1

Hybrid edge/cloud solutions for supporting autonomous vehicles

Loris Belcastro (lbelcastro@dimes.unical.it), Fabrizio Marozzo (fmarozzo@dimes.unical.it), Alessio Orsino (aorsino@dimes.unical.it)

Department of Informatics, Modeling, Electronics and Systems (DIMES), University of Calabria, Italy

Abstract

Researchers and IT companies have proposed in recent years the use of new hybrid edge/cloud solutions to efficiently process the huge amounts of data produced by IoT devices. In fact, edge combined with cloud computing is used in different application scenarios, such as that of autonomous vehicles, which often require low latency, energy saving, privacy protection and scalable services. Specifically, edge computing is useful in managing tasks that require real-time analysis and low response times, such as driving assistance, collision avoidance and road sign recognition. Instead, the use of the cloud is convenient for tasks that require a lot of resources and access to large data sets, such as diagnostic data collection and analysis, routing and targeted advertising. Designing and testing edge/cloud architectures to support autonomous vehicles are still open issues due to their large-scale, heterogeneity and complexity. In this chapter, we analyze how edge/cloud solutions can be exploited for efficiently managing tasks related to autonomous vehicle driving. In particular, through a simulation-based approach, we demonstrate that these solutions are capable of providing great performance benefits in support of autonomous vehicle systems, especially as the number of vehicles and computing nodes increase.

An Edited Volume, 1–13.

Keywords: Edge/cloud architecture, IoT infrastructure, Modeling and Simulation, Autonomous vehicles, Urban computing, Smart cities.

1.1 Introduction

In the last few years, huge volumes of data have been generated by and collected from several sources, such as sensors, cameras, smart meters, mobile devices, and wearables, which are commonly referred as Internet of Things (IoT) devices [13]. Such huge volumes of data, coupled with the speed with which they are generated, poses new research challenges in collecting, storing and analyzing them. To efficiently extract useful information and produce helpful knowledge for science, industry and public services, novel technologies, architectures and algorithms have been developed to capture and analyze this data [3].

Existing applications used for processing data from IoT devices are usually highly centralized and rely on the cloud for all operations related to data management, such as data collection, integration and analysis. However, fully relying on the cloud may result ineffective in terms of network traffic management, response times and energy consumption, particularly for some critical applications (e.g., medical and security) where it is essential to implement low-latency services to avoid serious problems, including fatal accidents [2]. Because of this, researchers and IT companies have proposed in recent years the adoption of the edge computing paradigm and the use of novel hybrid edge/cloud solutions for processing data closer to where it is generated. In this way, edge computing and cloud computing complement each other so that tasks that require real-time analysis and low response times can be run at the edge, while big data applications that benefit from data aggregation and compute-intensive analytics will run in the cloud.

Even in the field of autonomous vehicles, the use of edge/cloud solutions can prove to be extremely effective in managing the different tasks that are generated. For example, tasks such as driver assistance, collision avoidance and traffic sign recognition, which require real-time analysis and low response times, can take advantage of edge computing. Differently, tasks such as diagnostic data collection and analysis, route calculations and targeted advertising, which require a lot of computing resources and access to large data sets, can benefit from the use of cloud computing. When working with advanced machine learning tasks that require real-time analytics with minimal latency, edge-to-cloud cooperation turns out to be even more essential [16].

Designing and testing large-scale and multi-layer edge/cloud architectures are still open issues. In particular, designing and testing a distributed and heterogeneous infrastructure, composed of several components using different technologies and software stacks, could be very expensive and hard to manage [19]. For these reasons, simulation-based approaches are a powerful and flexible tool for reproducing and testing edge/cloud architectures, by avoiding the risks, costs and failures associated with extensive field experimentation [4].

