An Adaptive LoRa-Based Edge Architecture for Data Sensing and Collection in Natural Parks

Benjamín Arratia*, Erika Rosas*, Ángel Martín Furones†, Ángel Perles-Ivars‡,
Juan-Luis Posadas-Yagüe*, José M. Cecilia*, Pietro Manzoni*
*DISCA
† Department of Cartographic Engineering, Geodesy and Photogrammetry
‡ITACA Institute
Universitat Politècnica de València, Spain

***Abstract*—This article introduces the design of a flexible IoT/Edge architecture to overcome the challenges of deploying advanced sensors in remote settings such as natural parks. These environments demand reliable communication and considerable computational resources to manage the extensive real-time data from IoT devices like sensors and cameras. Our approach combines two LoRa-based protocols to efficiently handle large data blocks and ensure long-range communication, conserving energy and maintaining data integrity crucial for integrating sophisticated sensors with conventional IoT devices.**

The architecture offers edge computing within a container-based gateway, simplifying on-site data processing and minimizing data traffic complexities. The MQTT protocol supports interoperability among distributed services, enhancing adaptability and efficiency in processing data locally or via cloud services like digital twins.

We demonstrate this system's effectiveness through two case studies in natural parks: visitor counting using Time of Flight technology and water level measurement with GNSS data. These cases highlight the system's ability to handle frequent small and less frequent large data transmissions efficiently.

I. INTRODUCTION

Sensors, cameras, and other devices can collect real-time data through the Internet of Things (IoT), making it useful for various applications. In rural and extreme settings, like natural parks, IoT solutions require a reliable communication infrastructure, which poses several technical challenges. Moreover, the integration of Artificial Intelligence with IoT (AIoT) has transformed the data collection processes. AI enhances predictive analysis and supports informed decision-making. However, deploying AI models typically requires substantial computation resources, which are complex to deploy extensively in remote or protected natural environments. Also, in AIoT, big chunks of data must often be handled and transferred, for example, from cameras, microphones, or other advanced sensors.

These challenges require a data collection framework where data is gathered and processed at or near the collection site using edge computing techniques before being sent to the cloud for further analysis; the need for low energy consumption and reduced physical intervention mandate efficient, remote data transmission technologies. Moreover, standard protocols, such as pub/sub protocols like MQTT, must be used for various services to interconnect and compute a result at the edge node.

This article, to handle these requirements, presents a flexible IoT/Edge architecture that effectively combines LoRaWAN [1] with the LoRa-based protocol *AlLoRa* [2], which is designed to handle large data blocks efficiently while maintaining long-range communication distances. This dual-protocol approach ensures energy efficiency and maintains comprehensive coverage and high data integrity, allowing for the integration of both classical IoT devices and enhanced sensors. Additionally, the architecture embeds edge computing capabilities in a container-based gateway, allowing on-site data processing and simplifying traffic between different system parts, such as between the Internet and a private network or between different services within a cloud environment.

The flexibility and adaptiveness of this architecture are illustrated through two practical applications for natural parks: (1) visitor counting with Time of Flight (ToF) technology and (2) water level estimation using GNSS data. These case studies provide insight into how the system can collect and process data at the edge. The first use case employs LoRaWAN to handle frequent, smaller data transmissions typical in visitor counting, while the second use case leverages *AlLoRa* for less frequent but larger data transmissions necessary for accurate water level estimation.

The outline of this article is as follows: Section II reviews LoRa technology and IoT architectures. Section III details the IoT edge architecture and the adaptive gateway's design. Section IV explores the two aforementioned use cases, demonstrating the architecture's versatility. Section V discusses the prototypes' implementations and field deployments. Finally, Section VI concludes with our findings and directions for future research.

II. BACKGROUND AND RELATED WORK

Low-Power Wide Area Network (LPWAN) technologies such as SigFox, NB-IoT, and LoRaWAN are increasingly used in IoT applications for long-range, low-energy communication [3]. These technologies provide robust modulation for extensive coverage, making them ideal for environmental monitoring tasks like animal tracking, smart agriculture, and flood forecasting [4], [5].

LoRaWAN extends LoRa [6] by defining the MAC layer and employing a star-of-stars topology, allowing gateways to relay messages between end devices and a central server. It supports data rates up to 21.9kb/s, controls device duty cycles to avoid congestion, and uses the ALOHA protocol with orthogonal Spreading Factors [1]. Its adaptive data rate improves communication efficiency and battery life, ideal for areas with irregular connectivity.

