Evaluation of Large Scale RoI Mining Applications in Edge Computing Environments

Conference Paper · August 2021

4 authors:

L. Belcastro
Università della Calabria
30 PUBLICATIONS **179** CITATIONS

SEE PROFILE

Alberto Falcone
Università della Calabria
36 PUBLICATIONS **250** CITATIONS

SEE PROFILE

Alfredo Garro
Università della Calabria
116 PUBLICATIONS **1,338** CITATIONS

SEE PROFILE

Fabrizio Marozzo
Università della Calabria
81 PUBLICATIONS **817** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project ASPIDE View project

Project HLA Development Kit Framework View project

Evaluation of Large Scale RoI Mining Applications in Edge Computing Environments

Loris Belcastro, Alberto Falcone, Alfredo Garro, Fabrizio Marozzo

Department of Informatics, Modeling, Electronics and Systems Engineering (DIMES)
University of Calabria
via P. Bucci 41C, 87036 Rende (CS), Italy
{loris.belcastro, alberto.falcone, alfredo.garro, fabrizio.marozzo}@dimes.unical.it

Abstract—**Researchers and leading IT companies are increasingly proposing hybrid cloud/edge solutions, which allow to move part of the workload from the cloud to the edge nodes, by reducing the network traffic and energy consumption, but also getting low latency responses near to real time. This paper proposes a novel hybrid cloud/edge architecture for efficiently extracting Regions-of-Interest (RoI) in a large scale urban computing environment, where a huge amount of geotagged data are generated and collected through users's mobile devices. The proposal is organized in two parts: (i) a modeling part that defines the hybrid cloud/edge architecture capable of managing a large number of devices; (ii) a simulation part in which different design choices are evaluated to improve the performance of RoI mining algorithms in terms of processing time, network delay, task failure and computing resource utilization. Several experiments have been carried out to evaluate the performance of the proposed architecture starting from different configurations and orchestration policies. The achieved results showed that the proposed hybrid cloud/edge architecture, with the use of two novel orchestration policies (network- and utilization-based), permits to improve the exploitation of resources, also granting low network latency and task failure rate in comparison with other standard scenarios (only-edge or only-cloud).**

Index Terms—**IoT platforms, Cloud/Edge architecture, Modeling and Simulation, Orchestration policies, RoI mining**

I. Introduction

In the last years the ability to produce and gather data has increased exponentially. Huge amounts of digital data are generated by and collected from a plethora of Internet of Things (IoT) devices, such as sensors, cams, in-vehicle infotainment and wearable devices [1], [2]. Such data, after being appropriately collected, cleaned and analyzed, are extremely useful in many fields of application such as security, trade and monitoring. As an example, geotagged data generated by mobile devices can be used to find the Regions-of-Interest (RoIs), i.e., the most visited regions by users. The analysis of RoIs is highly valuable in many scenarios, e.g.: tourism agencies and municipalities can discover the most visited touristic places and the time of year when such places are visited; transport operators can discover the places and routes where is it more likely to serve passengers and crowed areas where more transport facilities need to be allocated [3].

Nowadays, existing IoT platforms used for processing data from IoT devices are highly centralized and rely on cloud solutions for data collection, integration and analysis. The data management and processing approach in the cloud can be ineffective in terms of network traffic management, response time and power consumption. For example, in many medical and security applications, it is essential to offer low-latency services, as the delay caused by the transfer of data from an application to the cloud (and vice versa) can cause strong disservices and even loss of life. For this reason, researchers and large companies have proposed in recent years the use of IoT solutions, which allows to process data closer to where they are generated, in order to reduce network traffic and energy consumption and also getting low latency responses near to real time.

Even the analysis of data generated by devices in IoT platforms requires novel technologies for processing and analyzing large volumes of data in order to extract hidden information [4]. Advanced machine learning and data mining algorithms are continuously used for this purpose, i.e., they are able to discover patterns, correlations and trends that occur in the collected data [5]. However, in the IoT environment such algorithms are often performed on devices with limited resources in terms of memory, computing power, energy and bandwidth [6]. For this reason, it is necessary to find a good compromise between the performance of the algorithms (e.g., accuracy) and the amount of resources needed for their execution.

In addition, due to the large scale, heterogeneity and complexity of IoT systems and networks, designing and testing IoT services are still open issues. Prototyping using a large number of hardware nodes could be very expensive and inflexible, and bench-marking and setting up real experiments could be very challenging [7]. For these reasons, the Modeling and Simulation approach (M&S) results to be a powerful and flexible tool for reproducing and testing IoT systems and networks. More in details, M&S allows to effectively analyze and evaluate different design alternatives by avoiding risks, costs and fails associated with extensive field experimentation. This opportunity becomes crucial, when complete and actual tests are too expensive to be performed in terms of cost, time, and computational resources [8], [9]. To improve the performance of large-scale simulations, specific approaches and models for collaborative simulation execution on hybrid systems, such as CPU+GPU, have been proposed [10], [11].