In this chapter, we analyze how edge/cloud architectures can be exploited as an effective solution for managing tasks related to autonomous vehicle driving. In particular, we defined a hybrid edge/cloud architecture that distributes the workload between the cloud and the edge nodes in order to optimize application performance and manage a large number of autonomous vehicles. Furthermore, through a simulation-based approach, we also evaluated different design choices for improving the performance of autonomous vehicles tasks in terms of processing time, network delay, task failure and computing resource utilization.

The structure of this chapter is as follows. Section 1.2 discusses related work. Section 1.3 describes the hybrid edge/cloud architecture. Section 1.4 presents a case study and a performance evaluation. Finally, Section 1.5 concludes the chapter.

1.2 Related Work

With the widespread availability of IoT devices, there was a growing demand for new solutions for extracting useful information from large amounts of data produced by such devices. These solutions are necessary for big data analytics in IoT environments in order to process and analyze huge volumes of structured, unstructured, or semi-structured data. Machine learning algorithms are used to identify patterns, trends, and correlations in data [14, 1], but it is important to consider that these algorithms are run by devices having limited resources such as memory, processing, bandwidth, and energy [23]. As a result, it is essential to strike the correct balance between performance (e.g., the accuracy of the machine learning model) and the amount of resources required for computing.

To analyze and validate these solutions, thorough testing on IoT environments with an ever-increasing number of devices is necessary. Modeling and simulation techniques can help in the design and validation of IoT environments. The primary issues in testing big IoT applications were described by

Bosmans et al. [6]. The authors, in particular, proposed a novel simulation-based testing technique capable of facilitating interactions between real-life and virtual local IoT entities. Instead, D'Angelo et al. [7] presented the main issues of modeling IoT systems and how to improve scalability using simulation approaches. Through a performance analysis related to the use of services in a smart city, the same authors in [8] analyzed the feasibility of using simulation to construct and compose heterogeneous IoT simulation scenarios.

In terms of tools and software solutions, different simulators, such as iFog-Sim [9], EdgeCloudSim [21], IoTSim [24] and FogNetSim++ [17], have been proposed in recent years to simulate IoT environments [15]. Among them, EdgeCloudSim results to be well-suited for modeling complex IoT systems thanks to the support of architectures of different types and sizes. Moreover, it also supports mobility, which is a typical aspect to be considered when modeling autonomous vehicles scenarios.

These simulation tools are frequently used in the field of autonomous vehicles to perform in-depth tests that take into account the features of the vehicular applications (e.g., upload/download sizes, task description, network models, and mobility) in order to demonstrate how task offloading schemes perform in a realistic IoT scenario. For example, Sonmez et al. [21] proposed a two-stage machine learning-based vehicular edge orchestrator, which considers both task completion success and service time, and used EdgeCloudSim for a detailed performance evaluation. Hossain et al. [10] proposed an efficient dynamic task offloading approach based on a non-cooperative game (NGTO) and assessed its performance on different scenarios via EdgeCloudSim simulator.

Other research effort is devoted to design architectures and models for autonomous vehicles. As an example, Ibn-Khedher [11] designed an end-to-end architecture that enables the allocation of compute-intensive autonomous driving services to edge computing servers, ensuring reliability and low latency. Sasaki [20] presented an edge/cloud computing model for autonomous vehicles based on the Autoware software platform.

1.3 System Architecture

Although cloud computing provides highly scalable and dynamically allocable computing resources, it could have performance issues due to the centralization of the operations of data collection and processing [12, 18]. Using a hybrid edge/cloud architecture could address these issues, enabling a distributed and efficient management of data produced by IoT devices. Even in the context of autonomous vehicles these architectures may improve computation times and scalability, reduce network congestion and task failure rate. Figure 1.1 shows a three-layer edge/cloud architecture that we propose for supporting the management of autonomous vehicles.

