A Novel Scalable Cloud-enabled Spectrum Sensing Architecture

Partemie-Marian Mutescu
Interdisciplinary Cloud and Big Data Center at Ştefan cel Mare University of Suceava
Suceava, Romania
marian.mutescu@usm.ro

Alin Calinciuc
ASSIST Software
Suceava, Romania
alin.calinciuc@assist.ro

Valentin Popa
Computers, Electronics and Automation Department, Stefan Cel Mare University of Suceava
Suceava, Romania
valentin.popa@usm.ro

Ovidiu-Andrei Schipor
Interdisciplinary Cloud and Big Data Center at Ştefan cel Mare University of Suceava, Romania
ovidiu.schipor@usm.ro

***Abstract*—Spectrum Sensing techniques have evolved significantly, transitioning from traditional approaches to more complex Machine Learning and Deep Learning based approaches. This evolution has led to enhanced performance capabilities but has also introduced substantial computational demands, which in turn have constrained the scalability of large-scale deployments due to associated costs. Nevertheless, the advent of Cloud Computing offers a promising solution by enabling the centralized processing of data at the cloud level, thereby reducing the computational load on field-level sensors. This shift facilitates cost-effective deployment and enhances the scalability of spectrum sensing applications. In this study, we introduce and evaluate a Cloud-Enabled Spectrum Sensing architecture, with a specific focus on the computational and data transmission requirements from the perspective of the sensing node. We explore and compare two data formats for radio signal processing: I/Q samples and FFT series. Our evaluation covers the computational demands for processing and transmitting these data formats to a Cloud Data Fusion Center across a range of radio spectrum bandwidths. Moreover, we examine the latencies involved in data transfer. Our analysis reveals that the FFT series data format offers considerable advantages for this architecture, achieving an optimal balance between computational requirements and network load. Specifically, for the recording and transmission of a 30 MHz wide radio band using the TCP protocol, the FFT format requires only 0.3 GFLOPS for processing and 6.7 MB/s for data transfer. These findings underscore the potential of the FFT series as a highly efficient radio signal representation format for Cloud-Enabled Spectrum Sensing architectures, promising significant improvements in scalability and operational efficiency.**

Keywords—Cloud Computing, TCP, Spectrum Sensing, WSN, IQ, FFT

I. Introduction

In recent years, the deployment of Wireless Sensor Networks (WSNs) within the Internet of Things (IoT) paradigm has experienced significant growth due to the large potential for applicability in multiple applications such as environment monitoring [1], human machine interaction systems [2], [3], [4] and healthcare [5],[6]. This expansion has led to an increase in the number of WSNs, each requiring access to radio spectrum resources for the transmission of data collected from the sensors to gateways, which then relay this data to the internet for subsequent storage and analysis. Predominantly, these networks employ Low Power Wide Area Network (LPWAN) wireless protocols, such as LoRaWAN [7] and Sigfox [8], for data transmission to gateways. These protocols are primary using the 868 MHz Short Range Device (SRD), 915 MHz Industrial, Scientific, and Medical (ISM), and 2.4 GHz ISM frequency bands, all of which offer limited bandwidth. Furthermore, the majority of LPWAN protocols utilize the ALOHA [9] method, which does not provide acknowledgment for transmitted data packets. Given these constraints, the increase in connected wireless sensors is anticipated to generate interference, collisions, data loss, and consequently, a low Quality of Service (QoS) and constrained scalability of WSNs.

To address these challenges, in years there has also been a notable advancement in Spectrum Sensing (SS) techniques. SS encompasses a suite of methodologies aimed at collecting radio signals from the radio spectrum, processing them, and analyzing specific metrics as to determine its current state. The applications of SS are diverse, ranging from the detection of the presence or absence of radio transmissions within a frequency band [10], identification of the wireless communication protocol or modulation being used [11], to anomaly detection [12]. These developments in spectrum sensing techniques are set to overcome the aforementioned issues, enhancing the efficiency and scalability of WSNs in the IoT ecosystem.

