



HAL
open science

A fuzzy based approach for managing smart edge-enhanced IoT devices in mission-critical applications

Carol Habib, Nathalie Mitton

► To cite this version:

Carol Habib, Nathalie Mitton. A fuzzy based approach for managing smart edge-enhanced IoT devices in mission-critical applications. 28th Conference on Innovation in Clouds, Internet and Networks, Mar 2025, Paris, France. ⟨hal-04946021⟩

HAL Id: hal-04946021

<https://hal.science/hal-04946021v1>

Submitted on 17 Feb 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

A fuzzy based approach for managing smart edge-enhanced IoT devices in mission-critical applications

Carol Habib
Inria, France
carol.habib@inria.fr

Nathalie Mitton
Inria, France
nathalie.mitton@inria.fr

Abstract—In the last decade, edge computing emerged as a paradigm allowing close-to-the-source processing. It brings multiple benefits for mission-critical applications especially that they cannot tolerate delays, downtime or failure. Particularly, in post-disaster management, where the network is scarce and access to the Internet might not be possible, edge servers can be embarked on the robots that are exploring the disaster area transforming them to edge-enhanced devices. When required, they run resource-intensive tasks and provide local decision-making to other constrained devices in the disaster area such as a Wireless Sensor Network (WSN). The robots are battery-powered and are used for a mission-critical application where sensitivity and low tolerance for delays are crucial. Therefore, these edge-enhanced devices must be properly managed to meet the needs of the application. In this paper, a Fuzzy Inference System (FIS)-based approach is proposed enabling robots to decide, whenever a resource-intensive task must be executed, whether they can stop exploring the disaster area and act as edge servers. Six parameters reflecting the status of the network and the application are used by the FIS for the decision-making. Preliminary simulation results show that, in the proposed approach, the energy consumption in the network is 1.7 times less than the baseline network.

Index Terms—Post-disaster management, Wireless Sensor Networks, Edge-enhanced devices, Fuzzy Inference System

I. INTRODUCTION

Applications that require high availability, low latency, security, scalability and resilience are called mission-critical applications. In such applications, downtime, failure or delays can have severe consequences such as threats to human safety, significant financial losses or damage to infrastructure [1]. Examples of mission-critical applications include healthcare systems, autonomous vehicles, industrial automation and public safety and emergency services. In these applications, the IoT-Cloud architecture is not the best candidate for multiple reasons. First, high latency and delay, that are introduced by the remote location of the cloud from IoT devices, are intolerable in time-sensitive applications [2]. Second, the amount of data collected by these end devices impact further the efficiency of the network and create bottlenecks when task processing is centralized and not fast enough. Third, in some mission-critical applications such as post-disaster management, the IoT network can be damaged or scarce resulting in an unreachable cloud. For instance, in post-disaster management applications, edge resources can be embedded on

mobile devices such as unmanned aerial vehicles (UAVs) or robots exploring the disaster area [3], [4]. It is important to note that these local servers are limited in resources. Therefore, they must be properly managed in order to efficiently handle the requests of users [1]. Moreover, these devices do not only act as edge servers, providing computing resources and close to the source decision-making, but they are also equipped with sensors (such as cameras, temperature, humidity, Infrared, etc.) and can collect data. Hence, resource management also includes the management of these devices in terms of the role assigned to them in the network. In fact, these devices change roles throughout the mission based on the identified needs and the available resources. Their default role is to explore the region in which the disaster occurred. But, they can exit this mode and become edge servers when resource-intensive tasks must be executed: enabling by that close to the source and timely decision-making and data processing. Thus, the following problem is identified: How can the edge server equipped devices, which we refer to edge-enhanced devices in the rest of the paper, in a post-disaster management application, be efficiently used to explore the disaster area and act as edge servers without compromising the mission ? In this paper, we propose a Fuzzy Inference System (FIS) based approach allowing edge-enhanced devices, to decide in real-time whether they can become edge servers. For this purpose, we define six parameters describing the status of the network and the mission: reachability to data sources, available energy, network efficiency, network delay, exploration coverage and critical level. Each time that a resource-intensive task is identified, the six parameters are measured and then, using the FIS, the decision is inferred. The main contributions of this paper can be summarized as follows:

- The definition of six decision-making parameters reflecting the requirements imposed by the network and the application. These inputs are used by edge-enhanced devices to decide if they can stop exploration and start computation.
- A FIS allowing edge-enhanced devices to switch from exploration to computation mode. The FIS uses the six decision-making parameters to produce a decision.

