2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech) | 978-1-6654-6297-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/DASC/PiCom/CBDCom/Cy55231.2022.9927883

From Cloud-Edge to Edge-Edge Continuum: the Swarm-Based Edge Computing Systems

Lorenzo Carnevale[1], Alessandro Ortis[2], Giancarlo Fortino[3], Sebastiano Battiato[2], Massimo Villari[1]
[1]*Department of Mathematics and Computer Science, Physics and Hearth Sciences*, University of Messina, Messina, Italy
{lcarnevale,mvillari}@unime.it
[2]*Department of Mathematics and Computer Science*, University of Catania, Catania, Italy
{ortis,battiato}@dmi.unict.it
[3]*Department of Informatics, Modeling, Electronics, and Systems*, University of Calabria, Rende, Italy
{giancarlo.fortino}@unical.it

***Abstract*—Modern cloud-edge-device computational platforms does not match the needs of artificial intelligence at the edge of the network. Indeed, the lack of computing power allows that only some AI processes can be performed on edge devices, having also to consider their constrained energy capacity. Moreover, the lack of computing continuum between nodes of the same layer, i.e., edge-to-edge, allows to only operate independently within the layer by sensing the environment where nodes stay. In this paper, we propose a lightweight framework for collaborative nodes with decentralized edge intelligence. Organized like a swarm, the groups of nodes emphasize the edge-to-edge continuum of the device-edge-cloud paradigm. This supports a paradigm shift from programming environments for individual devices to dynamic and cooperating groups of nodes. The nodes' coordination relays on green overlay and offloading mechanisms. Innovative mesh architectures with mixed topologies allow building overlays for having swarm coordination. Tasks offloading exploits the overlays to balance the swarm in near-real-time, according to forecasted energy consumption. Stemming from the proposed reference architecture, we also discuss a series of open challenges, which we believe represent relevant research directions in the nearest future.**

***Index Terms*—Cloud-Edge Continuum, Swarm Computing, Distributed Intelligence, Offloading, Overlay Network, Energy-Aware**

I. Introduction

Gartner's hype cycle[1] for artificial intelligence (AI) 2021 places the edge AI at the peak of inflated expectation, leaving the innovation trigger phase in only 12 months. Moreover, the IBM Institute for Business Value claims that the expected return on investment in green edge computing amounts to 10% in 2022. According to Gartner, however, there will still be room for further investments in edge AI, because it will steadily reach the Plateau of Productivity within a maximum of 5 years. In fact, when talking about the computation of AI tasks at the edge of the network, the literature shows solutions that rely on well-known infrastructures that involve devices, edge, and cloud systems.

In this regards, tasks are often offloaded from devices to edge to cloud, due to constrained computational and energy resources, following the computing continuum paradigm. Considering there are about 13 billion connected devices in 2022 (Statista source[2]) and the amount of data they generate is expected to reach 73.1 ZB (zettabytes) by 2025 (International Data Corporation source), the risk of congestion in the network is not far away. On the other hand, having to cross the Public Internet to complete the computation, device-edge-cloud continuum paradigm is not suitable for near real-time applications. Furthermore, the use of energy-intensive computational resources, such as the cloud, affects the climate impact of the adopted solutions [1]. In summary, the literature clearly shows the need for further study since the solution to these problems cannot be adequately gleaned from the existing approaches.

In this paper, we propose a SwARm-based eDge computINg systEm, hereinafter referred to as "SARDINE", to investigate a different computing solution at the edge of the network for executing AI tasks. We will examine how edge nodes collaborate when organized like a swarm, a large number of small nodes where each individual performs a simple task, but whose action produces a complex behavior as a whole. Relying on many servers at the edge of network, we will determine a paradigm shift from programming environments for an individual to dynamic and cooperating groups of nodes. The device-edge-cloud paradigm will be then simplified. Offloading will not need to go through the Public Internet to be completed, which means minor latency and high distributed edge computation. Therefore, this turns into a potential near real-time solution. On the other hand, SARDINE will determine the carbon footprint of many energy-constrained computational resources at the edge of network, in opposition to a small number of nodes with high-performance resources.

