Securing clustered edge intelligence with blockchain

Chinmaya Kumar Dehury, *Member, IEEE*, Satish Narayana Srirama, *Senior Member, IEEE*, Praveen Kumar Donta, *Member, IEEE*, and Schahram Dustdar, *Fellow, IEEE*

***Abstract*—Clustering the intelligence at the edge, irrespective of devices' type, location, and other attributes, is referred to as Clustered Edge Intelligence (CEI). CEI allows the devices to share their knowledge and events with other devices and the remote fog or cloud servers. The recent advancements facilitate the traceability of the events' history through analyzing the event logs, which are compute-intensive and easy to alter. This article focuses on a blockchain-based solution for CEI that makes the edge devices' events history immutable and easily traceable. Such a secure CEI mechanism can be applied in establishing a transparent and efficient smart city, supply chain, logistics, and transportation systems.**

***Index Terms*—Edge Intelligence, IoT, clustered edge intelligence, blockchain, security, smart environment**

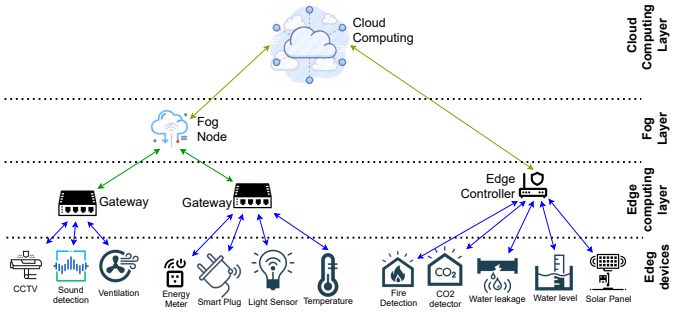


Fig. 1: Hierarchical representation of edge, fog and cloud computing environments.

I. Introduction

Internet of Things (IoT) has made it possible to further improve the quality and comfort of human being. This leverages the capability of surrounding tangible and intangible physical entities that can be controlled and managed remotely. According to an estimation published by Techradar[1], it is expected that over 125 billion physical objects will be connected to the Internet, thanks to IoT. With such a massive number of connected devices, the global IoT market is expected to reach a value of USD 1,386.06 billion by 2026 from USD 761.4 billion in 2020 at a CAGR (Compound Annual Growth Rate) of 10.53%, during the forecast period (2021-2026)[2].

The devices at the edge of the network, also referred to as Edge devices, are not only collecting data or sensing the surrounding environment, but also preprocessing the data and learning the basic meaning of that data. This is done by implementing an essential intelligence atop the limited computing resources available with the edge devices, also known as edge intelligence [1]. In such a scenario, the edge devices are primarily managed by the central control center that usually resides in a cloud computing environment and works in a master-worker approach. Further to improve the Quality of Service (QoS) through several supplements such as lesser latency and response time, fog computing environment is introduced between cloud and edge computing layers [2], as shown in Figure 1.

Corresponding author: Satish Narayana Srirama.

C. K. Dehury is with the Institute of Computer Science, University of Tartu, Estonia (e-mail: chinmaya.dehury@ut.ee).

S. N. Srirama is with the School of Computer and Information Sciences, University of Hyderabad, Gachibowli, Telangana, India (e-mail: satish.srirama@uohyd.ac.in).

P. K. Donta and S. Dustdar are with Distributed Systems Group, TU Wien, Vienna, Austria. (e-mail: {pdonta,dustdar}@dsg.tuwien.ac.at)

Manuscript received Aug XX, 2021; revised Month XX, 2021.

[1]https://www.techradar.com/news/rise-of-the-internet-of-things-iot

[2]https://www.mordorintelligence.com/industry-reports/internet-of-things-moving-towards-a-smarter-tomorrow-market-industry

However, in the case of Clustered Edge Intelligence (CEI), the main focus is on creating context-based intelligence clusters. In doing so, the primary focus has now shifted from clustering of devices to clustering of intelligence of several edge devices irrespective of their location and other characteristics. Such an approach may help reduce the system complexity [3]. While doing so, it is highly essential to consider the attributes of underlined edge devices, such as mobility nature, the type and purpose of the devices, what kind of data the device generates, and so on.

One of the research challenges in CEI is the ability to track the chain of tempered-proof historical events such as failure of actions/devices and performance degradation [4]. One of the conventional ways to track a past event is to analyze large size system logs collected from all the devices [4], [5]. The possibility of tampering the system logs is a matter of concern in such an approach [4].