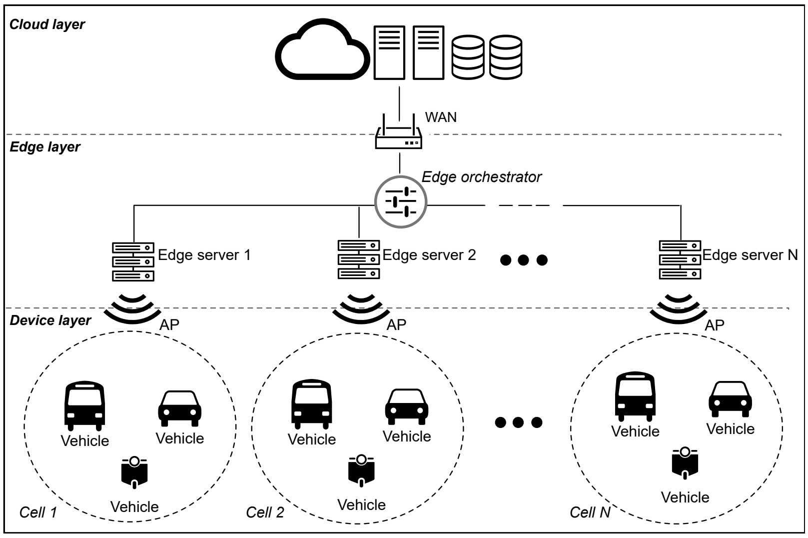


Figure 1.1 The three-layer edge/cloud architecture.

The *device layer* includes the components that are exploited by vehicles to share information during their movements across different urban cells, which define a partitioning of an urban area. These vehicles produce a very high volume of data by using the embedded components (e.g., GPS, infotainment devices, on-board cameras) and send it to the edge server of the current cell. This data can be combined with personal data of the users (e.g., preferences and behaviors) and information about the surrounding environment, so as to offer advanced, customized and context-aware services.

The *edge layer* includes different types of hardware (e.g., Arduino, FPGA, and Raspberry Pi) that make up the infrastructure for gathering the raw data generated at the device layer. The edge layer processes incoming data as long as it has sufficient computing and storage resources. When such resources are no longer sufficient, data is forwarded to the cloud for further processing by the *Edge Orchestrator (EO)*. In particular, the EO is a component that can be configured to apply different orchestration policies in order to improve the performance of the whole architecture. These policies can take into account different parameters, such as the level of network congestion, volume of data to be processed, and workload of edge and cloud. With reference to the specific case study presented in this chapter, two orchestration policies have been used, i.e., *Network Based (EO-NB)* and *Utilization Based (EO-UB)*, capable of efficiently and effectively managing the data produced by autonomous vehicles (for more details see Section 1.4).

Finally, the *cloud layer* represents a large set of computing and storage resources, which can be dynamically allocated, for executing tasks that cannot be performed on behalf of edge servers. From a client perspective, the cloud is an abstraction for remote, infinitely scalable provisioning of computation and storage resources, which has emerged as an effective computing platform to face the challenge of processing big data repositories in limited time, as well as to provide efficient data analysis environments to both researchers and companies [5].

1.4 Performance evaluation

To evaluate the performance of our hybrid edge/cloud architecture for supporting autonomous vehicle driving, we considered three common tasks:

- *Collision avoidance*, which consists in alerting the driver and/or initiating an automated emergency braking action to avoid an accident with a very low-latency response.
- *Route calculation*, which consists in finding routes from a starting location to a given destination.
- *Targeted advertising*, which consists in providing drivers with the most appropriate products or services, such as rest stops, hotels, and restaurants, based on their personal interests and behaviors. Differently from the other two tasks, this results to be a latency-tolerant task.

Specifically, we used the EdgeCloudSim simulator, a Java-based, open-source, and discrete event-based simulator designed for modeling IoT devices, applications, and hybrid edge/cloud architectures. Following the approach proposed in [22], the tasks are generated through a Poisson distribution with different active/idle task generation patterns and interarrival times. A large number of experiments have been carried out using an architecture composed of a cloud and 20 edge servers, each of which provides services to the vehicles that are located on a specific cell of an urban area. In particular, the cloud has been configured as a set of 10 virtual machines (VMs) having

2 cores, 8 GB of RAM and 512 GB of storage memory each. Instead, edge servers have been configured as VMs having 4 cores, 8 GB of RAM and 256 GB of storage memory. In addition, we considered a variable number of autonomous vehicles, ranging from 100 to 1, 500.