In this work, we propose a novel IoT architecture for efficiently extracting Regions-of-Interest (RoIs) in a large

scale urban computing environment, where a huge amount of geotagged data are generated and collected through users' mobile devices. There are two main contributions in the paper. Firstly, we defined a hybrid cloud/edge architecture capable of managing a large number of devices connected to it. In this way there is the possibility to configure how the workload will be distributed between the cloud and edge nodes, in order to optimize the performance of the application. Secondly, we evaluated through a simulation approach the different design choices for improving the performance of RoI mining algorithms in terms of processing time, network delay, task failure and computing resource utilization. To this end, two orchestration policies have been defined (*network-based* and *utilization-based*) for managing large amount of data to be processed.

Several experiments have been carried out to evaluate the performance of the proposed architecture starting from different configurations and orchestration policies. In particular, to test a high number of possible configurations, the Edge-CloudSim simulator [12] has been used. The simulation results showed that the proposed hybrid cloud/edge architecture with the use of two orchestration policies (network- and utilization-based) permits to improve resources utilization, also granting low network latency and task failure rate in comparison with other traditional scenarios (only-edge or only-cloud). Although the cloud-only scenario obtained a good performance in processing time, it produces a high percentage of failed tasks (e.g., 60% using 800 devices). Also, compared to the edge-only scenario, the proposed hybrid approach lead to a reduction in processing time and task failure rate up to 60% and 81%, respectively. Thus in general, the results obtained confirm that the proposed solution is effective for this type of problem even as the number of connected devices and the number of computing nodes increase. Compared to the state of the art, this turns out to be one of the first works that addresses the RoI mining problem in the IoT field, by proposing a hybrid cloud/edge architecture and evaluating its performance compared to standard solutions.

The structure of the paper is as follows. Section II provides some definitions and the problem statement. Section III discusses related work. Section IV describes the proposed hybrid cloud/edge architecture. Section V presents a case study and a performance evaluation by using the two orchestration policies introduces above. Finally, Section VI concludes the paper.

II. Problem statement

With the widespread use of mobile devices and location-based services, everyday people share information about the places they visit, often indicating the exact coordinates of such places. As an example, millions of users share on social media platforms posts about their activities, also including images, videos, and information about the places they visit. Such posts are often geotagged, which means they contains the spatial coordinates (latitude/longitude) of the place where they was created. Thus, geotagged data gathered from social media can be used to discover Places-of-Interest(PoIs) that have attracted many visitors.

A *Place-of-Interest* (*PoI*) is a location that someone finds useful or interesting, such as tourist attractions (e.g., squares or museums) or business locations (e.g., shopping malls). In this study, the terms *Place-of-Interest* and *Point-of-Interest* are considered similar, and thus are used interchangeably in the paper.

Since information on a PoI is generally limited to an address or GPS coordinates, it is hard to match users' trajectories with PoIs. For this reason, it is useful to define *Region-of-Interest* (*RoI*) representing the boundaries of the PoI's area [13]. RoIs can be defined as "spatial extents in geographical space where at least a certain number of user trajectories pass through" [14]. Thus, RoIs represent an useful abstraction for partitioning the space into meaningful areas and, correspondingly, to associate a label to a place. In literature, RoIs are also named as *regions of attraction* or *frequent (dense) regions* [15]. Therefore, the RoI detection process represents a key step for aggregating and comparing trajectories, which must be addressed before carrying out the trajectory extraction process. A *trajectory* is a sequence of spatial regions followed by a user, and consequently a *frequent trajectory* is a sequence of spatial regions that emerge as frequently visited by users [16]. Such frequent trajectories can be obtained through advanced techniques that are called trajectory data mining [17].

A geotagged item can be associated to a PoI $\mathcal{P}$ if its text or tags refer to $\mathcal{P}$. For example, geotagged items that have been created in the area of the Colosseum in Rome usually contain keywords such as "Colosseum", "Coliseum" or "Coliseo". By grouping all the items that refer to a PoI and applying a clustering algorithm, a suitable RoI for the Colosseum can be obtained. Most existing clustering techniques have high computational complexity and they have scalability issues [18]. To deal with these issues, many research efforts are focusing on the definition of distributed clustering approaches, where many computing nodes combine their effort to solve a large problem [19].

After extracting the RoIs, the user's trajectories are transformed from sequences of coordinates (usually coming from GPS sensors) into sequences of RoIs. Then, a trajectory mining algorithm can be used to discover frequent patterns in user movements. RoI and trajectory mining algorithms are widely used in different domains (e.g., urban planning, disaster prevention, trade area analysis, store rollout planning, transportation and tourism) for analyzing and discovering users' mobility patterns [20].

The next section reviews the existing literature contributions dealing with both the use of IoT-based architecture and the modern M&S techniques to evaluate system performance.