Summarizing, we highlight and address the following major challenges: (i) the lack of computing power allows that only some AI processes can be performed on an edge device capable of withstanding inference and small learning tasks,

[1]The 4 Trends That Prevail on the Gartner Hype Cycle for AI, 2021, https://www.gartner.com/en/articles/the-4-trends-that-prevail-on-the-gartner-hype-cycle-for-ai-2021-

[2]Internet of Things (IoT) and non-IoT active device connections worldwide from 2010 to 2025, https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/

and then having to rely on the Cloud for those of high computational complexity; (ii) the constrained energy capacity of edge devices limits the available computation time; and (iii) the lack of computing continuum between nodes of the same layer, i.e., edge-to-edge, allows to only operate independently within the layer by sensing the environment where nodes stay.

The remaining of paper is organized as follow. A study of literature is reported in the Section II, highlighting recent solutions about Cloud-Edge continuum, offloading of distributed systems, and methodologies of nodes coordination with a focus on energy-efficient techniques. The Section III proposed SARDINE methodology and both the its system development and applications point of views. Challenges and research trends are instead described in Section IV. Finally, Section V reports conclusion and light to the future activities.

II. Related Works

Distributed systems use multiple processors to serve multiple applications and users. Data processing jobs are then distributed among the processors accordingly to which one can perform each job most efficiently. In this regard, distributed systems have quickly evolved driven by the numerous Internet of Things (IoT) devices. Indeed, according to Juniper Research, the number of IoT devices in 2021 was 46 billion, with an increase of 200% when compared to 2016. The consolidated capacity of sensing the environment has given new vigor to the need for managing and interpreting data, as well as the application of AI tools for future predictions. To avoid network congestion due to the massive transfer of data across the public Internet [2], researchers have initiated studies [3] to start computing at the network edge. The need for a new computational paradigm suddenly became urgent because the traditional cloud-based model was running into scalability challenges. Therefore, this gap was bridged by the research on compute continuum [4], a recent technological evolution for device-edge-cloud computing management. Numerous architectures have been then presented in the literature [5], [6], [7], but all of them constitute of a multi-tiered infrastructure. While IoT devices are concentrated at the edge of the network, edge devices are distributed between the network edge itself and the network core. Cloud is instead further away from IoT devices, and the upcoming requests must traverse the public Internet.