The experiments are used to assess the behavior of the hybrid edge/cloud solution compared to centralized ones that exploit only cloud or edge resources. Specifically, the three configurations we evaluated are the following:

- *Cloud-only*: the tasks are performed exclusively on the cloud.
- *Edge-only*: the tasks are performed directly on the edge.
- *Edge/cloud*: the tasks are performed locally on edge servers; however, if the edge servers are unable to complete the tasks (e.g., due to the lack of resources), the edge orchestrator offloads them to the cloud.

The edge orchestrator was configured to distribute the workload between the cloud and the edge using two policies, namely *EO-UB* and *EO-NB*. Specifically, *EO-UB* schedules tasks based on the average utilization of edge nodes. If the average edge utilization is greater than a fixed threshold, the incoming task is offloaded to the cloud; otherwise it is assigned to a generic edge device. On the other hand, *EO-NB* measures the network congestion from the edge device to the cloud to decide where incoming tasks must be performed.

Table 1.1 reports the main simulation parameters along with their description. The actual values used for configuring the EdgeCloudSim simulator are listed in Table 1.2. Instead, Table 1.3 reports the parameter values related to the three tasks described above.

Table 1.1 Description of the main EdgeCloudSim simulation parameters

Parameter	Description
Simulation time	Duration of the simulation in seconds.
Mobile devices	Number of mobile devices used in the simulation scenarios.
Edge servers	Number of edge servers.
MIPS for edge server VM	Computing processor's speed of edge servers in terms of Million Instructions Per Second.
MIPS for cloud VM	Computing processor's speed of cloud in terms of Million Instructions Per Second.
Poisson interarrival	Mean interarrival time between two tasks.
Active period	The active period of the task.
Idle period	The idle period of the task.
Upload data size	Mean input file sizes to upload.
Download data size	Mean output file sizes to download.
Task length	Mean number of instructions to execute the emerging task.

Figure 1.2 reports the performance of the different configurations as the number of vehicles increases. Specifically, Figures 1.2(a), 1.2(b), 1.2(c) and

Table 1.2 EdgeCloudSim simulation parameters

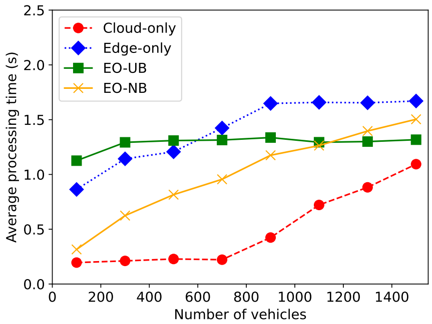
Parameter	Value
Simulation time (min.)	36
Mobile devices (i.e., vehicles)	100-1500
Edge servers	20
MIPS for edge server VM	10,000
MIPS for cloud VM	37,500

Table 1.3 Parameter values of the three tasks

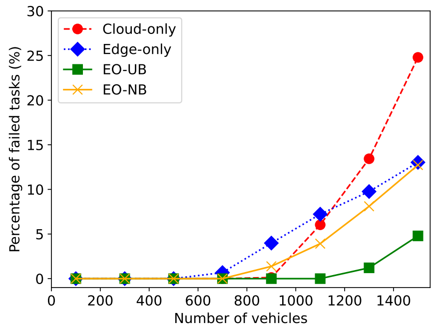
Parameter	Task		
	Collision Avoidance	Route Calculation	Targeted Advertising
Poisson interarrival	5	3	15
Active period	3600	3600	3600
Idle period	1	1	15
Upload data size	40 KB	20 KB	20 KB
Download data size	20 KB	80 KB	20 KB
Task length	10,000 MIPS	3,000 MIPS	20,000 MIPS

1.2(d) report the average processing time, percentage of failed tasks, network delay and VM utilization, respectively. As a general consideration for evaluating results, low percentage of failed tasks and low network delay are essential to ensure a high quality of services (e.g., in terms of reliability and responsiveness). Also reducing VM utilization is a crucial aspect in large-scale applications, as it enables cost and energy consumption optimization, as well as handling any unexpected workload spikes.