However, spectrum sensing techniques have seen a shift from traditional signal processing approaches, to methodologies that are based on machine learning (ML) and deep learning (DL) algorithms [11], [13], [14], [15], [16]. These novel methodologies offer a significant leap in performance over traditional methods like energy detection [17], but come with higher computational demands. These demands can be effectively managed using GPU hardware acceleration, however, such a SS architecture wouldn't be scalable, as each sensing node in the SS architecture would require high-end hardware configuration, implying a high implementation cost.

The challenges posed by the high computational demands of modern SS techniques, particularly those based on ML and DL algorithms, can be effectively addressed through cloud computing. Cloud computing enables individuals and organizations to access and utilize computing resources, such as servers, storage, databases, networking, software, analytics, and AI over the internet. It offers the ability to scale computing resources quickly and efficiently, which often leads to cost savings. Additionally, cloud computing provides high

979-8-3503-4929-0/24/$31.00 ©2024 IEEE

flexibility, improved performance, and reduced maintenance requirements for IT resources. Considering this, in this paper, we propose and analyze a Cloud-Enabled Spectrum Sensing (CESS) architecture. This architecture delegates all necessary computational tasks for data processing to a centralized entity in the cloud. Meanwhile, the sensing nodes are tasked solely with gathering the spectrum data and forwarding it to the cloud. This approach leverages the power of cloud computing to handle the intensive computational needs of advanced spectrum sensing methods, thereby simplifying the role of sensing nodes and enhancing the overall efficiency and scalability of the system. Our analysis focuses on the architecture from the perspective of the sensing node. Initially, we undertake a comprehensive examination of the most efficient data format, considering the computational needs of the sensing node and the internet bandwidth necessary for transmitting the collected spectrum data to the cloud across various recorded signal bandwidths. Following this, we analyze the latencies associated with the process of data transmission.

The structure of this paper is organized as follows: Section II provides an overview of our proposed architecture for CESS. Section III presents the implementation details of our architecture, outlining the specific methodologies and technologies utilized. Section IV presents the results obtained from our implementation, along with an analysis of the performance metrics of our CESS architecture. Finally, Section V concludes the paper, summarizing the key findings and contributions of our work.

II. Cloud-enabled Spectrum Sensing

As stated in the introduction, the literature has shown in the past years a transition from the traditional SS approaches to ML and DL approaches, due to the advantages it brings in terms of performance, accuracy and the versatility of applications, moving from the detection of unused frequency channels to wireless protocol or modulation detection, spectrum anomaly detection, and even wireless network optimization. If the latter approaches based on traditional signal processing methodologies could be easily deployed at the sensing node level, as they required low computational resources [18], for example, in the case of energy detection, the node would have to calculate the total energy in a frequency band and compare it to a threshold, the ML and DL approaches require high computational resources for inference. For example, in one of our previous papers [11], we used the second iteration of the YOLO architecture [19] for performing IoT technology detection and classification, which roughly needs 8.52 GFLOPs in order to perform detection. Other approaches in the literature show the use of Autoencoder networks for anomaly detection [12], residual CNNs [16], or image classification CNNs [20], all requiring high computational resources, as it has been shown multiple times that shallow networks with a low number of layers and parameters can generate suboptimal accuracies.