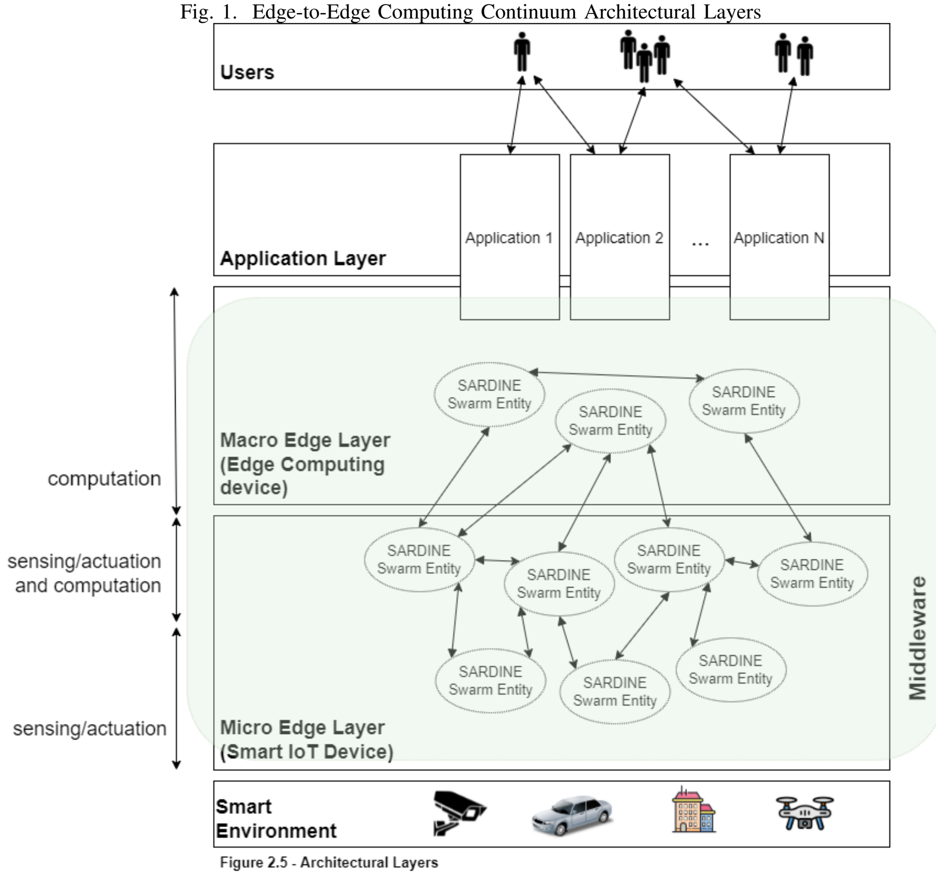
Although edge nodes can perform computation (i.e., machine learning), limited computational and energy resources require the distribution of tasks across the continuum. Such designs provide possibilities to map a variety of computing tasks along the device-edge-cloud tiers to achieve different levels of intelligence at different costs and energy budgets. Containerization can also be used in increasing flexibility by allowing live migration of containers either horizontally at the edge levels or vertically between the edge and cloud levels [8]. Each task is loaded in one corresponding container, which shares the physical resources with other containers in the same node. Therefore, a container migration manager not only monitors the resource requirement, latency, and power consumption of nodes but also determines the migration strategy [9]. Recent progress in offloading middleware applies transparent partition into smaller units [10], [11], such as threads and methods, and then smoothly offload to devices based on the analysis of execution time and energy consumption. However, the highlighted offloading approaches point out the lack of collaboration among nodes. Indeed, a task is typically assigned to one single node, and it is never shared between more of them, resulting in independent computational units. This increases the need to offload towards upper tiers. Data in device-edge-cloud infrastructures must then typically go through one or more tiers, connected by a flexible and adaptive network. The decision on how connecting nodes depends on the specific technological scenario [12]. A scenario where nodes are used to process sensor data will typically take advantage of wireless connections, while nodes employed in manufacturing processes will likely use wired connections. When talking about AI, solutions [13] consider the problem of learning model parameters from data distributed across multiple edge nodes, without sending raw data to a centralized place. Model parameters obtained at different edge nodes are then sent to an aggregator, which is a logical component that can run on the remote cloud. In terms of nodes' activities coordination, peer-to-peer networks (i.e., mesh networks) stand as an essential enabler for collaborative nodes for exchanging acquired status information on the computing environment and making decisions that are critical for the correct functioning of the entire system [14].

The problem related to the energy-constrained edge and IoT devices is another key factor in developing a collaborative computing infrastructure. Researchers focus on energy-aware computation schemes [15] in which computation offloading and resource allocation are optimized to make a tradeoff between energy consumption and latency considering the limited battery lifetime and latency-sensitive tasks. The residual energy of edge devices is introduced into the definition of a weighting factor, used in an iterative search algorithm to obtain an optimal offloading decision and resource allocation, which optimizes local computing frequency scheduling, channel allocation, power allocation, and computation offloading in a distribution method. The problem becomes even more evident by analyzing architectures for mobile nodes, such as the ones deployed in smart environments. The problem of managing constrained devices has been typically addressed by using reinforcement learning [3], [16].

III. Methodology

Based on the state of the art, we clearly pointed out poor initiatives about edge-to-edge continuum. The SARDINE approach will be based on the creation of a collaborative edge infrastructure, exploiting the properties of a swarm. The main object is the SARDINE Swarm Entity (SARDINE-SE), which is a lightweight agent that can perceive its environment, act on it, and share data and computation tasks with other SARDINE-SEs. They do not know each other, but they cooperate and communicate indirectly. SARDINE-SEs have inherent

Fig. 1. Edge-to-Edge Computing Continuum Architectural Layers



hardware and software constraints, e.g., low processing and transmission power, memory, and battery life. SARDINE-SEs follow a swarm organization and will be then distributed over the device-edge continuum according to the architectural layers reported in Figure 1. The Micro Edge Layer consists of smart Internet of Things (IoT) devices (i.e., camera, microphones, etc. represented by a SARDINE-SE) connected through diverse communication technologies and equipped with poor computational capabilities. The SARDINE-SEs can either sense and act the environment or perform computation, according to the specific application requirements. The Macro Edge Layer includes the edge devices (i.e., Raspberry Pi, Jetson Nano, etc. represented by a SARDINE-SE) connected through diverse communication technologies and equipped with medium computational capabilities. The SARDINE-SEs are herein only dedicated to computation, exploiting the whole performance from their constrained resources. This layer is also responsible for running applications based on the middleware capabilities. The use case driven application features are instead running in the Application Layer, which exploits the potentiality provided by the SARDINE approach regarding end-users.

