DOA) of the voice, by estimating the time delay of arrival (TDOA) of the signal to different microphones. One common algorithm is the Steered Response Power (SRP) algorithm, which analyzes the signals received by an array of microphones or sensors to determine the direction from which the sound is coming. The SRP algorithm uses a steered beamformer approach to enhance the desired signal coming from a particular direction while suppressing interference from other directions. This is done by computing the power spectral density of the received signals at different spatial locations and then steering the beamformer toward the direction with the maximum power, which corresponds to the estimated DOA of the sound source. However, a challenge with these systems is that their precision depends on the density of points to be evaluated, making them expensive for real-time applications. To solve this issue, the space to be scanned can be divided into sectors to identify possible sound sources, which can be further reduced by overlapping sectors in systems with multiple microphone arrays.

In addition to audio processing, video processing is also employed to identify potential speakers based on visual cues, such as lip movement and head orientation, and track them. These visual cues can be combined with audio-based DOA information to improve the robustness and accuracy of speaker tracking in complex environments with multiple speakers and background noise. This can be done by using the particle filter (PF) algorithm, adding the DOA information obtained from audio processing to the particle propagation stage of the PF. In particular, the PF is a type of sequential Monte Carlo (SMC) method used for estimating the state of a dynamic system based on noisy measurements. The basic idea is to represent the probability distribution of the system state using a set of discrete particles, where each particle represents a hypothesis or a guess about the true state of the system at a given time. These particles are propagated through time using a state transition model that describes how the system evolves over time. At each time step, measurements of the system are used to update the particle weights, which reflect the likelihood of each particle being the true state of the system. Therefore, particles with higher weights are considered more likely to represent the true state of the system.

Based on these concepts, different techniques have been proposed in the literature to address the problem of speaker localization and tracking in smart spaces. Qian et al. [15] proposed an algorithm that integrates audio and visual cues from a localized multi-modal sensor platform, using a PF framework to dynamically combine the cues while considering audio signals measured by the maximum Global Coherence Field (GCF). Liu et al. [10] proposed a two-layer PF algorithm for multimodal speaker tracking, generating two sets of particles from the audio and video streams independently and propagating them in separate audio and visual layers. The authors combined the audio and visual likelihoods using an adaptive sigmoid function that adjusts particle weights based on the confidence of the two modalities. In a subsequent work, the same authors [11] modified the prediction and update stage of the PF algorithm, refining the direction of the particles with multimodal information in the prediction stage and calculating the particle likelihood by combining visual distance and audio-visual direction information in the update stage. The distance likelihood was obtained using the camera projection model and the estimated size of the speaker's face, while the direction likelihood was determined by audio-visual particle fitness.

Compute continuum. The edge-to-cloud compute continuum refers to the continuum of computing resources that span from the edge of a network to the cloud. It typically consists of the following layers, categorized based on the proximity of computing resources to the data source and the level of processing that occurs at each layer:

- The *edge layer* is the closest to the data source and generally includes edge devices, such as IoT devices, which perform local processing of the data according to the edge computing paradigm. This allows for reducing latency and bandwidth usage by processing data locally, near the data source, especially in real-time or near-real-time scenarios, such as autonomous vehicles, smart industries, smart cities, and smart spaces.
- The *fog layer* is an intermediate layer between the edge and the cloud. It includes local data centers or computing resources that are closer to the edge but are more powerful than edge devices, such as gateways. Fog computing provides edge devices with additional processing capabilities, storage, and networking resources, allowing them to offload some of the processing tasks that they are unable to complete, while still maintaining lower latency than the cloud.
- The *cloud layer* refers to the centralized computing infrastructure that is typically located in remote data centers and provides on-demand computing resources, such as virtual machines, storage, and services, over the internet. The cloud layer offers scalability, elasticity, and cost-efficiency, and is used for compute-intensive data processing and storage.