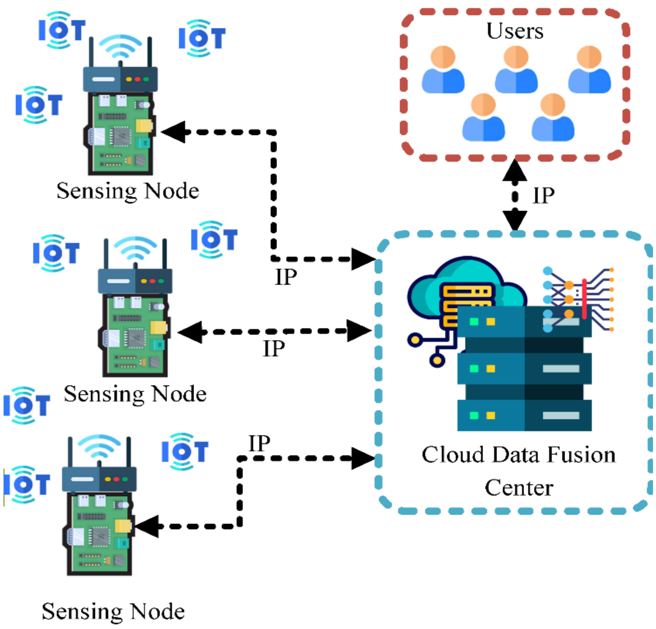

Fig. 1. Overview of the Cloud-enabled Spectrum Sensing architecture

However, as we have previously highlighted, the computational demand challenges of ML and DL algorithms can be significantly reduced through cloud computing services. Recent progress in technology has demonstrated that ML and DL services can run efficiently on cloud platforms, and effectively reallocating the processing workload from local devices to the more potent computational resources found in the cloud [21], [22]. This setup allows users to leverage advanced ML/DL algorithms without the need for extensive computing power on their end. A leading example of such innovation is OpenAI's ChatGPT [23], a cutting-edge AI chatbot, which offers AI processing services through API interfaces. Furthermore, deploying AI in the cloud has the added benefit of centralizing AI operations to a few core locations in the cloud, thereby eliminating the need to individually update each device running the AI algorithm. This approach not only simplifies the management of ML/DL deployments but also enhances the scalability and flexibility of AI applications in various domains. Building on this premise, we introduce CESS architecture as illustrated in Fig.1. This architecture is split into two main components. The core of this system is the Cloud Data Fusion Center (CDFC), a centralized hub where data from various sensing nodes is aggregated. At the CDFC, each sensing node is assigned a specific processing pipeline dedicated to data collection, AI-driven processing, and detection. The results of these detections are forwarded to an application, ensuring that the information is readily accessible to users whenever required.

The second component comprises the sensing nodes themselves. As previously mentioned, these nodes are designed to fulfill minimal processing requirements, primarily focusing on the collection of spectrum data. This data is then transmitted to the CDFC via internet protocol for subsequent analysis and processing. To enhance the architecture's deployment flexibility and adaptability, we advocate for the utilization of Software Defined Radios (SDRs) in the data gathering process. SDRs offer the advantage of being easily reconfigurable and tunable, avoiding the need for physical hardware modifications. This approach not only streamlines the process of spectrum data collection but also significantly expands the potential applications and efficiency of the CESS architecture.

III. Implementation and Analysis of the Cloud-enabled Spectrum Sensing Architecture

In this section we provide the implementation details of our proposed CESS architecture, along with the methodology for determining the performance requirements of such an architecture. The CESS architecture we implemented