III. Related Work

With the increasing popularity of IoT technologies, several novel research projects have been carried out for obtaining valuable insight from large data generated by IoT devices. Big data analytics in IoT requires technologies and tools that

can transform large amounts of structured, unstructured, or semi-structured data into more understandable data ready to be analyzed. Advanced machine learning and data mining algorithms are used to discover patterns, trends, and correlations over a variety of time horizons in the data [4], [5]; however, these tasks are executed by devices characterized by limited resources, such as memory, processing, bandwidth and energy [6]. Thus, it is necessary to find the right compromises between performance (e.g., accuracy) and amount of resources required for computation. Below is a series of research papers that propose data mining/machine learning solutions in IoT area.

There are many surveys produced in this area. For example, [21] covered supervised and unsupervised learning algorithms in the ioT area, and outlined several applications including pattern recognition, anomaly detection, computer vision, speech processing.

[22] examined the applicability of eight data mining/-machine learning algorithms for IoT data. These include, among others, the deep learning artificial neural networks (DLANNs), which build a feed forward multi-layer artificial neural network (ANN) for modelling high-level data abstractions. The achieved results on real IoT datasets show that decision tree algorithms have better accuracy, are memory efficient and have relatively higher processing speeds, while ANNs and DLANNs can provide highly accurate results but are computationally expensive.

[23] highlights how the current security and privacy solutions of IoT platforms suffer from a series of problems related to the dynamic nature of networks. The paper highlights how some of the gaps in current IoT platform solutions can be overcome by using advanced machine learning algorithms. Machine learning techniques can be used to enable the IoT devices to adapt to their dynamic environment, by supporting self-organizing operation and also optimize the overall system performance by learning and processing statistical information from the environment.

[21] covered supervised and unsupervised learning algorithms in the IoT area, and outlined several applications including pattern recognition, anomaly detection, computer vision, speech processing.

For a correct evaluation of machine learning algorithms, realistic tests are needed on an IoT infrastructure consisting of a large number of entities/devices connected to each other and running different types of software. Designing and validating IoT infrastructures is still a complex issue due to the even-increasing complexity and technological improvement that makes them difficult to study and evaluate. M&S plays an essential role in managing this complexity since it allows to imitate the structure and behaviour of a complex system during its lifecycle [24]–[26].

[27] introduced main issues on the simulation of IoT infrastructure, and discussed a new combination of M&S techniques to enhance scalability and permit the real-time execution of massively populated IoT environments (e.g., large-scale smart cities).

[28] presented an overview of the challenges that arise when testing large IoT applications at the system level. To overcome these challenges, the author proposed a hybrid simulation-based testing technique that allows to evaluate IoT infrastructure by orchestrating a real-time interaction between real-life and virtual local IoT entities so as to detect emergent behaviors.

Concerning the tools and software, there are many simulators that have been proposed in recent years in the IoT field [29]. Among them, EdgeCloudSim [12] is a CloudSim extension [30] that provides a modular architecture for supporting a variety of crucial functionalities such as network modeling specific to WLAN and WAN, device mobility model, realistic and tunable load generator.

Although IoT technologies have been successfully used in many urban computing scenarios, its adoption in large-scale RoI and trajectory mining applications requires to address fundamental issues, such as those involving the definition of scalable architectures capable to effective deal with the high amount of data produced by users. Existing techniques for finding RoIs are based on three main approaches: pre-defined shapes, density-based clustering, grid-based aggregation and hybrid approaches [19]. According to predefined shapes approach, shapes like circles and rectangles are used to represent RoIs. For example, [31] used circular RoIs to extract popular touristic routes from Flickr. In density-based clustering approach, RoIs are obtained by clustering a set of geographical locations. For instance, [32] used DBSCAN to discover tourist attraction areas from social media posts. For Grid-based aggregation approach the area under analysis is discretized in a regular grid and extracts RoIs by aggregating the grid cells. For example, [14] divide an area into grid cells and count the trajectories passing through each cell. Grid cells whose counters are above a certain threshold are expanded to form rectangular shaped RoIs. Hybrid approaches combine some aspects of the approaches mentioned above. As an example, G-RoI [3] exploits the indications contained in social media items (e.g. tweets, posts, photos or videos with geospatial information) to discover the RoI of a PoI with a high accuracy.

IV. SYSTEM ARCHITECTURE

Although cloud computing provides high computing resources, even dynamically allocable, it should be noted that centralizing the collection of data, and their processing, in a single network node could lead to performance issues [33], [34]. To overcome these issues, the proposed cloud/edge architecture takes full advantage of the capabilities offered by edge and cloud computing to efficiently manage the volume of data produced in an urban area. Thanks to its horizontal scalability, the defined system architecture allows to manage a variable and growing number of connected devices (i.e., more devices, more resources). Moreover, it enables to perform distributed computation on edge devices, which is moved to cloud-only when needed. In such a way, it is possible to improve computation time, limit network congestion (user data

are processed locally to the edge device without having to transfer it to Cloud), and reduce the number of failed tasks.