This distributed architecture provides several advantages and benefits that can greatly enhance the capabilities of modern computing systems. In fact, by leveraging the strengths of both edge and cloud computing, edge-cloud integration allows for real-time data processing and analysis, greater scalability, improved security and privacy, and more efficient use of resources. This can be especially useful in IoT applications such as industrial automation, autonomous vehicles, and intelligent environments like smart cities and smart spaces, where speed, reliability, and responsiveness are critical.

1.3 Related work

Deploying and testing an IoT system for multimodal speaker tracking in a real-world smart space can be costly and logistically challenging. To address these issues, simulation approaches can be key to guiding design choices for IoT system modeling and validation in edge-cloud architectures. Indeed, simulation allows the exploration of different deploying strategies in the compute continuum, resource allocation policies, and system configurations in a controlled and reproducible environment. Also, it enables the evaluation of system performance under varying conditions, such as changing network conditions, workloads, and dynamic resource availability across the various layers of the edge-cloud architecture. Furthermore, it provides insights into the impact of different parameters on system behavior, helping identify performance bottlenecks and optimization opportunities. Finally, simulation is cost-effective compared to real-world implementations, as it requires no physical infrastructure and allows for the evaluation of system performance in extreme or rare scenarios that may be difficult to replicate in real-world settings [4, 7, 9].

Various open-source simulators, such as iFogSim [5], IoTSim [20], and Edge-CloudSim [19], have been proposed in the literature to simulate IoT environments, and several research works have used simulation-based approaches to test specific IoT applications on edge-cloud architectures [18, 3, 12].

1.4 Modeling and simulation of a speaker tracking system

In this section, we describe how a speaker tracking system can be modeled and simulated using iFogSim, an open-source toolkit that provides a comprehensive framework for the modeling, simulation, and evaluation of fog computing environments. It enables the modeling of diverse elements in a fog computing environment, including fog devices, IoT devices, and cloud servers. Moreover, it facilitates the simulation of their interactions, along with the evaluation of fog computing systems' efficiency, scalability, and overall performance.

In the following of this section, we delve into the details of our study, beginning with the application modeling and simulation parameters, followed by a comprehensive overview of the edge-to-cloud architecture utilized in our simulations.

1.4.1 Application modeling

The speaker tracking system comprises different modules that are connected to each other, as shown in the Directed Data Flow (DDF) model in Figure 1.1. In this model, the application is represented as a directed graph where the vertices are application modules and the directed edges convey the flow of data between modules.

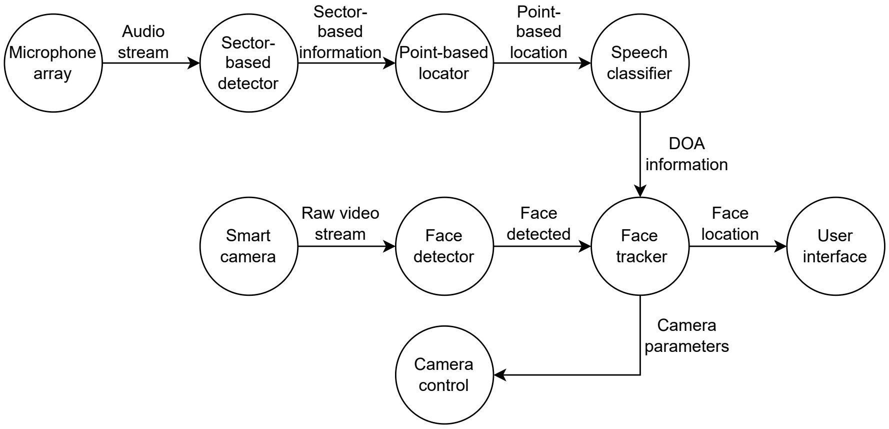


Fig. 1.1 DDF application model of a multimodal speaker tracking system.

In particular, the system is divided into nine modules, which are described below.

- The *sector-based detector*, which identifies when a specific sector is active, based on audio input signals. It utilizes signal processing techniques to extract relevant features and classify the sector's activity level. The information is then transmitted to the point-based locator for further processing.
- The *point-based locator*, which operates on a grid of points within the identified sector and calculates the SRP for each point. The information about the top two points with the highest SRP is then passed to the speech classifier, indicating the most probable location of the audio source.
- The *speech classifier*, which determines if the audio response detected by the microphones originates from a human speaker. Additionally, it processes the DOA information received from the point-based locator to estimate the speaker's location. The DOA information is also forwarded to the face tracker in order to be combined with the visual information detected by the smart cameras.