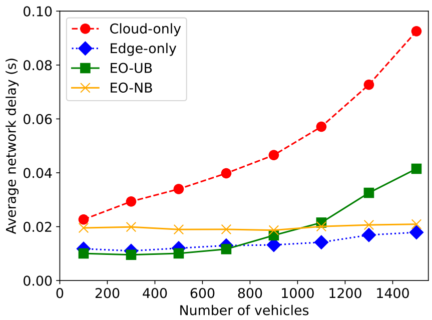
About the results obtained, the edge-only solution achieved the best results in terms of network delay, but with high processing times, failure rate and VM utilization. In fact, the average processing time ranges from 0.86 seconds with 100 vehicles to 1.67 seconds with $1,500$ vehicles, the percentage of failed tasks reaches up to 13%, and the VM utilization goes up to 68%. The cloud-only solution achieved a very low average processing time and VM utilization, but it drastically increased the percentage of failed tasks (25% for $1,500$ vehicles). Regarding the network usage, data transfer directly to the cloud caused a significant increase of delay (80% vs edge-only). Instead, the hybrid edge/cloud solution turned out to provide the best compromise among all the performance metrics considered. In fact, both orchestration policies ensured good processing times, low percentages of failed tasks and VM utilization, and limited network delays. However, if the number of vehicles is high, the EO-UB policy turned out to be the best choice, as it reached lower processing times and percentage of failed tasks.



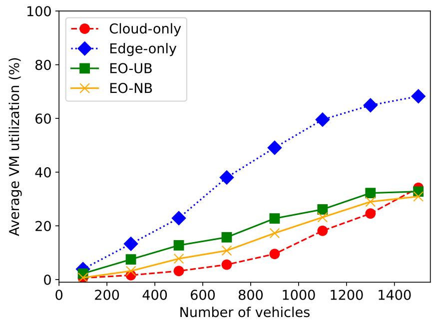
(a) Processing time.



(b) Failed tasks.



(c) Network delay.



(d) VM utilization.

Figure 1.2 Performance results of the tasks related to autonomous vehicle driving obtained for the different simulation scenarios.

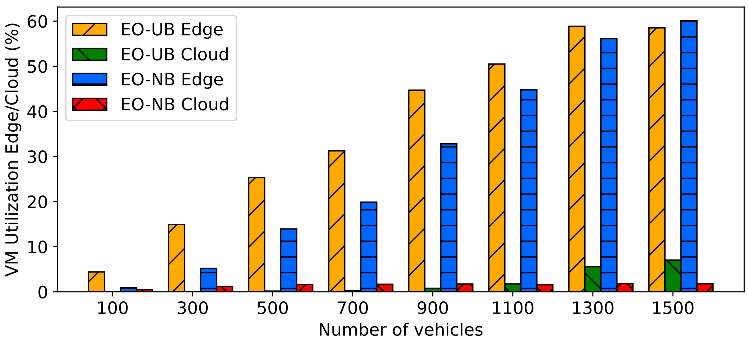


Figure 1.3 Comparison of EO-UB and EO-NB policies about the VM utilization on the edge and cloud.

In order to understand how computing resources are exploited by the hybrid edge/cloud solution with the two orchestration policies, Figure 1.3 shows the percentage of VM utilization for the cloud and edge. As shown, the two orchestration policies attempt to make the most of the edge layer's resources, employing the cloud only when necessary, and as a result are able to reduce both latency and percentage of failed tasks.

Overall, the presented architecture and orchestration policies outperformed the conventional cloud- or edge-only approaches, considerably improving processing times and percentage of failed tasks. Furthermore, they ensure better management of workload spikes because of a more efficient resource utilization.

