

# Electrosense: Open and Big Spectrum Data

Sreeraj Rajendran, Roberto Calvo-Palomino, Markus Fuchs, Bertold Van den Bergh, Héctor Cordobés, Domenico Giustiniano, Sofie Pollin and Vincent Lenders

While the radio spectrum allocation is well regulated, there is little knowledge about its actual utilization over time and space. This limitation hinders taking effective actions in various applications including cognitive radios, electrosmog monitoring, and law enforcement. The authors introduce Electrosense, an initiative that seeks a more efficient, safe and reliable monitoring of the electromagnetic space by improving the accessibility of spectrum data for the general public.

## ABSTRACT

While radio spectrum allocation is well regulated, there is little knowledge about its actual utilization over time and space. This limitation hinders taking effective actions in various applications including cognitive radios, electrosmog monitoring, and law enforcement. We introduce Electrosense, an initiative that seeks a more efficient, safe and reliable monitoring of the electromagnetic space by improving the accessibility of spectrum data for the general public. A collaborative spectrum monitoring network is designed that monitors the spectrum at large scale with low-cost spectrum sensing nodes. The large set of data is stored and processed in a big data architecture and provided back to the community with an open spectrum data as a service model, that allows users to build diverse and novel applications with different requirements. We illustrate useful usage scenarios of the Electrosense data.

## INTRODUCTION

In recent years, we have seen a tremendous increase in mobile data usage. To meet the data demands, more radio frequency (RF) bands are being used, cells become smaller, data rates increase and more users accommodated. In addition, novel technologies are being proposed that can coexist with legacy technology and modulation types, and short and long range IoT communications increase the variety of physical and medium access protocols connected to the Internet. As a result, RF spectrum use is fragmented, bursty and very diverse. The monitoring of this spectrum is a complex task, as it requires a very dense sampling in time, frequency and space, resulting in a radio spectrum data deluge. Nevertheless, with the increasing spectrum usage complexity, knowledge and understanding of the spectrum usage patterns is becoming more and more critical to ensure continued effective use of this scarce resource.

Spectrum resource monitoring is important for end users, operators, spectrum regulatory bodies and also military applications. Each use case has its own specific needs and challenges. The grand challenge is how to design a cost-effective solution that meets the requirements of all potential end users. Users might be interested in electrosmog or optimization of their indoor WiFi network. Regulatory bodies might be keen on

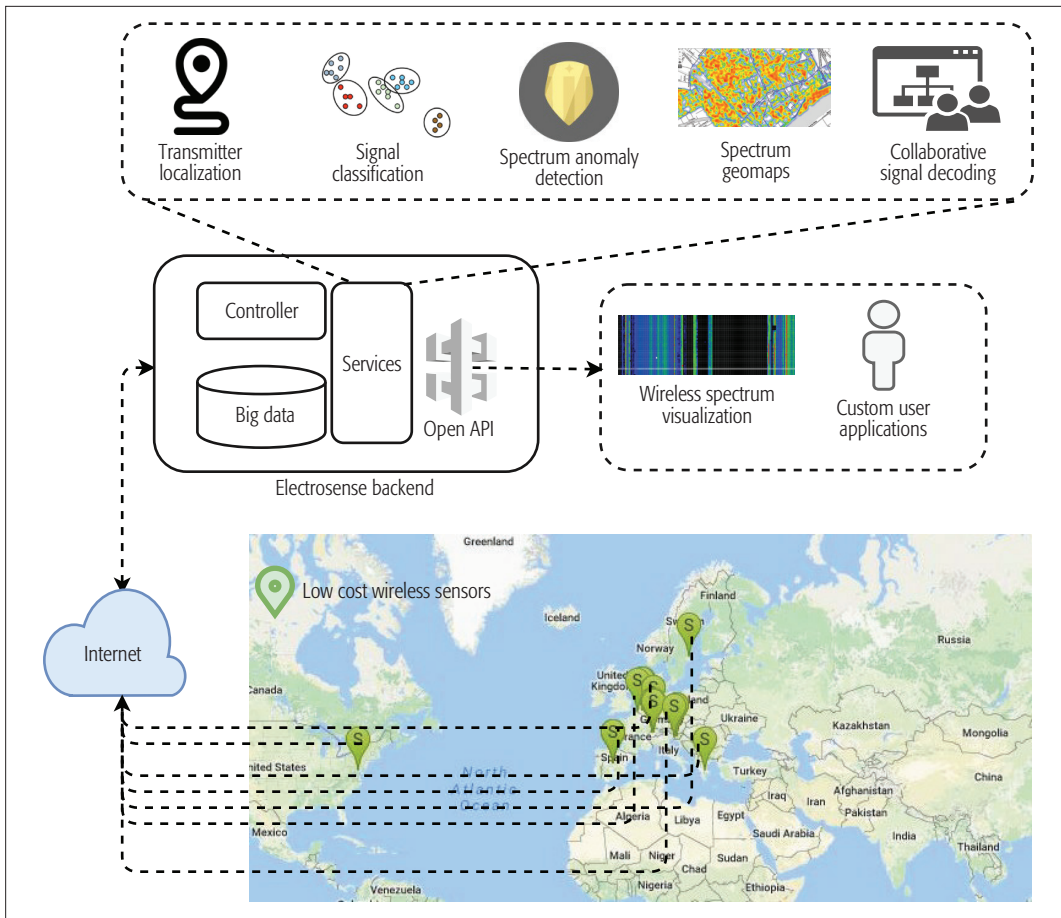
enforcing spectrum regulation. Operators might be concerned about coverage maps over time for optimizing their cell networks or refarming of their frequency bands. Military applications might be the most challenging, requiring the detection and positioning of any signal hidden on purpose. Novel operators might be interested in IoT cases, such as cooperative detection of signals transmitted by low-cost low-power transceivers. This article looks at one key aspect: is it possible to design a spectrum sensing network that meets all the aforementioned requirements and can be deployed quickly by synergistic cooperation between all stakeholders?