The rest of the paper is organized as follows. Section II

provides a background on the different strategies of resource management techniques at the edge. Section III describes the envisaged scenario and highlights our motivation. Section IV presents the proposed approach. Simulation results are presented in Section V. Finally, Section VI brings conclusions and suggests future work.

II. BACKGROUND AND RELATED WORK

A. Resource management at the edge

Multi-access Edge Computing (MEC) leverages edge nodes located closer to end users to process and analyze data locally. As mentioned in the introduction, this proximity reduces latency and supports real-time applications. Several works in literature have targeted different aspects of resource management at the edge: task placement [5], task offloading [4], [6], [7], task scheduling [8] and resource allocation at the edge server [9]. In [9], a multi-stacked reinforcement learning algorithm is proposed to select which edge node can process a task taking into account the required computing resources by the task. In [4], an Artificial Neural Fuzzy Inference System is proposed to decide for any given task if it should be processed locally, or offloaded to an edge server or the cloud. The decision-making parameters considered are bandwidth, CPU utilization, task length and task size. In [7], a meta reinforcement learning based approach is proposed to offload tasks in a mobile edge computing network. However, in all the aforementioned works, energy constrained edge servers are not considered. Moreover, edge servers are not assigned another role than task processing in the network. Application requirements regarding the performance of the network are not taken into account. In [10], an approach is proposed to assign the robot that is the closest and most suited for the identified task for task computation. The problem is formulated using Markov Decision Process and is solved using deep reinforcement learning. However, this approach is centralized and requires a server in proximity to robots to assign tasks among them. In this work, we propose a decentralized and lightweight algorithm allowing edge-enhanced devices, that have an application related primary role, to decide if they can act as edge servers when a resource-intensive task needs to be processed. Specifically, we consider robots exploring a disaster area as edge-enhanced devices and take as a use a post-disaster management application. The proposed algorithm takes into account the limited energy budget of robots, network constraints and application requirements.

III. SCENARIO AND MOTIVATION

As a specific use case, we consider a post-disaster scenario where multiple stakeholders are present and in urgent need of getting information about the environment, the presence of people, network statistics, maps of the area and sensing information [11]. Multiple first responder teams such as firefighter brigades, police officers and first-aid teams approach the area with high-tech devices such as drones, ground robots, wireless sensor nodes and computing facilities. We suppose that one of these teams sends a group of robots to explore the

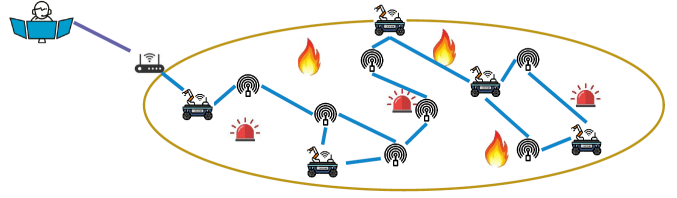


Fig. 1. Network architecture in post-disaster management application

region in which the disaster happened. These robots can deploy sensor nodes during the mission based on the identified needs: need for relay nodes or sensing nodes. In the rest, we refer to the sensing nodes as data sources. Given that the network can be scarce or damaged, we assume that the robots and the wireless sensor nodes can maintain connectivity among each other and create an ad-hoc Wireless Sensor Network (WSN) (c.f. Figure 1). These robots are edge-enhanced devices: each time that a resource-intensive task must be executed, they can become edge servers. Examples of resource-intensive tasks can be image classification, object detection, victim detection, network assessment, area assessment etc. We assume that all robots are identical in terms of resources and the set of tasks they can run. For each task T , a set of required data sources is predefined and a level of criticality is assigned. These edge-enhanced devices not only offer resources for data computation and storage but also generate data that can enhance the overall situational awareness. Thus, they are also considered to be data sources. Indeed, in such scenarios, the edge servers cannot be continuously processing resource-intensive tasks especially that the robots are battery powered and must smartly manage their resources to not compromise the mission.