The logical architecture of the SARDINE-SE is instead depicted in Figure 2. This is a cognitive architecture that attempts to model not only behavior but also structural properties of the modeled system. It defines artificial computational processes that act like certain cognitive systems (i.e., individuals of a swarm). The architecture enables to realize various cognitive abilities and coordination mechanisms. Furthermore, it is strongly decentralized (distributed), promoting parallel distributed computing, connectionism, and computing partition (i.e., divide et impera). Being inspired by biological systems, SARDINE applies a bottom-up model where the overall behavior (i.e., stigmergy) emerges from the interaction of simple nodes.

A. *From the Viewpoint of System Development*

SARDINE is organized into seven components (i.e., software managers).

Node Manager collects all capabilities to handle a node that takes part in the swarm, offering a unified abstract interface to manage all nodes similarly despite their specific heterogeneous characteristics. This Manager allows the configuration of the resource in the node infrastructure, as well as the registration of the associated sensing capabilities to enable context-awareness and data accessible to the Storage Manager. Herein are collected information associated with the status of resources and sensors. According to different profiles, it collects data about its dynamic status, hardware, and software characteristics, such as the utilization rate, battery life, physical position, tasks

Fig. 2. SARDINE Swarm Entity Logical Architecture

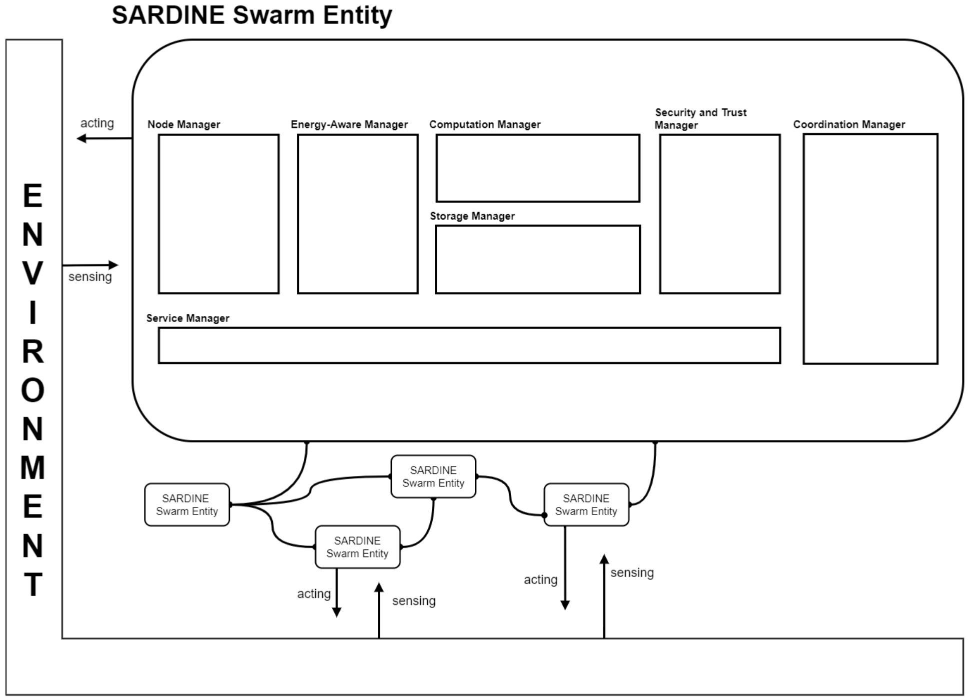


Figure 2.6 - SARDINE Logical Architecture

scheduling, available memory, network load, algorithms performance, and inference accuracy.

Storage Manager controls all processes associated with the collection and sharing of data coming from resources and sensors. It supports distributed indexing and data synchronization among individuals of the swarm, using both in-memory (i.e., cache) and persistent storage.