In this article, we revisited the research challenge as mentioned above. We proposed a blockchain-based security mechanism that would enable the administrators/users to track the chain of immutable events to analyze and investigate the reason for any anomaly event. Blockchain is one of the most suitable candidate technology for the implementation of several distributed systems [3], [6]. This would secure the details of the device's actions/events and establish the chain among them within the clusters. Due to its promising features, such as distributed, traceability of shared information, transparency, immutability, and reduced costs, the application of blockchain technology is diverse, ranging from edge intelligence [7] to vehicular network [8] to different areas of smart city [9]. There are several works that used Blackchain for Edge computing and they are summarized using Table I. Majority of these approaches do not follow cluster or distributed approach, and they are also expensive in terms of complexity and resource requirements.

TABLE I: Summary on Literature study

Reference	Method	Advantages	Limitations
Rausch et al. [1]	Studied the scope of AI in edge computing	Scalable provisioning and monitoring, explainability, privacy and trust	Blackchain technology is not adapted for edge
Xie et al. [4]	Online anomaly detection using deep learning	Data integrity & automatic anomaly detection	Require more resources
Qie et al. [7]	Learning performed at edge and distributed the knowledge using the blockchain	Ability to handle resource heterogeneity and trust worthiness	The accuracy of the learning protocols is not verified.
Treiblmaier et al. [9]	Studied advantage of blackchain for Smart cities	Discuss the benefits of the blackchain	List the challenges of incorporating blackchain for smart cities
Dustdar et al. [10]	Elastic smart contracts between multiple Blockchains	Energy efficient task offloading, joint resource allocation, service migration framework, trust-aware data trading marketplace	High cost of transactions
Zhang et al. [11]	Distributed heterogeneous edge resource scheduling	Efficient resource handling, flexible & secure edge service management and cost minimization	Secure interoperable transactions are not possible
Song et al. [12]	Uses genetic algorithm along with the quantum particle swarm optimization	Minimize energy consumption, and QoS provisioning	Require more computational resources and time
Bartoletti et al. [13]	Location-based people and network-centric data analysis	Secure sharing and localization services	Blackchain technology is not adapted for edge devices

II. Background

This section gives a brief overview of clustered edge intelligence and how it differs from generic edge intelligence. Upon realizing the cloud computing capabilities in traditional methods, most intelligence is imposed on cloud servers following a centralized system. However, to mitigate the limitation of cloud computing itself, research has started to bring the intelligence to the proximity of devices at the edge of the network, referring to as edge intelligence [1]. In some cases, edge intelligence is also referred to as deploying a minimal version of Artificial Intelligence (AI) on resource constraint edge devices.

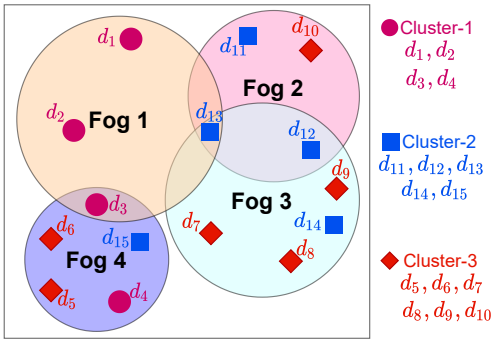


Fig. 2: Relationship of cluster members, clusters and fog environments.

To further minimize the network burden on the energy constraint edge device and reduce the system complexity, the edge devices are clustered based on their type, location or specific feature. Hence the driving factor to cluster the edge devices is to manage the devices and utilise the limited resources in an effective manner. For this, most of the researches focused primarily on the device-level management rather than the intelligence intended to be implemented in devices. The device-centric clustering mechanism introduces some bottleneck in an effective implementation of edge intelligence.

However, the CEI shifts the focus from the device to the intelligence itself. The primary goal of CEI is to allow the edge devices with similar and dependent intelligence to share their knowledge among peers and work in a collective manner towards a common objective. One such CEI example can be the energy usage analytics in a smart city scenario. With in this, multiple clusters can be formed: Analysing the green energy production by small solar panels deployed on city transportation (including buses and the bus stops), or the solar panels installed atop smart buildings, cold water usage analysis by the citizen, or in the public buildings. A cluster with an objective to analyze the green energy production by city transportation system involves both static solar panels installed atop bus stops having fixed location and mobile solar panels installed atop city buses.