The *AlLoRa* protocol, built on LoRa, facilitates larger data transfers in remote areas using a requester-gateway model with polling. It enables *AlLoRa* Gateway Nodes to request data, with Source nodes responding by sending segmented data. This protocol also supports mesh networking to enhance reach over more extensive and challenging terrains and includes configurable scheduling to optimize network performance and ensure fair data distribution among nodes [2].

Recent advancements have enhanced IoT gateways, transforming them from simple message relays into intelligent systems capable of processing data at the edge. This evolution reduces latency and increases efficiency. For example, Tran-Quang et al. [7] developed an IoT gateway supporting multiple protocols like LoRa, 4G/LTE, and Wi-Fi, showcasing its adaptability for various IoT applications. Similarly, Zhang et al. [8] created a latency-aware hardware prototype that optimizes task processing directly at the gateway level, thereby decentralizing data processing functions closer to the data source.

In parallel, other research focuses on cost-effective gateway solutions. Sun et al. [9] engineered a single-channel LoRa gateway using a Raspberry Pi and a LoRa extension designed to efficiently manage connectivity and data flow. Zhong et al. [10] enhanced gateway reliability by incorporating acknowledgment features, crucial for real-time applications, though not advancing in data processing. Chan et al. [11] merged LoRa and NB-IoT technologies within a gateway architecture that employs MQTT at the cloud level for better integration.

These studies highlight a range of gateway functionalities, from basic, cost-saving designs to sophisticated systems with comprehensive edge computing features. This diversity reflects various strategic approaches and technological advancements in IoT infrastructure.

Our work integrates two LoRa-based protocols, focusing on energy saving and covering long ranges. Through the edge approach and using a pub/sub solution based on a containerized gateway, we enable computing at the edge and ensure interoperability among different distributed services.

III. ARCHITECTURE OF THE PROPOSED SYSTEM

This section presents the architecture of our solution, which is designed for data acquisition and processing in rural IoT contexts. Our adaptive architecture enables low-power, long-distance connectivity for a wide range of sensors that might require some processing before getting to the cloud.

The IoT/Edge architecture uses a pub/sub model for data management and is organized into three distinct layers, as shown in Figure 1:

1) *Sensors:* The data is generated from the sensors equipped with a LoRaWAN or *AlLoRa* communication device. Different sensors can measure various environmental parameters, such as visitor counts, wildlife monitoring, and environmental conditions. These sensors typically operate in remote locations with autonomous energy capabilities or harvest energy from solar panels.
2) *Edge Gateway:* The primary data aggregator. It allows data from the sensors and possibly from the cloud to be combined using a container-based approach to perform local computation and forward the results to a cloud-based service. It incorporates a local network server, gateway functionality, and application processing capabilities. The number of gateways depends on the area to be covered. Using a pub/sub system allows for integrating various gateways among them and with the cloud.
3) *Cloud:* A flexible, optional cloud component enhances data accessibility and extensive processing capabilities. For example, a digital twin can be integrated to allow for the additional processes and visualization of the data coming from the scenario being sensed.

The main component of this architecture is the Edge Gateway. It orchestrates communication between different protocols and manages data collection and processing. The gateway supports two protocols that leverage LoRa technology to enable long-range communication: LoRaWAN for regular, smaller data packet transmission and *AlLoRa* for on-demand, larger file sizes, enabling the handling of basically unlimited data-size payloads. *AlLoRa* can augment the system's reach also through mesh network functionality.

LoRa's orthogonal SF allows *AlLoRa* (single-channel) to operate concurrently with LoRaWAN, minimizing cross-interference for diverse data transmission in IoT networks. While imperfect orthogonality can affect link performance [12], our implementation has experienced minimal interference, largely due to the low frequency of *AlLoRa*'s large data transmissions.

The Edge Gateway orchestrates a pub/sub model utilizing an MQTT Broker as the central communication hub. Upon reception at the Gateway, the data is immediately published to the MQTT Broker. For *AlLoRa*, the communication is facilitated by an *AlLoRa* Source node present within the sensor and an *AlLoRa* Gateway node situated at the Edge Gateway. The latter also functions as an MQTT client. In the case of LoRaWAN, an embedded Networks Server based on the open source ChirpStack[1], assumes the role of data receiver and publisher to the MQTT broker. Data integrity is maintained through checksums in *AlLoRa* and built-in redundancy in LoRaWAN, with retransmissions triggered when necessary.