- The *face detector*, which selects and transmits the frames captured by a smart camera to the face tracker.
- The *face tracker*, which employs an Audio-Visual Particle Filter (AV-PF) to accurately track the detected face's movement over time. The module continuously updates the estimated face position based on the audio and visual signals received and transmits the final estimated face position to the *user interface* for user interaction and to the *camera control* for tracking the speaker through the actuator.
- The *user interface*, which displays the frames captured by the camera and the detected face positions in real-time, allows users to track the speaker's movement. It may also provide additional functionalities, such as displaying the sector information, SRP values, and DOA information for further user interaction and analysis.
- The *camera control*, which modifies the position of the smart camera to track the speaker, based on the location detected by the face tracker. This module acts as the actuator of the system.

The properties of tuples carried by edges between the modules in the application are described in Table 1.1, in terms of CPU length, expressed in MIPS, and network length, which refers to the communication cost between fog devices or fog nodes in the fog layer. The values were chosen according to different works of the literature [13, 5, 16, 17, 1].

Tuple type	**CPU length (MIPS)**	**N/W length**
Audio stream	1,000	150
Sector-based information	1,500	150
Point-based location	1,000	150
DOA estimation	2,000	100
Raw video stream	1,000	20,000
Face detected	2,000	2,000
Face location	500	2,000
Camera parameters	100	100

Table 1.1 Tuple type parameters.

1.4.2 Edge-to-cloud architecture

To investigate the behavior of the system and evaluate its performance, we used an architecture composed of the following physical devices: *i*) smart cameras and microphone arrays at the edge layer, which capture real-time audio and video data from the smart space; *ii*) gateways at the fog layer, which support more resource-intensive tasks like speech classification or face tracking; and *iii*) the cloud, which receives the processed data from the fog layer for further analysis and storage.

In particular, the fog layer also handles communication with the Internet Service Provider (ISP) gateway, which provides access to the Internet for the cameras and microphone arrays. Moreover, the cloud layer also provides services and APIs for applications, including a user interface to interact with processed data. The system architecture is shown in Figure 1.2.

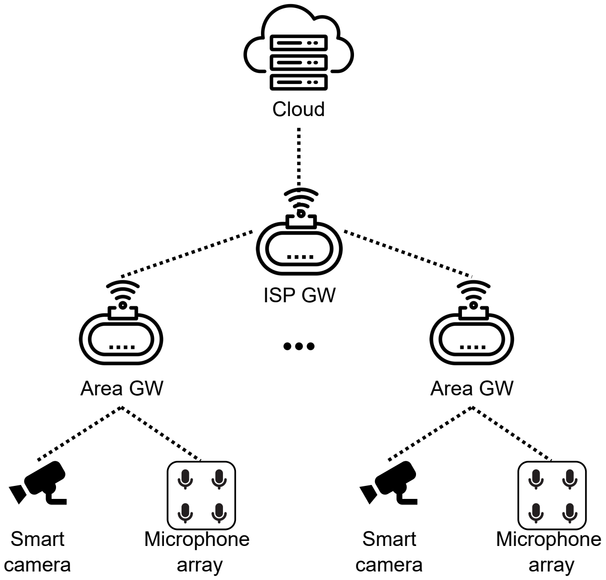


Fig. 1.2 Edge-to-cloud architecture to support a multimodal speaker tracking system.