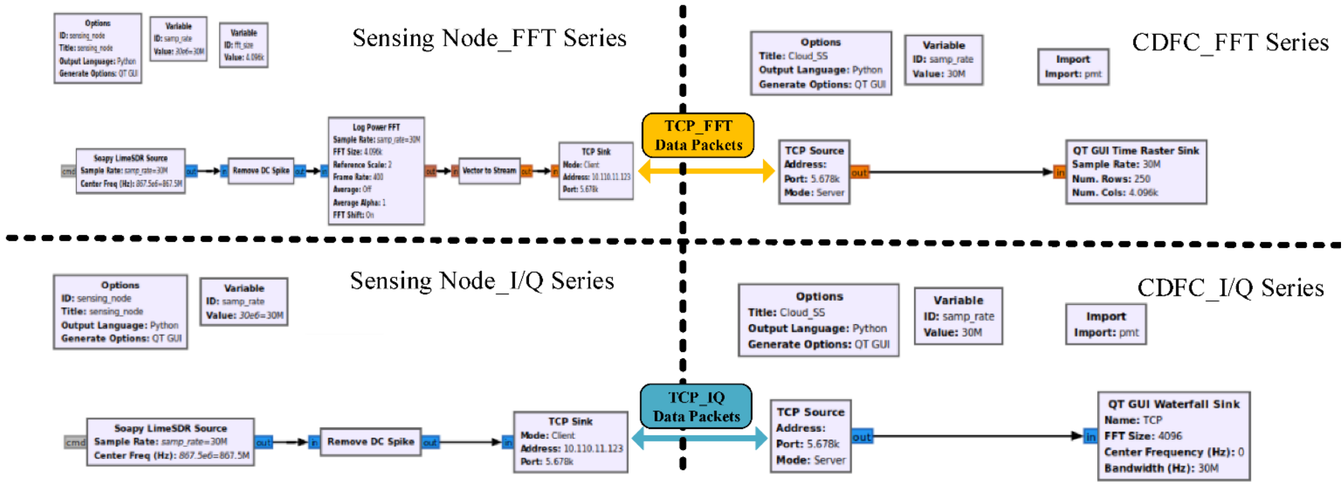

Fig. 2. GNURadio Flowgraphs: Sensing Node and CDFC for I/Q and FFT series data formats

comprises two main components, with the first being CDFC. We used the cloud computing infrastructure of Stefan cel Mare University of Suceava [24] to set up the CDFC, deploying a virtual machine (VM) designed for this purpose. This VM is powered by 8 Intel Xeon Icelake CPU cores and is equipped with 1 TB of RAM and 16 TB of storage space. It runs on Ubuntu version 22.04 and can deliver approximately 200 GFLOPS of computational power.

For the deployment of the sensing nodes, we opted for a Raspberry Pi 3B+, integrated with a Lime SDR [25], and running on PiSDR OS. To facilitate the interfacing with the LimeSDR we utilized GNURadio [26], an open-source application that provides software-based signal processing tools. One of the key features of GNURadio is its support for data transfer from the sensing node to the CDFC using two protocols: TCP and UDP. It includes specific sink and source blocks for these protocols. Despite the common preference for UDP in streaming applications due to its low latency, we chose TCP for our CESS architecture to prioritize spectrum data integrity.

Within this setup, we considered two data formats for transmitting captured radio spectrum data: I/Q coordinates and FFT series. Fig. 2 illustrates the GNURadio flowgraphs for processing and transmitting both data types. I/Q samples have the benefit of requiring no preprocessing before transmission to the TCP sink, which then forwards the data to the CDFC. On the other hand, FFT series, although needing processing, can enhance performance by reducing bandwidth requirements.

To evaluate the efficiency of these data types, we assessed the required computational power measured in GFLOPS and the necessary internet data rate for various radio signal bandwidths, including 500kHz, 1 MHz, 2 MHz, 5 MHz, 10 MHz, 20MHz, and 30 MHz. Furthermore, to comprehensively assess the CESS architecture's performance, we measured the Round-Trip Time (RTT) of the acknowledgment packet across all considered bandwidths and data representation formats. These measurements were conducted using Wireshark [27] at the CDFC level, capturing TCP packets for a 30 s interval of radio spectrum data transmission. We applied a filter for the designated TCP port and utilized the ***tcp.analysis.ack_rtt*** function to record the RTT, thereby obtaining a detailed analysis of the system's efficiency in terms of latencies vs. recorded bandwidth.

IV. Performance Analysis, Results and Discussions

In this section, we present the performance evaluation of the CESS architecture, using the performance metrics previously outlined. Our analysis initially focuses on the computational power required for processing the spectrum data across all recorded bandwidth values, considering both I/Q coordinates and FFT series data formats. To accurately determine the computational demands of each data type across the various bandwidths, we employed the ***perf stat*** tool, part of the ***linux-tools-common*** package. This tool is instrumental in measuring the computational requirements of an isolated process, providing accurate data on the processing power needed for our architecture to function efficiently. Moreover, we extended our analysis to include the internet data rate necessary for each data format at every bandwidth level under consideration. This aspect of the evaluation is crucial, as it directly impacts the efficiency of data transmission from the sensing nodes to the CDFC.