Computation Manager is responsible for managing the life-cycle of the tasks executed by the node resources. It supports the offloading according to node status profiled by the Node Manager, interacting with the Coordination Manager to enable the migration of tasks from one node to its neighbor. The execution status is reported to Service Manager.

Security and Trust Manager involves diverse levels of security requirements and resource self-protection mechanisms. It supports the encryption of data for both storage and computing, to preserve privacy according to the GDPR. Being SARDINE strongly decentralized, this Manager allows mechanisms of consensus for both nodes and data trust.

Energy-Aware Manager is responsible for applying policies that make the swarm's carbon footprint as lower as possible, according to the capabilities of nodes. It supports the execution of decision-makers to lifetime longer the battery-powered node. This Manager interacts with the Computation Manager for alerting when a node is running out of battery and then scheduling the task offloading.

Coordination Manager is responsible for connecting heterogeneous nodes that take part in a swarm. It supports the creation of overlays for both overall messages exchange and serving specific application goals. This Manager allows selecting the optimal available node of the swarm for executing the computation, according to the established nodes' collaboration and typologies of tasks each node can fulfill.

Service Manager is responsible for controlling all services (i.e., applications) running in a certain node and allocating these services over the most suitable node of the swarm. This Manager interacts with the Coordination Manager for discovering the most suitable available node. It also supports the quality-of-service management of running applications.

B. *From the Viewpoint of Applications*

SARDINE will find applications in several domains, among which industry 4.0, smart environment, mobility, healthcare, and cultural heritage. The applications will communicate with the infrastructure using a proper API, which will mask the underlying decentralized execution environment, as well as the coordination and offloading mechanisms. From the application point of view, a set of high-level services are available. Possible examples are vehicle tracking, license plate recognition, people counting, audio classification, etc. The application will be unaware of the involved sensory system, as well as of the distributed and dynamic computation, which is determined on-demand, depending on the whole swarm situation and the other pending tasks. In this way, applications will be developed based only on the inference target (e.g., monitoring an urban area, traffic monitoring, etc.) and the SARDINE's available capabilities (e.g., audio analysis, traffic anomaly detection). Considering this, the SARDINE infrastructure will allow the development of general applications.. Indeed, any application that exploits AI-based inferences performed dynamically on the sensory data flow coming from the smart environment layer can be designed and easily developed.

IV. Open Challenges and Research Directions

The implementation of the SARDINE vision into a realistic setting raises a number of challenges, which in turn open several research directions. Without any claim of completion, in the next sections we present those directions we argue are the most relevant in the current landscape of research.

A. Development of the model

A novel approach is needed to formalize swarms of smart devices at the edge of the IoT. This is a real challenge as currently available swarm models were conceived at a multi-agent systems level to address conventional distributed systems requirements, not considering at all the IoT device-edge continuum. We propose to begin from the ACOSO (Agent-oriented COoperating Smart Objects) model [17] as a basis to create the first swarm edge model, including computation and coordination, purposely conceived for addressing the requirements of the IoT device-edge continuum. The model will consider two coordinated layers: micro-edge and macro-edge. At the micro-edge, the smart devices are programmable sensing and/or actuating computing elements that coordinate with each other using a (logical and/or physical) stigmergic coordination model. At the macro-edge, the swarm can be seen as a "living organism" with macro goals dictated by a macro-programming language. This will notably allow to program the swarm at the macro level and then automatically translate the macro goals into programming directives embedded into the elements of the swarm.

B. Development of the middleware

Developing novel mechanisms of coordination (i.e., overlay) and computation (i.e., offloading) over the device-to-edge and edge-to-edge continuum is a key challenge for building highly distributed lightweight systems with a view on energy consumption. We aim to extend existing middleware to being green by design and supporting the offloading of edge nodes over other edge nodes, rather than the cloud. In this regard, we will use artificial intelligence algorithms (i.e., reinforcement learning) for reacting to data sensed from the environment, and according to residual resources (CPUs, memory, etc.) and energy (battery life, harvesting). As a coordination mechanism, we will build overlays through mesh architectures with mixed topologies. Nodes could then be grouped in one or multiple overlays for responding to fleet management instructions or serving the application layers.