IV. THE PROPOSED METHOD

A. Decision-making parameters

In this section, the six decision-making parameters which reflect the network and application status are presented. These parameters are the inputs of the proposed FIS.

Reachability to data sources: In order for the robot to run resource-intensive tasks, it requires data coming from specific data sources. The latter include sensor nodes in the network or sensors placed on the robots. Reachability can be represented then, by the total of hops between the robot and the data source. It is calculated as follows:

$$Reach = \begin{cases} 0 & \text{if } total_hops_i = 0, \\ \min_{i=1}^N (1/total_hops_i) & \text{if } total_hops_i < th, \\ 1/Reach_th & \text{otherwise.} \end{cases}$$

where N represents the total of data sources required for the task, $Reach_th$ is a predefined maximum hops threshold and such as $Reach \in [0, 1]$. The closer the data source is to the edge-enhanced device, the closer the value of $Reach$ gets to 1.

Available energy: In order for the robot to run resource-intensive tasks, it requires enough energy resources. It is calculated as follows :

$$Energy = Current_battery_level / Initial_battery_level$$

such as $Energy \in [0, 1]$. The more energy budget the edge-enhanced device has, the closer the value of $Energy$ gets to 1.

Network efficiency: In order for the robot to run resource-intensive tasks, it should receive data from required sources without data loss due to congestion. A robot is in a good position in the network if data packets from all sources can reach it with good efficiency. The crisp value is represented by the Packet Delivery Ratio (PDR) and is calculated as follows:

$$PDR = \begin{cases} 1 & \text{if data source} \\ \text{is on robot,} & \\ \min_{i=1}^N \left(\frac{Total_Packets_Delivered_i}{Total_Packets_Sent_i} \right) & \text{otherwise.} \end{cases}$$

where N is the total of data sources and such that $PDR \in [0, 1]$. The greater the measured PDR value for the poorest data source in terms of networking conditions, the closer the PDR value is to 1.

Network delay: In order for the robot to resource-intensive tasks, it should receive data from required sources within an acceptable delay. Usually, the parameter *Network Efficiency* is an indicator of the delay parameter. If PDR is low then delay is short. However, in some Quality of Service (QoS) policies, the network may prioritize high throughput for certain types of traffic while allowing longer delays for others. For example, real-time communication may be assigned a high priority for low delay, while other data transfers may be optimized for maximum throughput, potentially allowing longer delays. The network delay is calculated as follows :

$$delay = \max_{i=1}^N (delay_i / delay_th)$$

where $delay_th$ is a predefined maximum acceptable delay for the identified task and such that $delay \in [0, 1]$. If $delay_i > delay_th$ then it is set to $delay_th$. The longer the delay, the closer the value of $delay$ is to 1.

Exploration coverage: In order for the robot to run resource-intensive tasks, it should know if it can exit exploration mode. It can exit the mode, if there are enough robots exploring the unexplored region. We determine the exploration coverage by getting the number of robots that are exploring the unexplored region.

Usually in a post-disaster scenario, we do not have an idea about the surface of the region to explore. In that case, we are going to define its value as follows:

$$Cov = R_{exploration} / R$$

where $R_{exploration}$ is the total robots in exploration, R is the total of robots in mission, and such that $Cov \in [R_{min}/R, 1]$ where R_{min} is the minimum threshold of exploring robots

TABLE I
FUZZY INPUT VARIABLES WITH THEIR CORRESPONDING FUZZY SETS AND RANGES.

Input Variable	Notation	Fuzzy Set	Ranges
Reachability to data sources (hops)	Reach	Far Close	0-0.5 0.5-1
Available Energy (Joules)	Energy	Low High	0-0.5 0.5-1
Network Efficiency (%)	PDR	Low High	0-0.5 0.5-1
Network delay (s)	delay	Short Long	0-0.5 0.5-1
Exploration Coverage (%)	Cov	Insufficient Sufficient	0-0.5 0.5-1
Critical level of application (score)	Critical	Low High	0-0.5 0.5-1

that should be satisfied throughout the whole duration of the mission and R is the total of robots in mission. The more robots there are in exploration mode, the closer the value of Cov is to 1.