These edge devices are known as the Cluster Members (CMs). Figure 2 briefly presents the relationship or organization of CMs, clusters, and the fog environments. In CEI, the members of a specific cluster may reside in different locations and may be connected to multiple fog environments. For instance the members of *Cluster-1*, i.e. edge devices d_1, d_2, and d_3 are located in the communication range of *Fog 1* and devices d_3 and d_4 are located in the communication range of *Fog 4*. So the device d_3 is in the communication range of both *Fog 1* and *Fog 4*. Similarly, *Fog 1*, *Fog 2* and *Fog 3* are the nearby fog nodes to device d_{13}, which is a member of *Cluster-2*.

The other aspect of CEI is the decision making system, which takes place mostly on the edge devices rather than in the control center. Each device is equipped with a basic intelligence on when to take an action and how to share the action. Some devices are responsible for collecting the data, known as *Collectors* and some other devices are responsible for reacting to the environment, known as *Reactors* based on the data from the collectors. For instance, in a CEI cluster with the objective to manage the energy consumption in smart building, a smart bulb may depend on the environmental data collected by the proximity sensors, light sensors and potentially on the temperature sensors. Hence, instead of relying entirely on a centrally located control center, the smart bulb should depend on the knowledge of proximity, light, and temperature sensors. In this example, the intelligence of these four sensors may form a CEI.

Based on the responsibilities, the CMs can be classified into two categories: *Collectors* and *Reactors*. Collectors are mainly responsible for collecting the surrounding data or the knowledge. Examples of collectors may include temperature

sensor, light sensor, solar panel, proximity and motion sensor, and CO_2 sensor. The job of the reactors is to react to the surrounding environment based on the shared knowledge by the collectors. Examples of reactor may include smart bulb, ventilation system, and room heating system. It is assumed that the CMs have the knowledge about other members in the same clusters and the nearby fog nodes.

One of the primary research issues in CEI is the security and the traceability of the event/data generated by diverse edge devices. Edge devices share their knowledge on surrounding environment among other CMs. As the edge devices are getting intelligent enough to act on their own, it is highly essential to securely gather the chain of events from all the devices for future analysis and validation purposes. For this, one of the most suitable technology that can be integrated to the current edge computing infrastructure is Blockchain, an emerging distributed ledger technology, as discussed in next section.

A. A short background on Blockchain

As the name suggests, Blockchain refers to a sequential chain of blocks or records that is nearly unalterable and transparent. Each block consists of three basic elements: *(a)* the data payload, *(b)* a nonce, and *(c)* hash of the previous block. Some of the primary research questions when the blockchain is integrated with CEI are as follows:

- Who, how, when, and where to construct the blocks or the records?
- How to secure the generated data payloads from the edge of the network?
- How to maintain the blockchain?

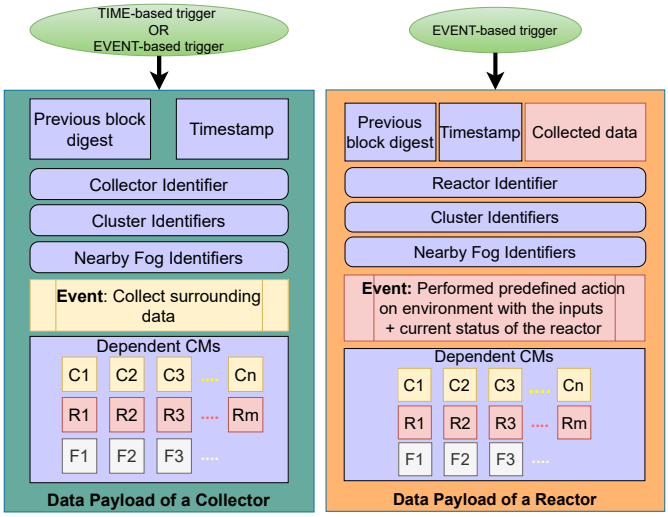
III. Blockchain-enabled CEI

This section details the integration of blockchain with the edge computing infrastructure that enables secure CEI. The objective here is to securely gather the chain of events from all the CMs for the analysis and validation purposes. As discussed in Section II, clusters are formed based on the intelligence capabilities of the edge devices, regardless of their location, type, and purpose. As shown in Figure 2 , it is possible for a single CM to be a member of multiple clusters.