As shown in Figure 1, the proposed architecture is composed of three layers.
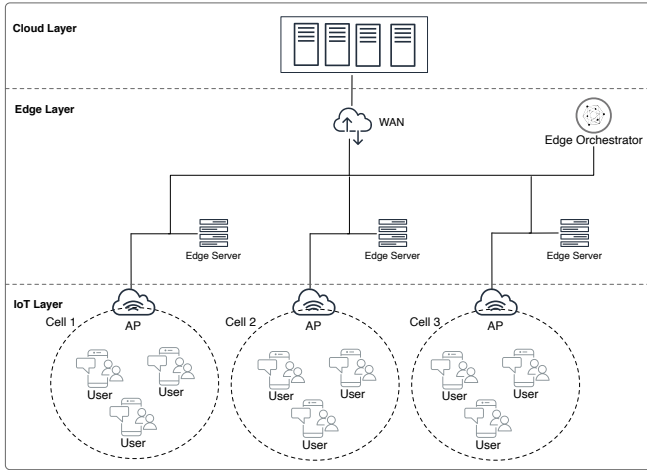

Fig. 1. The proposed IoT architecture.

The *IoT Layer* includes the mobile devices that are exploited by users to share contents during their movements. In particular, users visit PoIs that are located in a set of areas, which defines a partitioning of the whole urban area under analysis. The device collects user data together with additional information gathered by using the embedded components (e.g., the GPS coordinates and cell ID) and send information for processing and storage.

The *Edge Layer* represents the edge infrastructure that consists of different type of edge devices (e.g., Arduino and Raspberry Pi), which provide the infrastructure for collecting the raw data generated by users. The *Edge Layer* processes user data as long as computing resources are sufficient. When resources are no longer sufficient, tweets are forwarded to the cloud for further processing through the *Edge Orchestrator (EO)*. The EO component is the decision maker of the system, which uses the status information of the edge servers to decide how and where to handle incoming user requests and, if necessary, to offload them to other edge servers or Cloud. EO is a component that allows to simulate different policies and evaluate their effectiveness. EO can implement various policies, which take into account various performance metrics, such as the network congestion level, the status and load level of edge nodes and Cloud. In some cases, EO can also adopt machine learning algorithms or heuristics to solve the problem of optimally allocating data and tasks to the available computing resources (see, e.g., [35], [36]).

The *Cloud layer* represents a large set of computing resources that can be dynamically provisioned and released, where the tasks can be offloaded to be performed on behalf of the edge servers. From a client perspective, the cloud is an abstraction for remote, infinitely scalable provisioning of computation and storage resources, which have emerged

as effective computing platforms to face the challenge of processing Big Data repositories in limited time, as well as to provide effective and efficient data analysis environments to both researchers and companies(see, [37]).

Thanks to the presence of the EO, the proposed architecture is able to support different resource allocation policies, in order to efficiently manage RoI mining applications. Specifically, in support of this architecture, two new orchestration policies have been defined, *Network Based (EO-NB)* and *Utilization Based (EO-UB)*, capable of efficiently and effectively managing the geolocated data produced by users through the use of mobile devices. Such orchestration policies permit to improve the usage of computing resources (e.g., lower VM utilization), also reducing network latency and task failure rate.

V. EXPERIMENTAL EVALUATION

To evaluate the performance of the proposed architecture, a RoI mining application has been defined. In particular, it consists of two tasks:

- *Text Processing*, which consists in processing the user data in order to verify the presence of geolocation or textual information able to determine the spatial coordinates (latitude/longitude) of the place where the datum was created.
- *RoI Detection*, which consists in applying a clustering algorithm to extract a polygon representing the area of the Regions-of-Interest (RoIs). From a computational point of view, this task is very onerous, as the time complexity of the RoI detection algorithm may be $\mathcal{O}(n^2)$ or higher [3]. In many cases, using a parallel and distributed clustering approach leads to a significant reduction in computation time.

The two tasks are generated according to a Poisson distribution with a classic active/idle task generation pattern.

It is worth noting that the IoT layer is composed of users' mobile devices, which do not perform any computation, but operate as data sources for the generation of geotagged items containing information about the places visited by users. For these reason, in the rest of this section, distributing the computation on two layers, i.e. edge and cloud, is considered.

The architecture combines cloud and edge through two orchestration policies, i.e., *Network Based (EO-NB)* and *Utilization Based (EO-UB)*. The former exploits the network status to decide where incoming data must be processed; the latter policy plans executions based on the utilization of computing node (virtual machines - VMs). It should be noted that the proposed architecture is generic and does not place any constraints on the policy to be used. The EO can be easily extended to support more complex policies for allocating tasks to computing nodes, e.g., based on Artificial Intelligence (AI) algorithms [35]. Algorithm 1 shows the pseudo code of the orchestration procedure.