Critical level of application: In order for the robot to run resource-intensive tasks, it should take into consideration the level of criticality of the application to which the task belongs. The variable *Critical* has a predefined value ranging from 0 to 1 based on each application. The higher the value, the more the application is critical.

B. Fuzzy Sets and Membership functions

As shown in Table I, for each decision-making parameter, two fuzzy sets were defined. Thus, two sigmoid membership functions per parameter are required. The membership functions μ_{Far} , μ_{Low} , μ_{Low} , μ_{Short} , $\mu_{Insufficient}$, μ_{Low} are defined as follows:

$$\mu_{1,i}(x_i) = \frac{1}{1 + e^{10(x_i - 0.5)}}$$

where x_i is the input variable of the i^{th} fuzzy set. Whereas, the membership functions μ_{Close} , μ_{High} , μ_{High} , μ_{Long} , $\mu_{Sufficient}$, μ_{High} are defined as follows:

$$\mu_{2,i}(x_i) = \frac{1}{1 + e^{-10(x_i - 0.5)}}$$

where x_i is the input variable of the i^{th} fuzzy set.

The same membership functions were also defined for the fuzzy sets "Low" and "High" for the output parameter *Edge server* of the FIS.

C. Rule base

For any FIS, a rule base must be defined. The following condition has been considered in the construction of the rules: if at least three decision-making parameters are satisfied, then the robot becomes an edge server. Furthermore, the only operator used to form the rules is the "AND" operator. This yields to having a total of 64 rules combining all of the six decision-making parameters. An example of a rule is the following : *If Reachability is Close AND Available energy is High AND Network efficiency is High AND Delay is Short AND Exploration coverage is Insufficient AND Critical is Low then Edge server is High.*

TABLE II
THE NETWORK CONFIGURATION USED FOR SIMULATION IN NS-3

Size of Area	(50m, 50m)
Position of nodes	Random and fixed
Communication Technology	WiFi
Networking mode	Ad hoc
Routing protocol	OLSR
Application	UDP (448 Kb/s)

V. PRELIMINARY SIMULATION RESULTS

In order to evaluate our approach we have conducted a series of simulations using the ns-3 simulator. We consider a WSN composed of 10 nodes such as nodes 0, 1 and 2 are edge-enhanced devices and nodes 3, 4 and 5 are considered to be data sources for resource-intensive tasks that the edge-enhanced devices are capable of running. The default mode of the edge-enhanced devices is exploration mode. They are simply exploring the region and collecting or relaying data. The WSN is configured as shown in Table II. In each conducted simulation, a random number of resource-intensive tasks is generated and a random positioning of nodes within the fixed area size is selected. For each generated simulation setup, we compare the following two scenarios:

- **Scenario 1:** This scenario represents the baseline or default behavior. The edge-enhanced devices upon detecting the need to run resource-intensive tasks, switch immediately from exploration mode to edge mode. The nodes that are defined as data sources to run these tasks start sending data packets to the edge node at a data rate of 488 Kb/s.
- **Scenario 2:** This scenario represents our proposal. The edge-enhanced devices upon detecting the need to run resource-intensive tasks, measure the six parameters described in Section IV-A and using the FIS decide whether they can switch to edge mode or not. Only when the output of the FIS is greater than 0.5, the decision is YES.

The total duration of one simulation is set to 2 mins. For a better reporting of the obtained results, we categorize the conducted simulations into 2 categories based on the values of the six decision-making parameters estimated throughout the simulations. These categories describe different combinations of network and application statuses as shown in Table III. For both statuses, data sources are reachable from the edge-enhanced devices. In other words, this parameter is always satisfied. The exploration coverage value changes over time during each simulation. This value is dependent of the number of edge-enhanced devices that are in exploration mode at the moment of making a decision using the FIS. In other words, this parameter is not always satisfied. In status 1, the conducted simulations have all moderate network conditions and random critical levels as well. However, the edge-enhanced devices have low energy budget. This is represented by a crisp value less than 0.3. In other words, this parameter is never satisfied throughout the whole duration of the simulations. Finally, in status 2, the conducted simulations have all poor network

conditions. This is represented by a crisp value less than 0.3. However, the edge-enhanced devices have a good energy budget. In the following, we compare scenarios 1 and 2 in terms of:

- The amount of resource-intensive tasks executed by edge-enhanced devices;
- The energy consumption in the WSN;

in order to evaluate the performance of the proposed approach and its impact on the network and application.