A. Construction of data payload by CMs at the edge

The CMs are responsible for collecting the surrounding data/knowledge and sharing this event among other CMs in the same cluster and in the nearby fog nodes. For example, edge device d_3 would share its surrounding knowledge with all the CMs of *Cluster-1* and all the edge devices in the communication range of *Fog 1* and *Fog 4*.

For each event, the corresponding CM constructs a data payload with a set of additional information/artifacts. The data payload is then used to construct a block by the fog node. The content of a data payload depends upon the type of the CM, i.e. collector or reactor. Figure 3(a) and 3(b) show the set of data fields (described below) in the data payload constructed by collectors and reactors, respectively.



(a) (b)

Fig. 3: Data payload created by the (a) *Collector* (b) *Reactor*

- *Previous block digest*: It is similar to the hash value of the previous block in the blockchain. This value is always obtained from the nearby fog node which is responsible for constructing and maintaining the blockchain. However at time $t = 0$, this value is set to be empty.
- *Timestamp*: it is the time, at which the event is generated by the collector or the reactor.
- *Member Identifier*: The public key of each member is considered as the member identifier.
- *Cluster Identifiers*: This is the list of the unique identifiers of each cluster that the CM belongs to. In general, the cloud controller is responsible for assigning unique identifier to each cluster. For each CM, the cluster identifiers are set to be static unless there is manual updation.
- *Nearby Fog Identifiers*: Similar to above, this is a list of identifiers of nearby fog nodes. It is assumed that this field can not be empty, indicating that each edge device must be connected to at least one fog node.
- *Event*: *In case of a collector*, this data field contains the collected surrounding data. Using this value the current state of the environment can also be monitored, as shown in Figure 3a. However, in case of a *reactor*, this data field mainly contains the performed action and the current status of the reactor, as shown in Figure 3b.
- *Dependent CMs*: This is a set of other CMs including both collectors and reactors in the same cluster and the nearby fog nodes. All the generated events will be further shared among this set of edge devices and fog nodes.
- *Collected data*: This data field is very specific to the data payload generated by the reactor. Collected data is the surrounding environmental data shared by other CMs and this data acts as the input to the reactor. Based on this data, a reactor reacts to the environment.

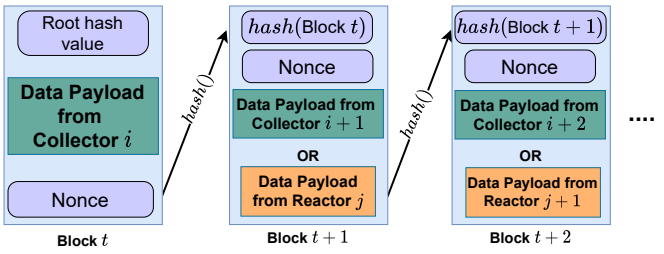


Fig. 4: Construction of sequential chain of block by Fog.

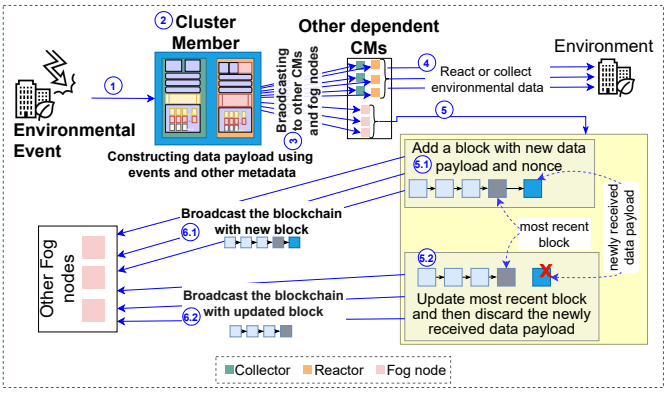


Fig. 5: The journey of a data payload, blocks and blockchain.

B. Constructing and updating the Blockchain

The CM broadcasts the newly constructed data payload to other dependent CMs and the near by fog nodes. A data payload constructed by an edge device is then used with the hash of the previous block in the blockchain and the *nonce* by the fog node to construct a block. Figure 4 shows a template of blockchain created by the fog environment. However, the first block, *Block t* in Fig 4, constructed by only a collector and contains a root hash value, which could be an empty value. This is based on the assumption that a reactor is always triggered by the event from other collectors or reactors, as shown in Figure 3b.