C. Seamless interoperability of devices-to-edge and edge-to-edge levels

Harmonizing the coordination and computation of heterogeneous device (i.e., smart IoT objects) and edge (i.e., microprocessors) systems in a decentralized manner is a key challenge to provide seamless interoperability among the nodes. We propose a swarm-based infrastructure to smoothly provide decentralized, self-organized, and robust systems with consideration of coordination. We want to exploit the collective behavior of systems composed of many nodes who interact locally with each other and with their environment using decentralized and self-organized control to achieve complex tasks. A common interface (i.e., APIs) will be implemented with the SARDINE-SE, through which nodes will share an interoperability language.

D. Security, Privacy, and Trust

Ensuring security, privacy, and trust management of data, coordination, and computation in SARDINE are key challenges to enable the widespread diffusion of SARDINE services and applications. We propose to store data by using homomorphic encryption to preserve data integrity and security of the citizen. The encryption will happen as soon as data is generated on the edge of the network and stored in local databases. The use of homomorphic encryption eliminates the need to decrypt data before using it. Data integrity and privacy are therefore preserved even during data computation. On the other hand, we propose to use consensus algorithms to trust nodes during coordination. By considering there will be a limited number of nodes deployed at the edge, the system must ensure that all can agree on a single source of truth, even if some nodes fail. This makes the system highly reliable.

E. Scalability

Limiting the computation over the edge nodes could increase the number of them, due to the need to have several systems with limited capabilities. We want to exploit the potential benefits of swarm-based infrastructure, because when modeling edge systems as a swarm, the control mechanisms do not depend on the number of devices within the system, but only on the neighbors of the target node. Indeed, the overhead of coordination does not increase when the size of the group increases, as well as the performance of the whole system does not degrade.

F. Energy-awareness

Being based both on stationary and mobile devices, SARDINE should be energy-aware from sensing, actuation, communication, and computing viewpoints to prolong the overall system lifetime according to application-specific requirements. We aim to build a swarm infrastructure that is energy-aware by design from all aspects of edge computing, including architecture, operating system, middleware, service provisioning, and computing offloading. Nodes will monitor themselves in terms of residual energy (i.e., battery life, harvesting) and deliver that status to neighbors. Such information will be then involved in the election of the node towards which offloading the computation.

G. Development of use case driven middleware features

The proposed infrastructure is expected to manage high numbers of instances of smart applications with significant data requirements, offered to a large range of users, and with limited delays. For these reasons, one of the main challenges is the development of proper task offloading systems. We aim to define middleware features properly designed for the specific use cases. Efforts will be devoted to the customization of the swarm middleware implementation toward the support of the specific use-case scenarios. Considering the possible needs of the applications in terms of computation distribution and time-aware constraints.

H. Development of multi-objective application

Typically, abnormal events occur infrequently compared to normal activities. Therefore, to alleviate the waste of work and time, the development of AI algorithms for the automatic detection of anomalies is an important need. This is a challenging task, as AI-based models for detection are usually based on the exploitation of large-scale and possibly balanced labeled data, with a set of pre-defined possible outputs (i.e., classes). Real-world anomalous events are complicated and diverse. It is difficult to list all possible anomalous events. Therefore, it is desirable that the anomaly detection algorithm is not based on a precise definition of events categorized into classes. In other words, anomaly detection should be done with minimal supervision. The goal of a practical anomaly detection system is to promptly report an activity that deviates from normal patterns and identify the time window of the anomaly that occurs. Therefore, anomaly detection can be thought of as rough-level continuous pattern recognition for signal understanding, filtering anomalies from normal models. Once an anomaly has been detected, it can be further classified into one of the specific activities using classification techniques.

V. Conclusions

This position paper presents the architecture of SARDINE, a swarm-based edge computing system that integrates a set of services and methodologies aimed at shifting the cloud-edge computing continuum paradigm to the edge-edge one for providing artificial intelligence at the edge of the network. The SARDINE design considers the deployment of two main mechanisms: edge nodes coordination and computation offloading. The reference architecture is used as a starting point to discuss several relevant open research directions that would shape the work toward the realisation of the SARDINE vision in the future.

Acknowledgment

This work has been partially supported by the National Operational Programme (PON) Research and Innovation through the project TALISMAN (Codice ARS01 01116 - CUP J66C18000360005) and by the National Operational Programme Metropolitan Cities 2014-2020(PON METRO) through the projects Agorà (CUP F41I17000170006), Cloud@ME (CUP F49E17000040006), ImpleME (CUP F41I17000180006), MeSm@rt (CUP F41I18000230006) and Urbamid +(CUP F41H18000030006).