Based on this architecture, we have explored two distinct strategies for the deployment and placement of application modules, namely the *cloud-only* and the *edge-ward* approaches. In the former, all the modules that make up the application are deployed in a remote data center, following the traditional cloud-based deployment model. In the latter, the deployment of application modules occurs closer to the network edge. However, devices located at the network edge, such as cameras and microphone arrays, may have limited computing power, which may not be sufficient to meet the application requirements due to their resource-constrained nature. Therefore, in a such case, the fog resources are iteratively exploited up to the cloud. Table 1.2 describes the simulation parameters used to configure the physical topology. Specifically, the physical devices in the architecture, i.e., smart cameras, microphone arrays, area gateways, ISP gateways, and cloud — ordered by their location from edge to cloud — are described in terms of MIPS of the CPU, RAM, and the upload latency to the destination device in the architecture.

In order to investigate how the edge-to-cloud compute continuum can support the described system, we analyzed different simulation configurations with varying numbers of devices. In particular, the system was tested across various physical

Device	CPU (MIPS)	RAM (GB)	Latency (ms)	Destination
Microphone array	500	1	1	Area gateway
Smart camera	500	1	1	Area gateway
Area gateway	11,200	16	2	ISP gateway
ISP gateway	22,400	64	100	Cloud
Cloud	44,800	128	-	-

Table 1.2 Configuration of the different physical devices of the architecture.

topology configurations with different numbers of cameras and monitored areas. We tested the use of 2 and 4 cameras and microphone arrays, combined with different numbers of covered areas, i.e., 2, 4, 8, and 16. Therefore, in each area, 2 or 4 cameras and microphone arrays can be strategically placed to monitor that area. Instead, the number of microphones was set constant to 4 throughout all simulations. In summary, a total of eight configurations, referred to as Config 1, Config 2, Config 3, Config 4, Config 5, Config 6, Config 7, and Config 8 in the experimental evaluation, have been tested as shown in Table 1.3. These configurations have been strategically chosen to simulate both small smart spaces that can be covered by 2 cameras and microphones, and larger spaces that need more cameras and microphones. Each simulation configuration is run according to the two deployments described above, i.e. *cloud-only* and *edge-ward*.

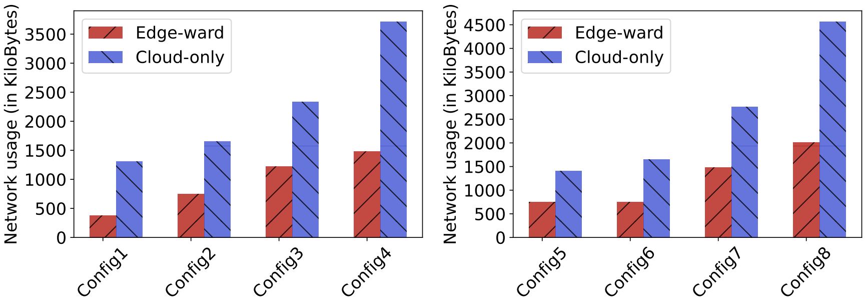
	Configuration							
	Config 1	**Config 2**	**Config 3**	**Config 4**	**Config 5**	**Config 6**	**Config 7**	**Config 8**
No. of areas	2	4	8	16	2	4	8	16
No. of cameras per area	2	2	2	2	4	4	4	4
No. of mic. arrays per area	2	2	2	2	4	4	4	4

Table 1.3 Description of the different simulation configurations with a varying number of components.

1.5 Performance evaluation

This section presents the results of simulations carried out on an edge-to-cloud compute continuum architecture, aimed at evaluating the efficiency of two deployment strategies for the described speaker tracking system, namely cloud-only and edge-ward. In the cloud-only strategy, all processing is done in the cloud, while in the edge-ward strategy, the processing is distributed along the edge-to-cloud continuum.

We compared the performance achieved by using the two deployment strategies in all configurations described in Table 1.3. The objective of these simulations was to understand to what extent an edge-cloud environment can reduce network usage and latency for a multimodal speaker tracking system.



(a) Small areas.