The *nonce*, also referred to as *Proof of Work (PoW)*, is a random special number that is used to resolve the conflicts that may arise when multiple fog nodes receive the same block from a CM. The nonce must meet the predefined cryptographic block hash value format and is calculated as the hash of the other two components: (a) received data payload of the collector or reactor and (b) the hash of the previous block in the blockchain, as shown in Figure 4. It can be observed from Figure 2 that *Fog 1*, *Fog 2*, and *Fog 3* always receive the data payload constructed by CM d_{13}. In this case, a conflict will arise between *Fog 1*, *Fog 2*, and *Fog 3* to update the blockchain. To handle such a conflict, the fog node that calculates the nonce before any other fog node would first broadcast the block to others and would be considered as the final version of the block. The hash or the digest of the previous block is mainly used to establish the connection among the sequence of the blocks/events.

The fog node does not just directly add the newly received block to the existing chain, rather it processes and analyzes the block beforehand. The fog node first verifies if the block is from an authorized CM. For this, fog node checks the cluster identifiers, collector/Reactor Identifier, and Previous block's digest. The other condition that needs to be verified is that if the data payload is from a reactor, the collected data in the received data payload should match the value of *Event* field of previous block, if the previous block is from a collector.

It is possible for any fog node to receive multiple consecutive data payloads from the same CM. In other words, the collector/reactor identifier of newly received data payload is same as that in the most recent block in the blockchain. If the CM is a collector, the fog node simply extracts the value of *event* data field from the newly received data payload and append that value to the *event* data field of the most recent block in the blockchain. However, if the CM is a reactor, the values of the *collected data* and *event* data field of the new data payload are extracted and appended to the respective data fields of most recent block in the blockchain. Following the append operation, the nonce would be recalculated by the same fog node. Once this process is done, the newly received data payload is simply discarded and the updated blockchain is broadcast to the other fog nodes in the network.

C. Operation Flow

This sub-section presents the detailed end-to-end flow, as shown in Figure 5, including the generation of data payload at the edge, forming the blockchain at fog, all network operations, and communications among all the entities. In the above subsections, the construction mechanism of data payload and blockchain by CM and fog are illustrated, respectively.

The entire process, as in Figure 5, starts when there is an event in the environment or a change in the state of the environment. The event can be based on the time. Such an event triggers the collectors that continuously monitor the state of the environment (step 1). Based on the event the corresponding collectors collect the environmental data and constructs the data payload with a set of other metadata, as discussed in Section III-A (step 2). For instance, if the outside luminosity level changes, the corresponding collector would be a light detector. If a person crosses a certain area, the motion detector would be triggered. The constructed data payload will then be broadcast to the other CMs and the nearby fog nodes (step 3). After receiving a new data payload from other CMs, a collector or reactor would extract the data payload's *event* data and accordingly either react to the environment or further collect the environmental data based on the predefined task.

On the other hand, upon receiving the same data payload, the nearby fog nodes either update the most recent block present in existing blockchain (step 5.1, if the newly received data payload is from the same CM) or add a new block to the existing blockchain (step 5.2, if the newly received data payload is from a different CM), depending on the source of the data payload. At this stage, all the fog nodes that receive the data payload will compete for finding the required nonce. Once this process is done, the updated blockchain will be further broadcasted to other fog nodes in the network (step 6.1 and 6.2). This way, all the fog nodes will have a list of updated blockchains.

IV. Challenges and Future Directions

The existing research on securing the CEI with blockchain faces numerous challenges due to the diverse characteristics of all the involved entities. In addition to network overhead, scalability, interoperability and technical validation based on real implementations, some of the other challenges that are also considered as future works are briefly discussed below:

- *Dynamic environment*: The nearby fog nodes of the mobile edge devices are never fixed. This introduces another level of complexity from constructing the data payload to updating the blockchain.
- *Validation rate*: Some edge devices may send the data payload faster than the rate at which the fog nodes would be able to update the blockchain. A lightweight consensus algorithm and block validation mechanism may need to be designed for the fog nodes in CEI to mitigate such an issue.
- *Computation overhead*: The nearby fog nodes are now responsible for both carrying out complex edge analytics and maintaining the blockchain for all the edge devices in its communication range. An efficient lightweight consensus algorithm and block validation mechanism can minimize such computation overhead.