Table 1. Required computational power and Internet Data Rate

Data type	**Recorded Radio Bandwidth (MHz)**	**Measured internet data rate (MB/s)**	**Required computational power (GFLOPS)**
I/Q	0.5	4.00	0.002634
I/Q	1	8.00	0.008003
I/Q	2	16.00	0.015451
I/Q	5	40.00	0.043534
I/Q	10	41.00	0.044638
I/Q	20	43.00	0.068523
I/Q	30	43.00	0.071155
FFT	0.5	2.00	0.007637
FFT	1	4.00	0.017193
FFT	2	6.20	0.029847
FFT	5	6.20	0.051396
FFT	10	6.20	0.105519
FFT	20	6.20	0.203757
FFT	30	6.70	0.300842

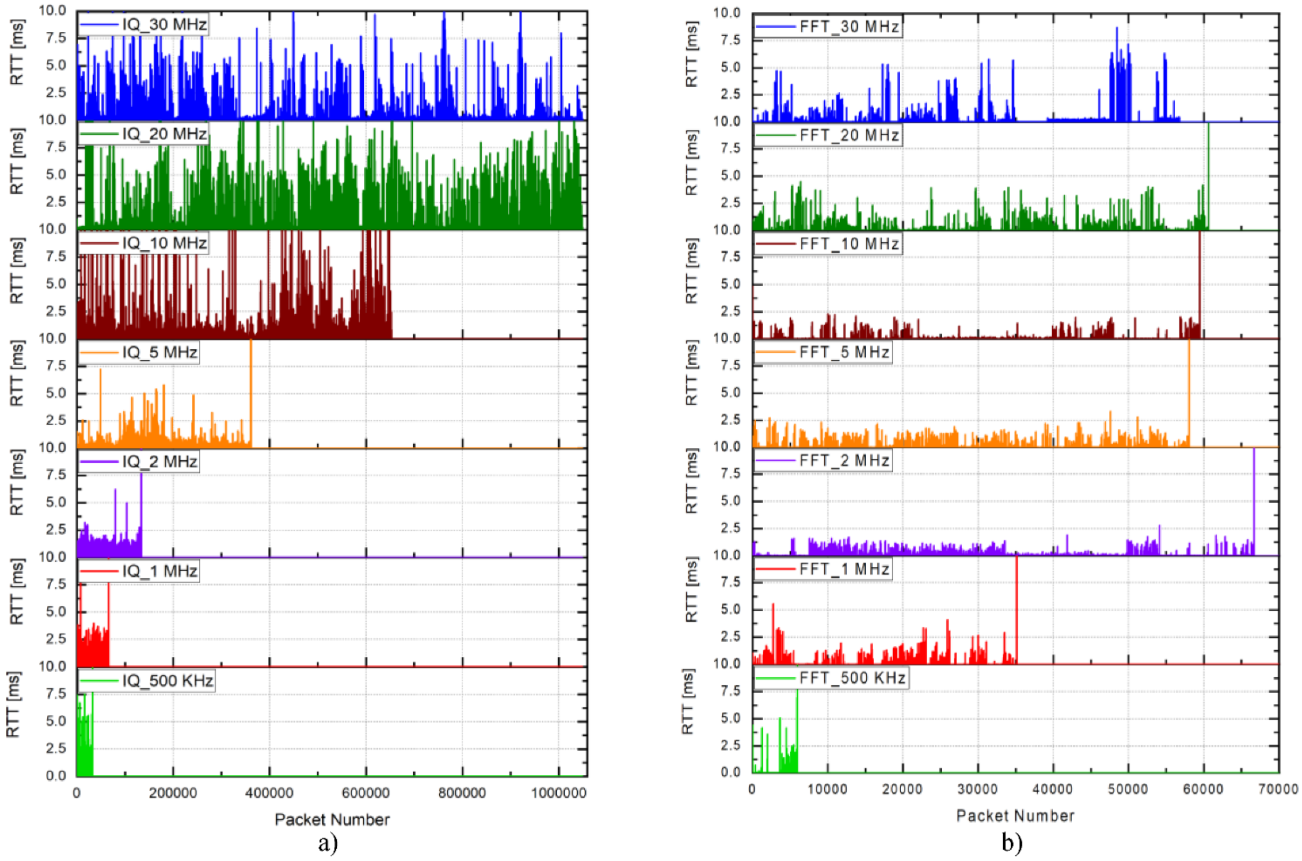

Fig. 3. Round-Trip Times of the TCP ACK packages for the: (a) I/Q data format, (b) FFT data format