References

[1] E. Ahvar, A. C. Orgerie, and A. Lebre, "Estimating Energy Consumption of Cloud, Fog and Edge Computing Infrastructures," *IEEE Transactions on Sustainable Computing*, 2020.

[2] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A Survey on the Edge Computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, nov 2017.

[3] F. Samie, L. Bauer, and J. Henkel, "From cloud down to things: An overview of machine learning in internet of things," *IEEE Internet of Things Journal*, vol. 6, pp. 4921–4934, jun 2019.

[4] D. Balouek-Thomert, E. G. Renart, A. R. Zamani, A. Simonet, and M. Parashar, "Towards a computing continuum: Enabling edge-to-cloud integration for data-driven workflows,"

[5] C. Mwase, Y. Jin, T. Westerlund, H. Tenhunen, and Z. Zou, "Communication-efficient distributed AI strategies for the IoT edge," *Future Generation Computer Systems*, vol. 131, pp. 292–308, jun 2022.

[6] G. L. Santos, P. Takako Endo, M. F. Ferreira da Silva Lisboa Tigre, L. G. Ferreira da Silva, D. Sadok, J. Kelner, and T. Lynn, "Analyzing the availability and performance of an e-health system integrated with edge, fog and cloud infrastructures," *Journal of Cloud Computing*, vol. 7, dec 2018.

[7] B. Omoniwa, R. Hussain, M. A. Javed, S. H. Bouk, and S. A. Malik, "Fog/edge computing-based IoT (FECIoT): Architecture, applications, and research issues," *IEEE Internet of Things Journal*, vol. 6, pp. 4118–4149, jun 2019.

[8] Z. Zou, Y. Jin, P. Nevalainen, Y. Huan, J. Heikkonen, and T. Westerlund, "Edge and Fog Computing Enabled AI for IoT-An Overview," *Proceedings 2019 IEEE International Conference on Artificial Intelligence Circuits and Systems, AICAS 2019*, pp. 51–56, mar 2019.

[9] Z. Tang, X. Zhou, F. Zhang, W. Jia, and W. Zhao, "Migration Modeling and Learning Algorithms for Containers in Fog Computing," *IEEE Transactions on Services Computing*, vol. 12, pp. 712–725, sep 2019.

[10] K. Fu, W. Zhang, Q. Chen, D. Zeng, and M. Guo, "Adaptive Resource Efficient Microservice Deployment in Cloud-Edge Continuum," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, pp. 1825–1840, aug 2022.

[11] R. Ezhilarasie, M. S. Reddy, and A. Umamakeswari, "A new hybrid adaptive GA-PSO computation offloading algorithm for IoT and CPS context application," *Journal of Intelligent Fuzzy Systems*, vol. 36, pp. 4105–4113, jan 2019.

[12] L. Bittencourt, R. Immich, R. Sakellariou, N. Fonseca, E. Madeira, M. Curado, L. Villas, L. DaSilva, C. Lee, and O. Rana, "The internet of things, fog and cloud continuum: Integration and challenges," *Internet of Things*, vol. 3-4, pp. 134–155, oct 2018.

[13] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, pp. 1205–1221, jun 2019.

[14] A. J. Ferrer, S. Becker, F. Schmidt, L. Thamsen, and O. Kao, "Towards a cognitive compute continuum: An architecture for ad-hoc self-managed swarms," tech. rep.

[15] J. Zhang, X. Hu, Z. Ning, E. C. Ngai, L. Zhou, J. Wei, J. Cheng, and B. Hu, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet of Things Journal*, vol. 5, pp. 2633–2645, aug 2018.

[16] M. G. R. Alam, M. M. Hassan, M. Z. Uddin, A. Almogren, and G. Fortino, "Autonomic computation offloading in mobile edge for iot applications," *Future Generation Computer Systems*, vol. 90, pp. 149–157, jan 2019.

[17] G. Fortino, W. Russo, C. Savaglio, W. Shen, and M. Zhou, "Agent-oriented cooperative smart objects: From iot system design to implementation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, pp. 1949–1956, nov 2018.