Edge-level data processing is facilitated by the computing module within the Gateway, where algorithms housed within Docker containers subscribe to relevant MQTT topics, collect and locally process the necessary data, and then publish the results back to the same MQTT broker.

---

[1] https://www.chirpstack.io/

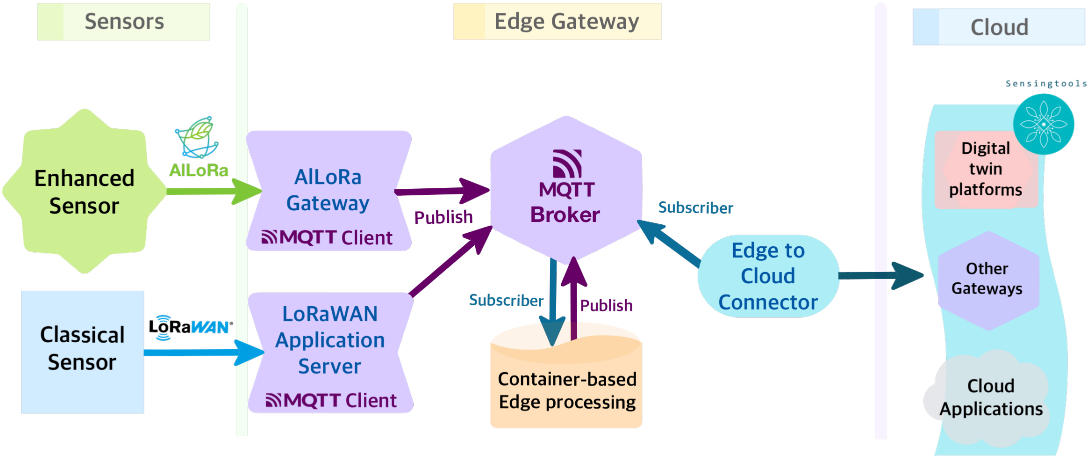

Fig. 1: Architecture solution overview

The Edge to Cloud Connector subscribes to the MQTT Broker, aggregating and channeling edge data to cloud services. The system integrates with any cloud platform; in our case, we use the *Sensing Tools*[2] platform via a REST API. This component ensures smooth communication between edge devices and the cloud, handles network interruptions and supports flexible integration for scalable, robust architecture.

This architecture encapsulates a comprehensive data management strategy, spanning initial data acquisition at the sensor level to refined processing within the cloud. It highlights the role of MQTT in unifying communication across heterogeneous sensor types and networking protocols.

IV. USE CASES: NATURAL PARK MONITORING

The adaptive IoT/Edge architecture facilitates a broad spectrum of information-based solutions in natural park management. This framework supports the deployment of simple sensors like temperature, humidity, and sound sensors within designated climate refuges to monitor and analyze microclimatic conditions, up to advanced sensors for sentiment analysis at scenic viewpoints, processing verbal visitor feedback to evaluate their experiences, where visitor impressions can significantly influence service improvements and conservation strategies.

This section will explore two specific use cases: one utilizing *AlLoRa* for high-capacity data tasks and the other employing LoRaWAN for frequent, smaller data transmissions. These examples will illustrate the practical applications and effectiveness of the proposed adaptive architecture in natural park settings.

A. GNSS-based water level estimation at the Edge

Monitoring large water bodies in natural parks poses unique challenges. For example, tracking the level of a salty lagoon does not allow the use of a sensor in the water since, due to the high salinity of the environment, the sensor itself could degrade quickly. We explore the application of a GNSS-IR technique, initially designed for manual execution on a traditional server architecture.

The GNSS-IR technique uses GNSS antenna signal-to-noise (SNR) ratio data to monitor environmental changes. SNR data from satellites at 5 to 25 degrees elevations are used to create precise Lomb Scargle Periodograms. This data helps isolate the dominant multipath frequency, which is then modeled to extract information like soil moisture or water levels. By identifying a primary wave from the periodogram, methods such as least squares regression (Martin et al. [13]) or inverse modeling (Purnell et al. [14]) are used to estimate environmental variables such as amplitude, phase, and distance from the GNSS antenna to the reflective surface.