The analysis results presented in Table 1 reveals a notable difference in the computational and internet data rate requirements between the I/Q data format and FFT series within the CESS architecture. For the I/Q data format, we observed that the computational power needed is minimal, peaking at just 0.07 GFLOPS. Such low computational demands are well within the capabilities of micro-computers like the Raspberry Pi. However, the I/Q format demands significantly higher internet data rates, which increase linearly with the bandwidth of the recorded radio signal. This increase is attributable to the size of the I/Q data format as each I/Q sample is stored in the ***numpy.complex64*** format, requiring 8 Bytes per sample. In theoretical terms, for bandwidths ranging from 10 MHz to 30 MHz, the internet data rate needed would scale from 80 MB/s to 240 MB/s. Nevertheless, due to hardware limitations of the Raspberry Pi, this data rate is not feasible beyond a 5 MHz bandwidth. On the other hand, converting the I/Q samples into FFT series resolves the high internet data rate requirement, without imposing substantial computational demands. The computational power needed for FFT series ranges from 0.007 GFLOPS at a 500 kHz bandwidth to 0.3 GFLOPS at a 30 MHz bandwidth. Additionally, the internet data rate for FFT series does not exceed 6.7 MB/s, irrespective of the bandwidth. This analysis indicates that FFT series offer a balanced compromise, effectively managing both computational power and internet data rate requirements. This balance is crucial for optimizing the performance and feasibility of deploying the CESS architecture, particularly when considering the limitations of the sensing nodes' hardware and the need for efficient data transmission to the CDFS.

In Fig. 3, we present the round-trip times (RTTs) of acknowledgment (ACK) packets for both the I/Q and FFT data formats, using a stacked plot to highlight differences across the various recorded bandwidths. This visualization method allows for a clearer understanding of how latency varies with each data format and bandwidth. The results indicate that, for both I/Q and FFT data formats, the latencies remain below 10 ms, which is within acceptable limits for near real-time processing requirements. However, there is a notable distinction in performance between the two formats. For the I/Q data format, the average RTT is significantly higher, reaching or exceeding 5 ms for bandwidths above 5 MHz. This leads to cumulative delays that could impact the efficiency of the spectrum sensing process, especially in scenarios requiring rapid data analysis and decision-making. The FFT data format demonstrates more favorable latency characteristics. The average RTT values for FFT are below 2.5 ms, although there are occasional spikes that reach up to 10 ms towards the end of transmissions. These spikes, while noticeable, do not detract from the overall efficiency of using FFT data format for data transmission. Furthermore, it is evident that transmitting the same 30 seconds of recorded spectrum data in the I/Q format requires a significantly greater number of data packets compared to the FFT format. This difference underscores the increased efficiency and reduces the network load achievable with FFT data format.

The comparative analysis of RTT and packet requirements solidifies the advantages of adopting the FFT data format for the CESS architecture. Not only does FFT reduce the computational and data transmission requirements on sensor nodes, but it also ensures lower latencies and fewer data packets are needed for transmission. These benefits collectively enhance the scalability and practicality of the CESS architecture, particularly for deployments leveraging sensor nodes with limited computational capabilities.

V. Conclusions

In this paper, we propose, implement, and assess a Cloud-enabled Spectrum Sensing architecture, evaluating its performance and the design implications for the sensing nodes. Our deployment utilizes a Cloud-based Virtual Machine hosted on the infrastructure of the Stefan cel Mare Cloud Computing Center to serve as the Cloud Data Fusion Center. At the sensing node level, we employed a Raspberry Pi micro-computer combined with a Software-Defined Radio, using the TCP protocol for data transmission in order to ensure data integrity. Two signal representation formats, namely I/Q and FFT were considered and evaluated in our analysis in terms of computational power demands, required internet data rate and latency. Our results show that using the FFT series signal representation format at the sensing node level presents several advantages over using raw I/Q data. Specifically, the FFT format significantly reduces network traffic and ensures lower latency, without substantially increasing computational demands across a wide spectrum of recorded radio bandwidths, obtaining an average latency below 2.5 ms, while only using 6.7 MB/s internet data rate and a computational power of 0.3 GFLOPS for a recorded radio signal bandwidth of 30 MHz.