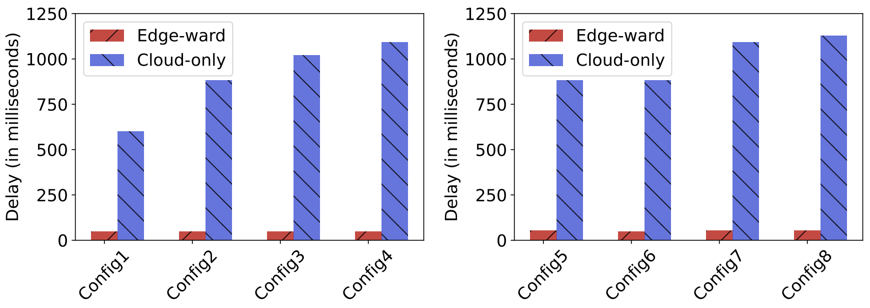
(b) Large areas.

Fig. 1.3 Comparison of the network usage achieved by the cloud-based and edge-ward policies with eight configurations.

Figure 1.3 shows the total network usage for each of the eight configurations. The results are similar both when considering small areas that can be monitored by 2 cameras and 2 microphones (Config 1 to 4), and large areas that require a higher number of cameras and microphones (Config 5 to 8). In particular, as the number of configurations becomes more complex in terms of the number of areas to be covered (from 2 to 16), the network load increases and becomes a significant challenge, especially when relying solely on cloud resources. Instead, as the results in Figure 1.3 suggest, the use of fog devices can significantly reduce network usage compared to cloud-only execution. In fact, using a cloud-based approach can result in uncontrolled growth of network usage, leading to network congestion and performance degradation of the application. On the contrary, the seamless integration of computing resources along the edge-to-cloud compute continuum allows for effectively reducing the network load, mitigating the risk of network congestion, and maintaining optimal performance of the application. In fact, instead of deploying modules on the cloud, the edge-ward policy distributes modules to different locations:

- the camera control runs by default on each smart camera;
- the face detector is deployed in the cameras at the edge layer;
- the sector-based detector is deployed in the microphones at the edge layer;
- the point-based locator and the speech classifier are deployed in the area gateways;
- the face tracker is deployed in the ISP gateway;
- the user interface runs by default in the cloud.

The findings from Figure 1.4 show the impact of the compute continuum on the latency of response of the system. In particular, the delays are considerably reduced when the modules are distributed over the edge-to-cloud continuum for each of the eight configurations, compared to the cloud-only deployment. This reduced delay results from the use of edge computing, where tasks are executed closer to the data source, in fog nodes or edge devices. This allows for faster processing and dramatically decrease in data transfer overheads, resulting in reduced delays.



(a) Small areas. (b) Large areas.

Fig. 1.4 Comparison of the delay achieved by the cloud-based and edge-ward policies with eight configurations.

1.6 Conclusions

The integration of computing resources across the edge-to-cloud compute continuum has garnered attention as a feasible solution for enabling efficient and effective collection, processing, and analysis of massive amounts of IoT data. In this chapter, we evaluated the use of an edge-to-cloud continuum architecture for a multimodal speaker tracking system, comparing two deployment strategies, namely cloud-only and edge-ward. The simulation results reveal that leveraging computing resources along the edge-to-cloud continuum, using the edge-ward approach, consistently outperforms the cloud-only approach in all simulated configurations, especially as the number of areas to be covered by smart cameras and microphone arrays increases. Therefore, this approach enables efficient resource utilization and reduces data transfer overheads, demonstrating better overall performance in terms of latency and network usage, while also ensuring robustness and scalability.

Acknowledgements This work has been supported by the ”RESTART - PNRR MUR – M4C2 - I 1.3” project - CUP C37G22000480001.