1.5 Conclusions

In recent years, the use of IoT solutions enabled the processing of data closer to where it is generated, so as to reduce network traffic and task failure rates. Generally, IoT infrastructures are composed of many heterogeneous components that interact with each other to pursue common goals. Designing, testing and deploying modern IoT infrastructures are going to be a great challenge in the next few years.

The chapter presented a hybrid edge/cloud solution to efficiently manage tasks concerning autonomous vehicles driving in a large-scale IoT computing environment. In particular, two orchestration policies were used for better balancing the workload between the edge and the cloud. Numerous experiments have been carried out to evaluate the behavior of the proposed solution compared to the centralized ones that exploit only cloud or edge resources. The results showed that our solution ensures a good compromise between processing time, network delay and task failure rate compared to conventional centralized architectures.

Authors

Loris Belcastro is a research fellow of computer engineering at the University of Calabria, Italy. He received a Ph.D. in Information and Communication Engineering at the University of Calabria. In 2012 he held a scholarship at the Institute of High-Performance Computing and Networking of the Italian National Research Council (ICAR-CNR). His research interests include cloud computing, social media and big data analysis, distributed and parallel computing, and edge computing.

Fabrizio Marozzo is an assistant professor of computer engineering at the University of Calabria. He received a Ph.D. in Systems and Computer Engineering at the University of Calabria. In 2011-2012 he visited the Barcelona SuperComputing Center for a research internship with the Grid Computer Research group in Computer Sciences department. He is a member of the editorial boards of IEEE Access, Journal of Big Data, SN Computer Science, Scalable Computing: Practice and Experience, Big Data and Cognitive Computing, Algorithms and Heliyon. His research focuses on big data analysis, social media analysis, parallel and distributed computing, cloud and edge computing, and machine learning.

Alessio Orsino is a PhD student in Information and Communication Technologies at the University of Calabria, Italy. His research interests include big data analysis, parallel and distributed computing, cloud and edge computing, and machine learning.

References

[1] Ejaz Ahmed, Ibrar Yaqoob, Ibrahim Abaker Targio Hashem, Imran Khan, Abdelmuttlib Ibrahim Abdalla Ahmed, Muhammad Imran, and Athanasios V Vasilakos. The role of big data analytics in internet of things. *Computer Networks*, 129:459–471, 2017.

[2] Alessandro Barbieri, Fabrizio Marozzo, and Claudio Savaglio. Iot platforms and services configuration through parameter sweep: a simulation-based approach. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1803–1808, 17-20 October 2021.

[3] Loris Belcastro, Riccardo Cantini, Fabrizio Marozzo, Alessio Orsino, Domenico Talia, and Paolo Trunfio. Programming big data analysis: Principles and solutions. *Journal of Big Data*, 9(4), 2022.

[4] Loris Belcastro, Alberto Falcone, Alfredo Garro, and Fabrizio Marozzo. Evaluation of large scale roi mining applications in edge computing environments. In *25th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, pages 1–8, 2021.

[5] Loris Belcastro, Fabrizio Marozzo, Domenico Talia, and Paolo Trunfio. Big data analysis on clouds. In Albert Zomaya and Sherif Sakr, editors, *Handbook of Big Data Technologies*, pages 101–142. Springer, December 2017.

[6] Stig Bosmans, Siegfried Mercelis, Joachim Denil, and Peter Hellinckx. Testing iot systems using a hybrid simulation based testing approach. *Computing*, 101(7):857–872, 2019.

[7] Gabriele D'Angelo, Stefano Ferretti, and Vittorio Ghini. Simulation of the internet of things. In *2016 International Conference on High Performance Computing & Simulation (HPCS)*, pages 1–8. IEEE, 2016.

[8] Gabriele D'Angelo, Stefano Ferretti, and Vittorio Ghini. Distributed hybrid simulation of the internet of things and smart territories. *Concurrency and Computation: Practice and Experience*, 30(9):e4370, Nov 2017.