This approach enhances the operational efficiency of the Cloud-Enabled Spectrum Sensing architectures, demonstrating its potential for scalable deployment in environments constrained by computational and network resources. The results discussed in this paper provide a foundational baseline for developing scalable, cost-effective cloud-based spectrum sensing architectures. This framework facilitates further growth and expansion of WSNs within the IoT ecosystem.

Acknowledgment

This work was supported by the project "Interdisciplinary Cloud and Big Data Center at Ștefan cel Mare University of Suceava", POC/398/1/1, 343/390019, co-founded by the European Union.

References

[1] S. L. Ullo and G. R. Sinha, “Advances in Smart Environment Monitoring Systems Using IoT and Sensors,” Sensors 2020, Vol. 20, Page 3113, vol. 20, no. 11, p. 3113, May 2020, doi: 10.3390/S20113113.

[2] O.-A. Schipor and R.-D. Vatavu, “Towards Interactions with Augmented Reality Systems in Hyper-Connected Cars,” 2019.

[3] O. Gherman, O. Schipor, and B. F. Gheran, “VErGE: A system for collecting voice, eye gaze, gesture, and EEG data for experimental studies,” 2018 14th International Conference on Development and Application Systems, DAS 2018 - Proceedings, pp. 150–155, Jun. 2018, doi: 10.1109/DAAS.2018.8396088.

[4] O. A. Schipor, R. D. Vatavu, and J. Vanderdonckt, “Euphoria: A Scalable, event-driven architecture for designing interactions across heterogeneous devices in smart environments,” Inf Softw Technol, vol. 109, pp. 43–59, May 2019, doi: 10.1016/J.INFSOF.2019.01.006.

[5] O. A. Schipor, L. B. Bilius, and R. D. Vatavu, “WearSkill: Personalized and Interchangeable Input with Wearables for Users with Motor Impairments,” Proceedings of the 19th International Web for All Conference, W4A 2022, Apr. 2022, doi: 10.1145/3493612.3520455.

[6] A. A. Maftei, P. M. Mutescu, V. Popa, A. I. Petrariu, and A. Lavric, “Internet of Things Healthcare Application: A Blockchain and LoRa Approach,” 2021 9th E-Health and Bioengineering Conference, EHB 2021, 2021, doi: 10.1109/EHB52898.2021.9657733.

[7] A. Lavric and A. I. Petrariu, “LoRaWAN communication protocol: The new era of IoT,” 2018 14th International Conference on Development and Application Systems, DAS 2018 - Proceedings, pp. 74–77, Jun. 2018, doi: 10.1109/DAAS.2018.8396074.

[8] A. Lavric, A. I. Petrariu, and V. Popa, “SigFox Communication Protocol: The New Era of IoT?,” 2019 International Conference on Sensing and Instrumentation in IoT Era, ISSI 2019, Aug. 2019, doi: 10.1109/ISSI47111.2019.9043727.

[9] C. Goursaud and Y. Mo, “Random Unslotted Time-Frequency ALOHA: Theory and Application to IoT UNB Networks,” pp. 1–5, 2016, doi: 10.1109/ICT.2016.7500489ï.

[10] T. Gale, T. Šolc, R. A. Moşoi, M. Mohorčič, and C. Fortuna, “Automatic Detection of Wireless Transmissions,” IEEE Access, vol. 8, pp. 24370–24384, 2020, doi: 10.1109/ACCESS.2020.2970840.

[11] P. M. Mutescu, A. Lavric, A. I. Petrariu, and V. Popa, “A Hybrid Deep Learning Spectrum Sensing Architecture for IoT Technologies Classification,” 2023 17th International Conference on Engineering of Modern Electric Systems, EMES 2023, 2023, doi: 10.1109/EMES58375.2023.10171667.