To implement this algorithm, we require continuous satellite data collection throughout the day, which is processed to estimate water levels during that period when combined with real-time online information. This method allows us to leverage a comprehensive data set, ensuring our water level assessments are accurate and timely. By integrating satellite and real-time data, we can adjust our models and forecasts to reflect current conditions, thereby improving our response to potential changes in water levels.

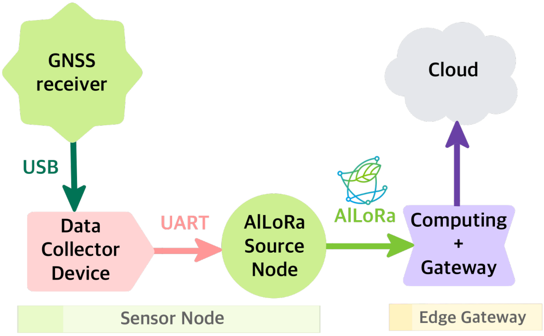

Fig. 2: Water level estimation components

Figure 2 presents the solution's conceptual design. At the

---

[2] https://sensingtools.com/

Sensor Node level of the architecture, the system can collect GNSS data and transmit it hourly via *AlLoRa* to the Edge Gateway. The Edge Gateway processes this incoming data using algorithms that integrate it with GNSS navigational files downloaded from the internet. After processing, the results are uploaded to the Cloud, facilitating remote access and further analysis.

The data collection is conducted using a low-cost *GNSS receiver* with an antenna capable of receiving signals across GPS, GLONASS, and GALILEO systems frequencies. The GNSS receiver is interfaced with a processing module (*Data collector Device*) responsible for retrieving data from the satellites and storing it locally.

After data collection, it is transmitted via *AlLoRa* through a LoRa-capable device connected to the *Data Collector Device* via UART. A daemon on the processing module sends compressed files to the *AlLoRa* Source Node, which stores them until the Edge Gateway polls the GNSS Sensor Node for retrieval. The polling frequency, set to once per hour, reduces unnecessary messages. If polling fails, data is stored locally and sent during the next cycle, ensuring no information is lost, even with occasional connectivity issues.

We used the inverse modeling method to estimate the water level using GNSS data collected from the field, which was originally not constrained by computational limits. To adapt this algorithm for edge computing, we have refactored and optimized the code to run efficiently within the capabilities of the Edge Gateway. The software has been modularized and containerized to streamline input/output management, enhancing maintainability and scalability. This containerized service subscribes to the topic receiving GNSS data from the Sensor Node via *AlLoRa*. Daily, the service downloads necessary GNSS navigation files and subsequently executes the algorithm. The results of these estimations are then transmitted to the MQTT broker located at the gateway and subsequently published to the selected cloud service.

B. Visitor counting at remote locations in Natural Parks

This case study manages visitor statistics at key entry points and trailheads, reducing infrastructure needs and maintenance frequency. The data helps optimize crowd management, maintenance scheduling, and resource allocation.

ToF sensors emerge as a robust solution for outdoor environments due to their resilience against various climate conditions [15]. While existing systems may rely on 4G/5G for data transmission, the associated costs are often excessive and not scalable; more cost-effective solutions typically depend on Wi-Fi access, which is impractical for remote locations in natural parks. To address these constraints, we have devised a solution that pairs a people counting sensor with a device that has LoRa capabilities, thus forming a sensor node adapted for remote deployment.

Figure 3 shows the components of our proposed system. The sensor node, positioned remotely within the park, comprises two devices: the People Counter and a LoRa-enabled module. The People Counter component generates data on visitor traffic, which it then transmits via the MQTT protocol to a broker. This process requires a Wi-Fi hotspot, which is provided by the accompanying LoRa-capable device.