A. Amount of resource-intensive tasks executed

During each simulation, a random number of resource-intensive tasks are identified. The baseline behavior of edge-enhanced devices implies that all tasks are executed. Thus, at the end of the simulation, all edge-enhanced devices could potentially be in edge mode, receiving packets from data sources to process tasks, and none would be exploring the area. As shown in Figure 2, it is clear to see that in the proposed approach (Scenario 2), not all resource-intensive tasks are necessarily executed. In the first subplot of Figure 2, we study the impact of having low energy level on the decision-making process. As the results show, in all simulations, the total of executed tasks is reduced compared to the baseline. Indeed, the FIS gives an output greater than 0.5 only if the task is highly critical and there is enough exploration coverage. Therefore, all critical tasks (> 0.5) are executed in the reported experiments and all the low critical tasks are not executed. A total of 13 tasks are executed out of 24 tasks. This surely has a positive impact on the energy budget of the edge node where energy is preserved for critical tasks when network conditions are moderate. In the second subplot of Figure 2, we study the impact of having poor network conditions in terms of network efficiency and delay on the decision-making process. Given the rule that has been followed to construct the rule base, at least 4 input parameters need to be not satisfied in order to have a "No" as an output. Therefore, given that only 2 parameters are not satisfied in this scenario, the FIS is insensitive and will always produce an output greater than 0.5 unless the varying input parameter *Exploration coverage* becomes very low and coincides with a low critical task (which value was randomly generated in simulations). One way to address these situations is by adjusting the threshold value used by the edge-enhanced device to compare against the output of the FIS. For the reported simulations, we set the threshold value to 0.7. As a result a total of 10 tasks were executed from the 24 identified resource-intensive tasks.

B. Energy Consumption

Using the energy model provided in [6] (c.f. Table IV), the total energy consumption of the WSN can be estimated as follows:

$$E_{tot} = E_{radio} + E_{comp}$$

where E_{radio} is the energy consumed by the communication units and E_{comp} is the energy consumed by the processing

TABLE III
THE DETAILS OF THE TWO NETWORK AND APPLICATION STATUSES CONSIDERED FOR SIMULATION

Status	Description	Available energy	Network efficiency	Network delay	Critical level
1	Moderate Network conditions & Low amount of energy	<0.3	0.5	0.5	Random
2	Poor Network Conditions	>0.7	<0.3	>0.7	Random

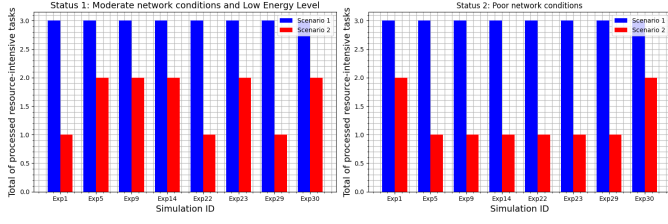


Fig. 2. Comparison of the amount of executed resource-intensive tasks by edge-enhanced devices between the proposed approach (Scenario 2) and the baseline network behaviour (Scenario 1).

TABLE IV
PARAMETER VALUES OF THE ENERGY MODEL

Parameter	Value	Parameter	Value
I_{TX}	17.4 mA	V	3.3
I_{RX}	19.4 mA	ϵ_{add}	2.13 nJ
T_{TX}	1.78×10^{-5} s	ϵ_{mul}	6.39 nJ
T_{RX}	1.78×10^{-5} s	ϵ_{cmp}	2.13 nJ

units. The energy E_{radio} required for transmitting k bytes per hop can be expressed as follows:

$$E_{radio}(k) = kV(I_{TX}T_{TX} + I_{RX}T_{RX})$$

Whereas, the energy E_{comp} required to execute the proposed approach can be expressed as follows:

$$E_{comp} = N_{add}\epsilon_{add} + N_{mul}\epsilon_{mul} + N_{cmp}\epsilon_{cmp}$$

where N is the number of operations and ϵ_{add} , ϵ_{mul} and ϵ_{cmp} are the energy consumption of addition, multiplication and comparison. The energy consumed by the WSN in Scenario 1 (65.36 J) is approximately 1.7 times more than the energy consumed for radio communication by the WSN in the proposed Scenario 2 (Statuses 1 and 2: 38.13 J and 40 J respectively). This is due to the fact, that in the former case, all resource-intensive tasks that are detected, over a given period of time, by the edge-enhanced devices are executed leading to more data packets exchanged between nodes acting as data sources and edge-enhanced devices. Concerning the energy consumption for computation, we estimate the energy required to run the FIS using the energy model as shown in Table IV. The required energy is negligible compared to the energy required for transmitting k bytes over 1 hop and is approximately $4.12\mu\text{J}$.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a FIS based approach allowing edge-enhanced devices in a post-disaster management

use case to decide whether they can stop exploration and act as edge servers. Six parameters reflecting the network and application statuses were proposed and served as inputs for decision-making. The results show that the proposed approach reduces the amount of processed tasks by privileging critical tasks and maintaining a required exploration ratio imposed by the mission. As a consequence, energy consumption in the network is reduced. As future work, we aim to improve the FIS in terms of sensitivity by better refining the threshold value at the output, the weights assigned to the inputs and the rules. These parameters could be learned throughout the mission.

ACKNOWLEDGMENT

This work was funded by the European Union's Horizon Europe research and innovation program under grant agreement No. 101070487 (NEPHELE).

REFERENCES

- [1] N. Santi and N. Mitton, "A resource management survey for mission critical and time critical applications in multi access edge computing," *ITU Journal on Future and Evolving Technologies*, vol. 2, no. 2, 2021.
- [2] N. M. Quy, L. A. Ngoc, N. T. Ban, N. V. Hau, and V. K. Quy, "Edge computing for real-time internet of things applications: Future internet revolution," *Wireless Personal Communications*, vol. 132, no. 2, pp. 1423–1452, 2023.
- [3] J. Liu and Q. Zhang, "Computation resource allocation for heterogeneous time-critical iot services in mec," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2020, pp. 1–6.
- [4] M. K. Ibrahim, A. Sajid, I. Ullah, T. Ali, M. Ayaz, and E.-H. M. Aggoune, "Artificial neural fuzzy inference rule based (anfis) model for offloading tasks for edge, cloud and uavs environment," *IEEE Access*, 2024.
- [5] S. Yang, F. Li, M. Shen, X. Chen, X. Fu, and Y. Wang, "Cloudlet placement and task allocation in mobile edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5853–5863, 2019.
- [6] N. Zenasni, C. Habib, and J. Nassar, "A fuzzy logic based offloading system for distributed deep learning in wireless sensor networks," in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–8.
- [7] J. Wang, J. Hu, G. Min, A. Y. Zomaya, and N. Georgalas, "Fast adaptive task offloading in edge computing based on meta reinforcement learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 1, pp. 242–253, 2020.
- [8] B. Lin, C. Lin, X. Chen, M. Lin, G. Huang, and Z. Xu, "Cost-driven scheduling for workflow decision making systems in fuzzy edge-cloud environments," *IEEE Transactions on Automation Science and Engineering*, 2024.
- [9] S. Wang, M. Chen, X. Liu, C. Yin, S. Cui, and H. V. Poor, "A machine learning approach for task and resource allocation in mobile-edge computing-based networks," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1358–1372, 2020.
- [10] M. Afrin, J. Jin, A. Rahman, S. Li, Y.-C. Tian, and Y. Li, "Dynamic task allocation for robotic edge system resilience using deep reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.
- [11] L. Militano, A. Arteaga, G. Toffetti, and N. Mitton, "The cloud-to-edge-to-iot continuum as an enabler for search and rescue operations," *Future Internet*, vol. 15, no. 2, p. 55, 2023.