[12] S. Rajendran, W. Meert, V. Lenders, and S. Pollin, “SAIFE: Unsupervised Wireless Spectrum Anomaly Detection with Interpretable Features,” 2018 IEEE International Symposium on Dynamic Spectrum Access Networks, DySPAN 2018, Jan. 2019, doi: 10.1109/DYSPAN.2018.8610471.

[13] S. Peng, S. Sun, and Y. D. Yao, “A Survey of Modulation Classification Using Deep Learning: Signal Representation and Data Preprocessing,” IEEE Trans Neural Netw Learn Syst, vol. 33, no. 12, pp. 7020–7038, Dec. 2022, doi: 10.1109/TNNLS.2021.3085433.

[14] A. Vagollari, V. Schram, W. Wicke, M. Hirschbeck, and W. Gerstacker, “Joint Detection and Classification of RF Signals Using Deep Learning,” IEEE Vehicular Technology Conference, vol. 2021-April, Apr. 2021, doi: 10.1109/VTC2021-SPRING51267.2021.9449073.

[15] F. Zhang, C. Luo, J. Xu, Y. Luo, and F. C. Zheng, “Deep learning based automatic modulation recognition: Models, datasets, and challenges,” Digit Signal Process, vol. 129, p. 103650, Sep. 2022, doi: 10.1016/J.DSP.2022.103650.

[16] T. J. O’Shea, T. Roy, and T. C. Clancy, “Over-the-Air Deep Learning Based Radio Signal Classification,” IEEE Journal on Selected Topics in Signal Processing, vol. 12, no. 1, pp. 168–179, Feb. 2018, doi: 10.1109/JSTSP.2018.2797022.

[17] S. Atapattu, C. Tellambura, and H. Jiang, “Energy Detection for Spectrum Sensing in Cognitive Radio,” 2014, doi: 10.1007/978-1-4939-0494-5.

[18] N. N. Dao, W. Na, A. T. Tran, D. N. Nguyen, and S. Cho, “Energy-Efficient Spectrum Sensing for IoT Devices,” IEEE Syst J, vol. 15, no. 1, pp. 1077–1085, Mar. 2021, doi: 10.1109/JSYST.2020.2986030.

[19] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, vol. 2017-January, pp. 6517–6525, Nov. 2017, doi: 10.1109/CVPR.2017.690.

[20] N. Daldal, Z. Cömert, and K. Polat, “Automatic determination of digital modulation types with different noises using Convolutional Neural Network based on time–frequency information,” Appl Soft Comput, vol. 86, p. 105834, Jan. 2020, doi: 10.1016/J.ASOC.2019.105834.

[21] M. E. Karar, F. Alsunaydi, S. Albusaymi, and S. Alotaibi, “A new mobile application of agricultural pests recognition using deep learning in cloud computing system,” Alexandria Engineering Journal, vol. 60, no. 5, pp. 4423–4432, Oct. 2021, doi: 10.1016/J.AEJ.2021.03.009.

[22] A. M. Ghosh and K. Grolinger, “Edge-Cloud Computing for Internet of Things Data Analytics: Embedding Intelligence in the Edge with Deep Learning,” IEEE Trans Industr Inform, vol. 17, no. 3, pp. 2191–2200, Mar. 2021, doi: 10.1109/TII.2020.3008711.

[23] “GPT-4.” Accessed: Mar. 18, 2024. [Online]. Available: https://openai.com/research/gpt-4

[24] “Interdisciplinary Cloud and Big Data Center at Ștefan cel Mare University of Suceava.” Accessed: Mar. 18, 2024. [Online]. Available: https://cloudusv.ro/

[25] “LimeSDR - Lime Microsystems.” Accessed: Apr. 24, 2023. [Online]. Available: https://limemicro.com/products/boards/limesdr/

[26] “GNU Radio - The Free & Open Source Radio Ecosystem · GNU Radio.” Accessed: Jun. 13, 2022. [Online]. Available: https://www.gnuradio.org/

[27] “Wireshark · Go Deep.” Accessed: Mar. 18, 2024. [Online]. Available: https://www.wireshark.org/