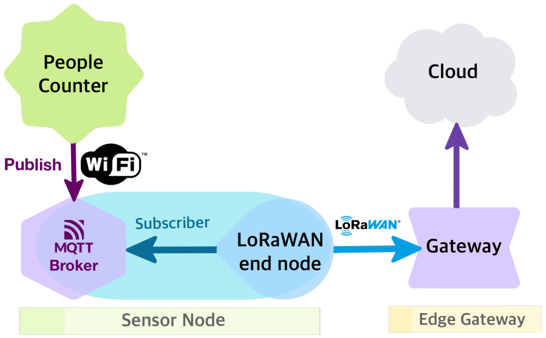

Fig. 3: Visitor counting components

This intermediary node hosts a service that subscribes to the MQTT broker. It employs LoRaWAN to transmit the information to the gateway upon receiving data. As there is no need for edge processing in this context, data is relayed to the cloud immediately upon gateway reception. This setup ensures that visitor data is collected and communicated effectively, even from the park's remote areas.

V. PROTOTYPE DESIGN AND DEPLOYMENT

We developed a prototype of the architecture for both study cases. The Edge Gateway was built using a RAK7391[3] Gateway powered by a Raspberry Pi 4 compute module with an inbuilt LoRaWAN gateway. *AlLoRa* is integrated through a custom solution, as delineated in Figure 4, utilizing a LoRa-enabled ESP32-S3 device (LILYGO T3S3 V1.2[4]) connected to the Raspberry Pi through serial interface. A dedicated HAT (Hardware Attached on Top) ensures seamless communication between the ESP32-S3 and the Raspberry Pi module. This Edge Gateway was deployed in a research project focusing on the La Mata-Torrevieja Natural Park, Alicante, Spain.

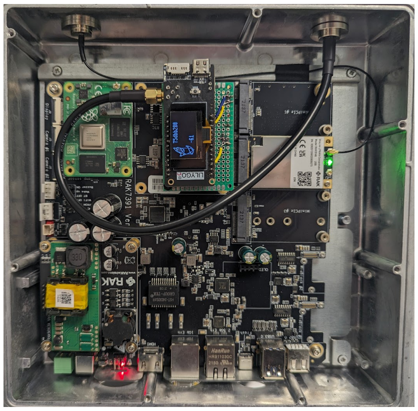

Fig. 4: RAK Gateway

---

[3] https://store.rakwireless.com/products/wisgate-connect-base-kit-rak7391

[4] https://www.lilygo.cc/products/t3s3-v1-0

A. GNSS Reflectometry testing

To assess the system's functionality in an environment similar to its intended deployment site at La Mata-Torrevieja Natural Park (Alicante) we established a test setup on the outskirts of Valencia. We installed the *AlLoRa* integrated GNSS Reflectometry and Source Node system and conducted a 10-day operational test monitoring a local irrigation reservoir where no water level variation was expected.

Fig. 5: Sensor Node implementation

We used a Navilock NL-8022MU Multi GNSS Receiver u-blox 8[5], capable of processing signals from up to 72 satellites and supports the NMEA 0183 protocols. This device offers an optimal balance between cost and performance, providing a cost-effective solution without significantly compromising accuracy. The receiver connects via micro-USB to a Raspberry Pi Zero W (1GHz CPU, 512MB RAM), where a service stores the collected data and generates a new file every hour. A LILYGO T3S3 V1.2 serves as an *AlLoRa* Source Node, receiving compressed files from the Raspberry Pi Zero via serial communication and storing them on a micro-SD card. The files are then transmitted to the Edge Gateway upon request. Figure 5 displays the initial prototype of the Sensor Node for this test.

Fig. 6: Experimental setup and location for GNSS Reflectometry testing

The Edge Gateway was placed 3.26 kilometers away at a height of 10 meters to ensure a clear line of sight. The system autonomously collected, compressed, and transmitted satellite data hourly. File sizes ranged from 46 to 166 KB, depending on the satellite data captured, and were compressed to 7–9 KB before transfer to the ESP32-S3.

Each *AlLoRa* file's data transmission duration varied between 30 and 45 seconds, and they were successfully sent hourly. Once the data is received in the Edge Gateway, it follows the steps described in IV-A, where the final results are sent to the cloud for availability.

The 10-day test of our GNSS Reflectometry prototype confirmed the GNSS receiver's reliability and cost-effectiveness. With ±2.5 cm accuracy in water level detection using GPS and GALILEO satellites (Figure 7), our system rivals more expensive devices, proving that high accuracy can be achieved affordably.