In the performed experiments, the IoT architecture defined in Figure 1 has been composed of one cloud server and 20 edge servers (one for each cell of the urban area). In addition, a variable number of mobile devices, ranging from

Algorithm 1 Edge Orchestrator
```
1:  Initializing of Simulation Manager – SimManager.
2:  Initializing EO and applications.
3:  procedure DEVICE TO OFFLOAD(task, τ₁, τ₂)
4:      deviceID ← null
5:      policy ← EdgeOrchestrator.getPolicy()
6:      if policy == NetworkBased then
7:          wanDelay ← SimManager.getCurrentWanDelay()
8:          if wanDelay ≥ τ₁ then
9:              deviceID ← CLOUD_ID
10:         else
11:             deviceID ← getAvailabeEdgeDeviceId()
12:         end if
13:     else                          ▷ Utilization Based policy
14:         utilization ← SimManager.getAvgEdgeUtilization()
15:         if utilization ≥ τ₂ then
16:             deviceID ← CLOUD_ID
17:         else
18:             deviceID ← getAvailabeEdgeDeviceId()
19:         end if
20:     end if
21:     return deviceID
22: end procedure
```

200 to 800, has been considered. Table I reports the main simulation parameters along with the values that were used for configuring the simulator.

Figure 2 reports the performance of the proposed IoT architecture as the number of mobile devices varies. In particular, Figures 2(a) and 2(b) report the average processing time and average percentage of failed tasks, respectively. A task can fail due to lack of VM resources or low network bandwidth. In particular, if the VM utilization is too high, new tasks may fail since they are not be accepted by any VM. Similarly, if too many clients connect to the same node, some tasks may be interrupted or fail due to network congestion. Concerning both processing time and percentage of failed tasks, the edge-only scenario obtained the worst results, since the performance degrades significantly as the number of devices increases. Although the cloud-only scenario obtained an average reduction of 76% in processing time, this result is mainly due to the high percentage of failed tasks.

Compared to the edge-only scenario, the two orchestration policies defined within the the proposed IoT architecture lead to a good reduction in processing time (38% for EO-UB to 60% for EO-NB), also minimizing the number of failed tasks (81% for EO-UB to 66% for EO-NB).

Figure 2(d) shows the average VM utilization for the different scenarios. Reducing VM utilization is a very important aspect to optimize large-scale applications that involve large computing resources. As an example, the capability of modern computing devices to adapt the CPU frequency (and therefore the energy consumption) to the workload allow to obtain significant benefits in terms of cost and energy consumption reduction. Furthermore, reducing the risk of saturating computational resources permits to handle any unexpected workload peaks that may occur.

As reported in Figure 2(d), EO-UB and EO-NB were able

to efficiently balance the load between the two layers of the IoT architecture (edge and cloud). As an example, considering 800 mobile devices, the orchestration policies permit to obtain a VM utilization that is on average 13% and 19% lower than edge- and cloud-only scenarios, respectively.

Figure 2(e) provides a detailed view of the performance comparison between the two orchestration policies on edge and cloud, respectively.

Figure 2(e) presents a detailed view of the distribution of the VM utilization between the edge and cloud layers, when using the orchestration policies. When a small number of devices are used (e.g., less than 600), the orchestration policies tend to favor the allocation of tasks on the Edge, up to the saturation of either computational or network resources. Beyond certain thresholds (see Algorithm 1, τ₁ and τ₂), the tasks are moved to the cloud for computation, so as to limit failure due to lack of resources. Such a conservative approach also aims at limiting the network delay (see Figure 2(b)), since data is transferred to the cloud-only if necessary. Table II summarized the numerical results obtained in our experiments.

In conclusion, the proposed architecture and orchestration policies obtained better results than standard approach based on Cloud, significantly improving processing times and percentage of failed tasks. Moreover, thanks to a more efficient resource utilization, they ensure a better management of any workload peaks generated by mobile devices in the urban area.

VI. CONCLUSIONS

In recent years the use of IoT infrastructures and solutions enable the processing of data closer to where it is generated, reducing the network traffic, but also task failures. Generally, IoT infrastructures are composed of many heterogeneous components that are characterized by independent behaviors and interact each other to pursuit common objectives. Understanding, studying and designing modern IoT infrastructures are going to be a great challenge in the next years. In this context, Modeling and Simulation (M&S) represents a powerful analytical method that allows to reproduce layers and behaviors of such infrastructures.

The paper presented an IoT architecture for efficiently extracting Regions-of-Interest (RoIs) in a large scale urban computing environment. To effectively manage the huge amount of data generated by users's mobile devices, two orchestration policies, i.e., Network Based (EO-NB) and Utilization Based (EO-UB), have been defined.