V. Conclusions and future works

This paper addresses the security aspect of clustered edge intelligence, where the primary goal is to keep the events history intact and traceable. Events history analysis is required when it comes to find the cause of unexpected failure in edge intelligence or the reason behind the events. Blockchain technology can address this issue, by enabling the event to be immutable. As discussed, the edge devices are primarily responsible for building the data payload, which is the basic blockchain unit. Nearby fog nodes are mainly responsible for constructing and updating the blockchain. Once the blockchain is updated, it is hard for any system or intruder to modify the events chain, as the nonce/PoW is laborious to recalculate.

This paper also summarizes some of the significant challenges which are also part of the future works. We envision integrating advanced blockchain-based technology with the edge infrastructure will result in a more efficient and transparent secured CEI.

Acknowledgement

We thank financial support to UoH-IoE by MHRD, India (F11/9/2019-U3(A)).

References

[1] T. Rausch and S. Dustdar, "Edge Intelligence: The Convergence of Humans, Things, and AI," in *2019 IEEE International Conference on Cloud Engineering (IC2E)*, June 2019, pp. 86–96.

[2] Y. Du, Z. Wang, and V. C. M. Leung, "Blockchain-Enabled Edge Intelligence for IoT: Background, Emerging Trends and Open Issues," *Future Internet*, vol. 13, no. 2, p. 48, Feb. 2021. [Online]. Available: https://www.mdpi.com/1999-5903/13/2/48

[3] X. Shao, C. Yang, D. Chen, N. Zhao, and F. R. Yu, "Dynamic iot device clustering and energy management with hybrid noma systems," *IEEE Trans Industr Inform.*, vol. 14, no. 10, pp. 4622–4630, July 2018.

[4] X. Xie, Y. Fang, Z. Jian, Y. Lu, T. Li, and G. Wang, "Blockchain-driven anomaly detection framework on edge intelligence," *CCF Transactions on Networking*, vol. 3, no. 3, pp. 171–192, Dec. 2020.

[5] S. Xu, Y. Qian, and R. Q. Hu, "Data-driven network intelligence for anomaly detection," *IEEE Network*, vol. 33, no. 3, pp. 88–95, May 2019.

[6] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and G. Das, "Everything you wanted to know about the blockchain: Its promise, components, processes, and problems," *IEEE Consumer Electronics Magazine*, vol. 7, no. 4, pp. 6–14, July 2018.

[7] C. Qiu, X. Wang, H. Yao, Z. Xiong, F. R. Yu, and V. C. M. Leung, "Bring Intelligence among Edges: A Blockchain-Assisted Edge Intelligence Approach," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, Dec. 2020, pp. 1–6.

[8] N. Malik, P. Nanda, A. Arora, X. He, and D. Puthal, "Blockchain based secured identity authentication and expeditious revocation framework for vehicular networks," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 2018, pp. 674–679.

[9] H. Treiblmaier, A. Rejeb, and A. Strebinger, "Blockchain as a Driver for Smart City Development: Application Fields and a Comprehensive Research Agenda," *Smart Cities*, vol. 3, no. 3, pp. 853–872, Sep. 2020.

[10] S. Dustdar, P. Fernandez, J. María García, and A. Ruiz-Cortés, "Elastic smart contracts in blockchains," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 12, pp. 1901 – 1912, Dec. 2021.

[11] K. Zhang, Y. Zhu, S. Maharjan, and Y. Zhang, "Edge intelligence and blockchain empowered 5G beyond for the industrial internet of things," *IEEE Network*, vol. 33, no. 5, pp. 12–19, Oct. 2019.

[12] L. Song, K. K. Chai, Y. Chen, J. Schormans, J. Loo, and A. Vinel, "QoS-Aware Energy-Efficient Cooperative Scheme for Cluster-Based IoT Systems," *IEEE Syst J.*, vol. 11, no. 3, pp. 1447–1455, Sep. 2017.

[13] S. Bartoletti, L. Chiaraviglio, S. Fortes, T. E. Kennouche, G. Solmaz, G. Bernini, D. Giustinlano, J. Widmer, R. Barco, G. Siracusano, A. Conti, and N. B. Melazzi, "Location-Based Analytics in 5G and Beyond," *IEEE Commun. Mag.*, vol. 59, no. 7, pp. 38–43, July 2021.