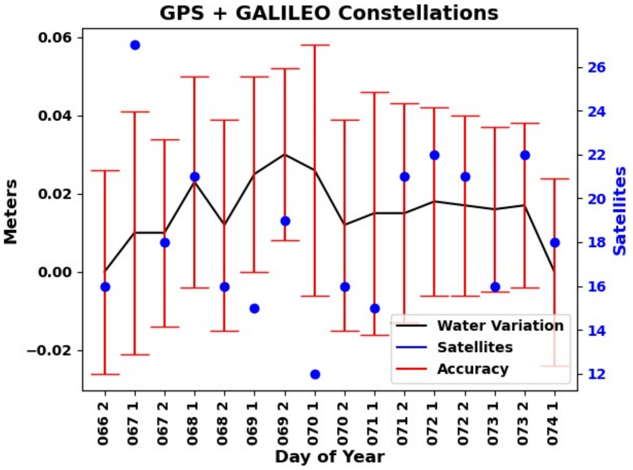

Fig. 7: GNSS water level analysis

The system uses the *AlLoRa* protocol for deployment over large and remote areas, reducing the logistical challenges and costs of traditional sensors that require manual data collection. This remote data collection method boosts efficiency and reduces the need for physical intervention, making advanced technology affordable and widely applicable for environmental monitoring.

B. Visitor counting tests

For the visitor counter system, we used a SensMax TAC-B 3D-WP outdoor people counting radar[6] at the Natural Park Lagunas De La Mata-Torrevieja visitor center's entrance. This sensor has Wi-Fi connectivity, which was a limiting factor when deploying it in a remote location like the one we intended. To solve this, we implemented a LoRaWAN end node to Wi-Fi adapter using a LILYGO T3S3 V1.2 that provides a Wi-Fi hotspot to the sensor and enables the MQTT Broker for the sensor to publish to it. The T3S3 also acts as

---

[5] https://www.navilock.com/produkt/62532/merkmale.html

[6] https://sensmax.eu/devices/sensmax-tac-b-3d-w-radar-people-counter/

a LoRaWAN end node that transmits the updates to the main Gateway.

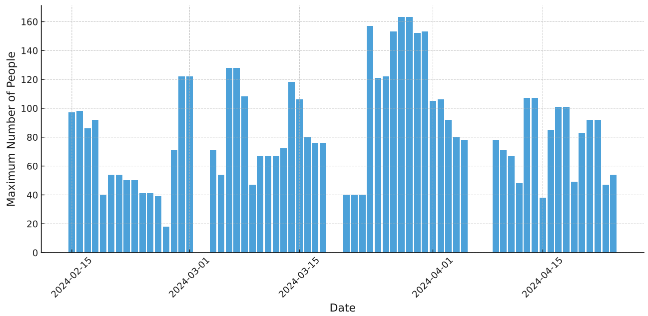

Fig. 8: Maximum Daily Park Attendance from February 15, 2024, Onwards

Figure 8 illustrates the data presentation on the Sensingtools' dashboard, designed to offer valuable insights to visitors and park administration. The dashboard effectively displays real-time and historical data, allowing for a comprehensive understanding of visitor trends and park usage. This enhanced visibility aids in better resource management and improves the overall visitor experience by allowing the administration to make informed decisions regarding maintenance and services based on actual usage patterns. Additionally, visitors benefit from this system by receiving updates on park occupancy and optimal visit times, enhancing their overall experience.

VI. CONCLUSION

This article presented a robust IoT/Edge architecture to meet the unique demands of deploying advanced sensors in challenging environments like natural parks. By merging two LoRa-based protocols, our system ensured long-range communication and efficient handling of large data blocks, significantly reduced energy consumption, and maintained high data integrity. This was vital for integrating advanced sensors with standard IoT devices.

Incorporating edge computing within a container-based gateway facilitated on-site data processing and reduced data traffic complexities. Furthermore, integrating pub/sub protocols supported computing capabilities at the edge and enabled interoperability among diverse distributed services. Our architecture demonstrated exceptional adaptability and efficiency, and we are capable of processing data directly at collection sites or through cloud-based services such as digital twins.

Through practical applications in natural parks—visitor counting with ToF technology and water level estimation using GNSS data—our studies confirmed the system's proficiency in effectively managing frequent small and periodic large data transmissions. These case studies underscored the practicality and scalability of our IoT architecture.

Future work includes improving the robustness and security of the *AlLoRa* communication protocol. Additionally, integrating advanced AI and machine learning algorithms to predict environmental changes more accurately and further developing the digital twin of the monitored environment in the Cloud platform could further increase the efficacy and responsiveness of this IoT/Edge architecture.