To assess the effectiveness of the defined architecture, a RoI mining application has been evaluated through simulation by using the *EdgeCloudSim* simulator. The simulation results showed that the architecture permits to improve resources utilization, also granting low network latency and task failure rate in comparison with other scenarios. As an example, compared to the edge-only scenario, the two orchestration policies lead to a reduction in processing time (from 38% to 60%) and task failure rate (from 66% to 81%).

Future research efforts will be devoted to define novel and more complex orchestration polices that exploit Artificial Intel-

TABLE I

EDGECLOUDSIM SIMULATION PARAMETERS

| Parameter	Description	Value
Simulation time	Duration of the simulation in seconds.	300s
Mobile devices	Number of mobile devices used in the simulation scenarios.	200-800
Edge servers	Number of edge servers.	20
Edge server processing speed	Computing processor's speed of edge servers in terms of Million Instructions Per Second.	2441 MIPS
cloud processing speed	Computing processor's speed of cloud in terms of Million Instructions Per Second.	89600 MIPS
Text Processing Task		
Poisson interarrival	Mean interarrival time between two tasks.	3
Active period	The active period of the task.	300s
Idle period	The idle period of the task.	10s
Upload data size	Mean input file sizes to upload.	20 KB
Download data size	Mean output file sizes to download.	2 KB
Task length	Mean number of instructions to execute the emerging task.	2000 MIPS
RoI Detection Task		
Poisson interarrival	Mean interarrival time between two tasks.	120
Active period	The active period of the task.	300s
Idle period	The idle period of the task.	10s
Upload data size	Mean input file sizes to upload.	2500KB
Download data size	Mean output file sizes to download.	200KB
Task length	Mean number of instructions to execute the task.	150000 MIPS

TABLE II

SIMULATION RESULTS USING DIFFERENT SCENARIOS AND ORCHESTRATION POLICIES

Metric	Number of Devices	Only Edge	Only Cloud	Two layers with EO Utilization Based	Two layers with EO Network Based
Average Processing Time (sec)	200	0.99	0.06	1.03	0.38
	400	1.32	0.42	0.75	0.51
	600	1.82	0.54	0.68	0.77
	800	1.91	0.55	0.77	0.91
Average Failed Tasks (%)	200	0.047	0.22	0.05	0.01
	400	32.54	23.27	0.04	0.57
	600	62.50	48.28	6.85	13.35
	800	73.52	60.19	27.03	32.24
Average Network Delay (ms)	200	2.93	15.46	3.13	8.65
	400	2.87	14.23	6.97	7.08
	600	3.03	14.28	10.00	6.34
	800	3.05	15.03	9.24	6.02
Average VM Utilization (%)				(Edge/Cloud)	(Edge/Cloud)
	200	49.17	16.41	48.23/0	16.84/6.86
	400	89.93	84.12	73.98/6.98	47.33/20.32
	600	94.61	91.11	76.27/66.95	82.55/51.76
	800	98.16	94.80	77.75/87.54	90.56/75.29

ligence (AI), Machine Learning (ML), and Fuzzy approaches for improving task allocation.

REFERENCES

[1] P. Escamilla-Ambrosio, A. Rodríguez-Mota, E. Aguirre-Anaya, R. Acosta-Bermejo, and M. Salinas-Rosales, "Distributing computing in the internet of things: cloud, fog and edge computing overview," in *NEO 2016: Results of the Numerical and Evolutionary Optimization Workshop NEO 2016 and the NEO Cities 2016 Workshop held on September 20-24, 2016 in Tlalnepantla, Mexico*, pp. 87–115, Springer, 2018.

[2] L. Belcastro, F. Marozzo, and D. Talia, "Programming models and systems for big data analysis," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 34, no. 6, pp. 632–652, 2019.

[3] L. Belcastro, F. Marozzo, D. Talia, and P. Trunfio, "G-roi: Automatic region-of-interest detection driven by geotagged social media data," *ACM Transactions on Knowledge Discovery from Data*, vol. 12, no. 3, 2018.

[4] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa, and I. Yaqoob, "Big iot data analytics: architecture, opportunities, and open research challenges," *ieee access*, vol. 5, pp. 5247–5261, 2017.

[5] E. Ahmed, I. Yaqoob, I. A. T. Hashem, I. Khan, A. I. A. Ahmed, M. Imran, and A. V. Vasilakos, "The role of big data analytics in internet of things," *Computer Networks*, vol. 129, pp. 459–471, 2017.

[6] S. Zahoor and R. N. Mir, "Resource management in pervasive internet of things: A survey," *Journal of King Saud University-Computer and Information Sciences*, 2018.

[7] M. Salama, Y. Elkhatib, and G. 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*, pp. 251–261, 2019.

[8] R. M. Fujimoto, A. W. Malik, A. Park, *et al.*, "Parallel and distributed simulation in the cloud," *SCS M&S Magazine*, vol. 3, pp. 1–10, 2010.