References

1. Alam, M.S., Jabin, S.J., Alam, A., Hossain, M.I.: Comparative analysis of cloud and fog environment based on network usage and cost of execution using ifogsim. In: 2021 International Conference on Decision Aid Sciences and Application (DASA), pp. 132–137. IEEE (2021)
2. Belcastro, L., Cantini, R., Marozzo, F., Orsino, A., Talia, D., Trunfio, P.: Programming big data analysis: Principles and solutions. Journal of Big Data **9**(4) (2022)
3. Belcastro, L., Marozzo, F., Orsino, A., Talia, D., Trunfio, P.: Edge-cloud continuum solutions for urban mobility prediction and planning. IEEE Access (2023). DOI 10.1109/ACCESS.2023.3267471
4. D'Angelo, G., Ferretti, S., Ghini, V.: Simulation of the internet of things. In: 2016 International Conference on High Performance Computing & Simulation (HPCS), pp. 1–8. IEEE (2016)
5. Gupta, H., Vahid Dastjerdi, A., Ghosh, S.K., Buyya, R.: ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments. Software: Practice and Experience **47**(9), 1275–1296 (2017)
6. Hassan, N., Gillani, S., Ahmed, E., Yaqoob, I., Imran, M.: The role of edge computing in internet of things. IEEE communications magazine **56**(11), 110–115 (2018)
7. Kecskemeti, G., Casale, G., Jha, D.N., Lyon, J., Ranjan, R.: Modelling and simulation challenges in internet of things. IEEE Cloud Computing **4**(1), 62–69 (2017). DOI 10.1109/MCC.2017.18
8. Kılıç, V., Barnard, M., Wang, W., Hilton, A., Kittler, J.: Mean-shift and sparse sampling-based smc-phd filtering for audio informed visual speaker tracking. IEEE Transactions on Multimedia **18**(12), 2417–2431 (2016)
9. Lima, L.E., Kimura, B.Y.L., Rosset, V.: Experimental environments for the internet of things: A review. IEEE Sensors Journal **19**(9), 3203–3211 (2019)
10. Liu, H., Li, Y., Yang, B.: 3d audio-visual speaker tracking with a two-layer particle filter. In: 2019 IEEE International Conference on Image Processing (ICIP), pp. 1955–1959. IEEE (2019)
11. Liu, H., Sun, Y., Li, Y., Yang, B.: 3d audio-visual speaker tracking with a novel particle filter. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 7343–7348. IEEE (2021)
12. Maheshwari, S., Raychaudhuri, D., Seskar, I., Bronzino, F.: Scalability and performance evaluation of edge cloud systems for latency constrained applications. In: 2018 IEEE/ACM Symposium on Edge Computing (SEC), pp. 286–299. IEEE (2018)
13. Mahmud, R., Pallewatta, S., Goudarzi, M., Buyya, R.: ifogsim2: An extended ifogsim simulator for mobility, clustering, and microservice management in edge and fog computing environments. Journal of Systems and Software **190**, 111351 (2022)
14. Marozzo, F., Orsino, A., Talia, D., Trunfio, P.: Edge computing solutions for distributed machine learning. In: 2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), pp. 1–8 (2022)
15. Qian, X., Brutti, A., Omologo, M., Cavallaro, A.: 3d audio-visual speaker tracking with an adaptive particle filter. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2896–2900. IEEE (2017)
16. Rahman, F.H., Au, T.W., Shah Newaz, S., Haji Suhaili, W.S.: A performance study of high-end fog and fog cluster in ifogsim. In: Computational Intelligence in Information Systems: Proceedings of the Computational Intelligence in Information Systems Conference (CIIS 2018) 3, pp. 87–96. Springer (2019)
17. Silva, D.M.A.d., Asaamoning, G., Orrillo, H., Sofia, R.C., Mendes, P.M.: An analysis of fog computing data placement algorithms. In: Proceedings of the 16th EAI international conference on mobile and ubiquitous systems: computing, networking and services, pp. 527–534 (2019)
18. Sinqadu, M., Shibeshi, Z.S.: Performance evaluation of a traffic surveillance application using ifogsim. In: International Conference on Wireless Intelligent and Distributed Environment for Communication, pp. 51–64. Springer (2020)

19. Sonmez, C., Ozgovde, A., Ersoy, C.: Edgecloudsim: An environment for performance evaluation of edge computing systems. Transactions on Emerging Telecommunications Technologies **29**(11), e3493 (2018)

20. Zeng, X., Garg, S.K., Strazdins, P., Jayaraman, P.P., Georgakopoulos, D., Ranjan, R.: Iotsim: A simulator for analysing iot applications. Journal of Systems Architecture **72**, 93–107 (2017)