[9] Harshit Gupta, Amir Vahid Dastjerdi, Soumya K Ghosh, and Rajkumar Buyya. 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.

[10] Md. Delowar Hossain, Tangina Sultana, Md. Alamgir Hossain, Md. Abu Layek, Md. Imtiaz Hossain, Phoo Pyae Sone, Ga-Won Lee, and Eui-Nam Huh. Dynamic task offloading for cloud-assisted vehicular edge computing networks: A non-cooperative game theoretic approach. *Sensors*, 22(10), 2022.

[11] Hatem Ibn-Khedher, Mohammed Laroui, Mouna Ben Mabrouk, Hassine Moungla, Hossam Afifi, Alberto Nai Oleari, and Ahmed E. Kamal. Edge computing assisted autonomous driving using artificial intelligence. In *2021 International Wireless Communications and Mobile Computing (IWCMC)*, pages 254–259, 2021.

[12] Niloofar Khanghahi and Reza Ravanmehr. Cloud computing performance evaluation: issues and challenges. *Comput*, 5(1):29–41, 2013.

[13] Somayya Madakam, Vihar Lake, Vihar Lake, Vihar Lake, et al. Internet of things (iot): A literature review. *Journal of Computer and Communications*, 3(05):164, 2015.

[14] Mohsen Marjani, Fariza Nasaruddin, Abdullah Gani, Ahmad Karim, Ibrahim Abaker Targio Hashem, Aisha Siddiqa, and Ibrar Yaqoob. Big iot data analytics: architecture, opportunities, and open research challenges. *ieee access*, 5:5247–5261, 2017.

[15] Andras Markus and Attila Kertesz. A survey and taxonomy of simulation environments modelling fog computing. *Simulation Modelling Practice and Theory*, 101:102042, 2020.

[16] Fabrizio Marozzo, Alessio Orsino, Domenico Talia, and Paolo Trunfio. Edge computing solutions for distributed machine learning. In *The 8th IEEE International Conference on Cloud and Big Data Computing*, 2022. To appear.

[17] Tariq Qayyum, Asad Waqar Malik, Muazzam A Khan Khattak, Osman Khalid, and Samee U Khan. Fognetsim++: A toolkit for modeling and simulation of distributed fog environment. *IEEE Access*, 6:63570–63583, 2018.

[18] V Krishna Reddy, B Thirumala Rao, and LSS Reddy. Research issues in cloud computing. *Global Journal of Computer Science and Technology*, 2011.

[19] Maria Salama, Yehia Elkhatib, and Gordon Blair. Iotnetsim: A modelling and simulation platform for end-to-end iot services and networking. In *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing*, pages 251–261, 2019.

[20] Yu Sasaki, Tomoya Sato, Hiroyuki Chishiro, Tasuku Ishigooka, Satoshi Otsuka, Kentaro Yoshimura, and Shinpei Kato. An edge-cloud computing model for autonomous vehicles. In *In 11th IROS Workshop on Planning, Perception, Navigation for Intelligent Vehicle*, 2019.

[21] Cagatay Sonmez, Atay Ozgovde, and Cem Ersoy. Edgecloudsim: An environment for performance evaluation of edge computing systems. *Transactions on Emerging Telecommunications Technologies*, 29(11):e3493, 2018.

[22] Cagatay Sonmez, Can Tunca, Atay Ozgovde, and Cem Ersoy. Machine learning-based workload orchestrator for vehicular edge computing. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[23] Saniya Zahoor and Roohie Naaz Mir. Resource management in pervasive internet of things: A survey. *Journal of King Saud University-Computer and Information Sciences*, 2018.

[24] Xuezhi Zeng, Saurabh Kumar Garg, Peter Strazdins, Prem Prakash Jayaraman, Dimitrios Georgakopoulos, and Rajiv Ranjan. Iotsim: A simulator for analysing iot applications. *Journal of Systems Architecture*, 72:93–107, 2017.