[9] R. M. Fujimoto, *Parallel and distributed simulation systems*, vol. 300. Citeseer, 2000.

[10] K. Al-Zoubi and G. Wainer, "Modelling fog & cloud collaboration methods on large scale," in *2020 Winter Simulation Conference (WSC)*, pp. 2161–2172, IEEE, 2020.

[11] W. Tang, W. Cai, Y. Yao, X. Song, and F. Zhu, "An alternative approach for collaborative simulation execution on a cpu+ gpu hybrid system," *Simulation*, vol. 96, no. 3, pp. 347–361, 2020.

[12] C. Sonmez, A. Ozgovde, and C. Ersoy, "Edgecloudsim: An environment

(a) Processing time.

(b) Network delay.

(c) Failed tasks.
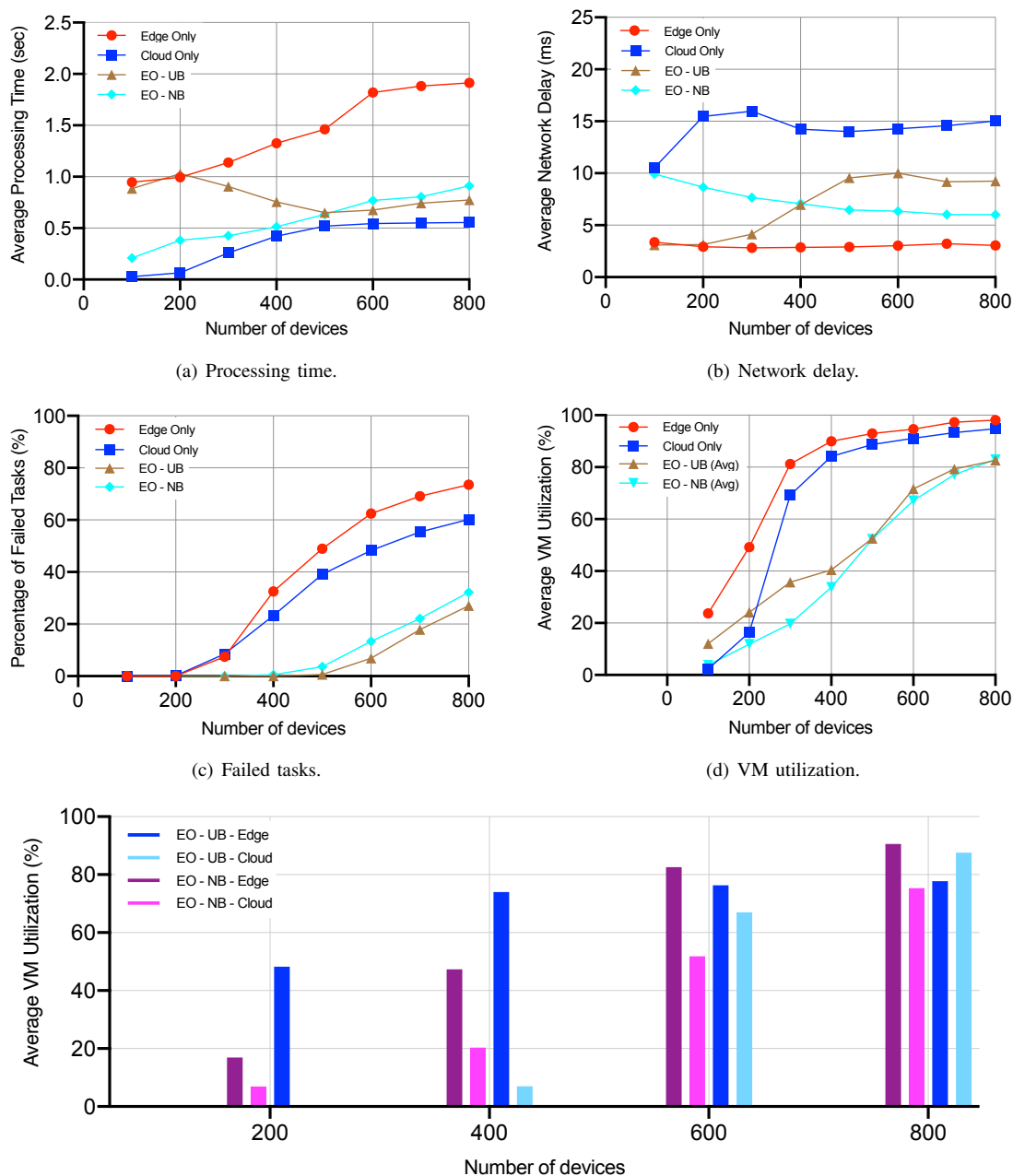
(d) VM utilization.

(e) Comparison on VM utilization between the two orchestration policies: EO-UB and EO-NB on edge and Cloud, respectively.