ACKNOWLEDGMENT

This work has been supported by the MCSA-SE REMARKABLE grant ID: 101086387; Research project CIPROM/2023/29, funded by "Direcció General de Ciència i Investigació" Generalitat Valenciana - SPAIN; EU's Horizon 2020 research and innovation program No 101017861, by project Ramón y Cajal Grant RYC2018-025580-I, funded by MCIN/AEI/ 10.13039/501100011033 and ESF Investing in your future, GVA GRISOLIAP/2021/103 and Grant TED2021-130890B funded by MCIN/AEI and by EU NextGenerationEU/PRTR.

REFERENCES

[1] "LoRaWAN v1.1 Specification," Available online: https://lora-alliance.org/resource_hub/lorawan-specification-v1-1/ (Accessed: 2023-02-10).

[2] B. Arratia, E. Rosas, C. T. Calafate, J.-C. Cano, J. M. Cecilia, and P. Manzoni, "AlLoRa: Empowering environmental intelligence through an advanced LoRa-based IoT solution," *Computer Communications*, vol. 218, pp. 44–58, 2024.

[3] C. Milarokostas, D. Tsolkas, N. Passas, and L. Merakos, "A Comprehensive Study on LPWANs With a Focus on the Potential of LoRa/LoRaWAN Systems," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 825–867, 2023.

[4] P. Majumdar, D. Bhattacharya, S. Mitra, and B. Bhushan, "Application of Green IoT in Agriculture 4.0 and Beyond: Requirements, Challenges and Research Trends in the Era of 5G, LPWANs and Internet of UAV Things," *Wirel. Pers. Commun.*, vol. 131, pp. 1767–1816, 2023.

[5] L. González, A. Gonzales, S. González, and A. Cartuche, "A Low-Cost IoT Architecture Based on LPWAN and MQTT for Monitoring Water Resources in Andean Wetlands," *SN Computer Science*, p. 144, 2024.

[6] Semtech, "LoRa and LoRaWAN: A Technical Overview," Semtech Corporation, Tech. Rep., 2020.

[7] V. Tran-Quang, H. D. Nguyen, and D. T. Tien, "Design and Implementatioin of a LoRa Communication System Supporting Edge Computing on the Smart Multi-Platform IoT Gateway," *JST: Smart Systems and Devices*, vol. 31, 2021.

[8] X. Zhang and Z. Huang, "A Design and Implementation of Decentralized Edge Intelligent LoRa Gateway," in *2023 IEEE 26th International Symposium on Real-Time Distributed Computing*, 2023, pp. 195–200.

[9] C. Sun, F. Zheng, G. Zhou, and K. Guo, "Design and Implementation of Cloud-based Single-channel LoRa IIoT Gateway Using Raspberry Pi," in *2020 39th Chinese Control Conference (CCC)*, 2020, pp. 5259–5263.

[10] C. Zhong and X. Nie, "A novel single-channel edge computing LoRa gateway for real-time confirmed messaging," *Scientific Reports*, vol. 14, p. 8369, 2024.

[11] Y.-W. Chan, E. Kristiani, H. Fathoni, C.-Y. Chen, and C.-T. Yang, "A smart edge computing infrastructure for air quality monitoring using LPWAN and MQTT technologies," *The Journal of Supercomputing*, vol. 80, pp. 9961–9985, 2024.

[12] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello, "Impact of LoRa imperfect orthogonality: Analysis of link-level performance," *IEEE Communications Letters*, vol. 22, no. 4, pp. 796–799, 2018.

[13] A. Martín, R. Luján, and A. B. Anquela, "Python software tools for GNSS interferometric reflectometry (GNSS-IR)," *GPS Solutions*, vol. 24, no. 4, p. 94, 2020.

[14] D. J. Purnell, N. Gomez, W. Minarik, D. Porter, and G. Langston, "Precise water level measurements using low-cost GNSS antenna arrays," *Earth Surface Dynamics*, vol. 9, no. 3, pp. 673–685, 2021.

[15] M. Stec, V. Herrmann, and B. Stabernack, "Using Time-of-Flight Sensors for People Counting Applications," in *2019 Conference on Design and Architectures for Signal and Image Processing*, 2019, pp. 59–64.