Fig. 2. Performance results of the RoI application obtained for the different simulation scenarios.

for performance evaluation of edge computing systems," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 11, p. e3493, 2018.

[13] V. De Graaff, R. A. De By, M. Van Keulen, and J. Flokstra, "Point of interest to region of interest conversion," in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'13, (New York, NY, USA), pp. 388–391, ACM, 2013.

[14] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 330–339, 2007.

[15] A. Altomare, E. Cesario, C. Comito, F. Marozzo, and D. Talia, "Trajectory pattern mining for urban computing in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 2, pp. 586–599,

2017.

[16] J. A. Long and T. A. Nelson, "A review of quantitative methods for movement data," *International Journal of Geographical Information Science*, vol. 27, no. 2, pp. 292–318, 2013.

[17] J. D. Mazimpaka and S. Timpf, "Trajectory data mining: A review of methods and applications," *Journal of Spatial Information Science*, vol. 2016, no. 13, pp. 61–99, 2016.

[18] F. Farnstrom, J. Lewis, and C. Elkan, "Scalability for clustering algorithms revisited," *ACM SIGKDD Explorations Newsletter*, vol. 2, no. 1, pp. 51–57, 2000.

[19] L. Belcastro, M. T. Kechadi, F. Marozzo, L. Pastore, D. Talia, and P. Trunfio, "Parallel extraction of regions-of-interest from social media data," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 8, p. e5638, 2021.

[20] J. Gudmundsson, P. Laube, and T. Wolle, "Computational movement analysis," in *Springer handbook of geographic information*, pp. 423–438, Springer, 2011.

[21] U. S. Shanthamallu, A. Spanias, C. Tepedelenlioglu, and M. Stanley, "A brief survey of machine learning methods and their sensor and iot applications," in *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pp. 1–8, IEEE, 2017.

[22] F. Alam, R. Mehmood, I. Katib, and A. Albeshri, "Analysis of eight data mining algorithms for smarter internet of things (iot)," *Procedia Computer Science*, vol. 98, pp. 437–442, 2016.

[23] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in iot security: Current solutions and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1686–1721, 2020.

[24] A. Falcone and A. Garro, "Distributed co-simulation of complex engineered systems by combining the high level architecture and functional mock-up interface," *Simulation Modelling Practice and Theory*, vol. 97, no. August, p. 101967, 2019.

[25] A. Falcone and A. Garro, "Pitfalls and remedies in modeling and simulation of cyber physical systems," in *24th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications, DS-RT 2020, Prague, Czech Republic, September 14-16, 2020*, pp. 1–5, Institute of Electrical and Electronics Engineers Inc., 2020.

[26] D. Bouskela, A. Falcone, A. Garro, A. Jardin, M. Otter, N. Thuy, and A. Tundis, "Formal requirements modeling for cyber-physical systems engineering: an integrated solution based on form-l and modelica," *Requirements Engineering*, Aug 2021.

[27] G. D'Angelo, S. Ferretti, and V. Ghini, "Simulation of the internet of things," in *2016 International Conference on High Performance Computing & Simulation (HPCS)*, pp. 1–8, IEEE, 2016.

[28] S. Bosmans, S. Mercelis, J. Denil, and P. Hellinckx, "Testing iot systems using a hybrid simulation based testing approach," *Computing*, vol. 101, no. 7, pp. 857–872, 2019.

[29] A. Markus and A. Kertesz, "A survey and taxonomy of simulation environments modelling fog computing," *Simulation Modelling Practice and Theory*, vol. 101, p. 102042, 2020.

[30] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and experience*, vol. 41, no. 1, pp. 23–50, 2011.

[31] E. Spyrou and P. Mylonas, "Analyzing flickr metadata to extract location-based information and semantically organize its photo content," *Neurocomputing*, vol. 172, pp. 114–133, 2016.

[32] Y.-T. Zheng, Z.-J. Zha, and T.-S. Chua, "Mining travel patterns from geotagged photos," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 3, pp. 1–18, 2012.

[33] N. Khanghahi and R. Ravanmehr, "Cloud computing performance evaluation: issues and challenges," *Comput*, vol. 5, no. 1, pp. 29–41, 2013.

[34] V. K. Reddy, B. T. Rao, and L. Reddy, "Research issues in cloud computing," *Global Journal of Computer Science and Technology*, 2011.

[35] C. Sonmez, C. Tunca, A. Ozgovde, and C. Ersoy, "Machine learning-based workload orchestrator for vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[36] C. Sonmez, A. Ozgovde, and C. Ersoy, "Fuzzy workload orchestration for edge computing," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 769–782, 2019.

[37] L. Belcastro, F. Marozzo, D. Talia, and P. Trunfio, "Big data analysis on clouds," in *Handbook of Big Data Technologies* (A. Zomaya and S. Sakr, eds.), pp. 101–142, Springer, December 2017.