Even though a majority of wireless researchers, industries and spectral regulators are keen to develop a worldwide spectrum monitoring infrastructure, albeit several attempts, the research community has not succeeded in deploying one. The multidisciplinary nature of the spectrum monitoring solution is one of the main challenges that prevents the realization of such a system, which in turn requires proper integration of new disruptive technologies. The infrastructure should flexibly address the variability and cost of the used sensors, large spectrum data management, sensor reliability, security and privacy concerns, which can also target a wide variety of the use cases, as mentioned before.

A few spectrum monitoring solutions are proposed in the literature. Some examples include, Microsoft Spectrum Observatory (<http://observatory.microsoftspectrum.com/>), which makes it possible to sense the spectrum using expensive sensors; Google spectrum (<https://www.google.com/get/spectrumdatabase/>) for measurements on TV white-spaces; and the IBM Horizon (<https://bluehorizon.network/documentation/sdr-radio-spectrum-analysis>) project, which proposed a generic decentralized architecture to share IoT data. While Google Spectrum and Blue Horizon fail to cover a large part of the spectrum as they are application-specific deployments, Microsoft observatory on the other hand fails to enable a large scale sensor deployment mainly due to the cost of the sensing stations.

We introduce and design Electrosense (<https://electrosense.org/>). Electrosense follows the crowdsourcing paradigm to collect and analyze spectrum data using low-cost sensors as the main devices for sensing the spectrum. The main goal of this initiative, as shown in Fig. 1, is to sense the whole spectrum in different regions of the world

*Sreeraj Rajendran, Bertold Van den Bergh, and Sofie Pollin are with KU Leuven; Roberto Calvo-Palomino, Héctor Cordobés, and Domenico Giustiniano are with IMDEA Networks Institute; Roberto Calvo-Palomino is also with Universidad Carlos III of Madrid; Markus Fuchs is with SeRo Systems; Vincent Lenders is with armassuisse.*



Electrosense follows the crowdsourcing paradigm to collect and analyze spectrum data using low-cost sensors as the main devices for sensing the spectrum. The main goal of this initiative is to sense the whole spectrum in different regions of the world and provide the processed spectrum data to any user interested in acquiring a deeper knowledge of spectrum usage.

**Figure 1.** High-level overview of the Electrosense network: Low-cost sensors collect spectrum information which are sent to the Electrosense backend. Different algorithms are run on the collected information in the backend and the results of these algorithms are provided to the users as a service through an open API. Users can develop their own applications from the spectrum information retrieved using the API.

and provide the processed spectrum data to any user interested in acquiring a deeper knowledge of spectrum usage, enabling applications such as the ones mentioned above [1, 2].

Worldwide deployments are plausible when the crowdsourcing paradigm is combined with low cost sensors. Electrosense sensors are designed using inexpensive and easily accessible software-defined radio (SDR) front-ends and embedded platforms like Raspberry Pi, to reduce the cost that spectrum data contributors have to bear. A low-cost down-converter design (<https://github.com/electrosense/hardware>) is also provided to extend the range of the low-end SDR platforms from 1.7 GHz to 6 GHz. Even though the aim of the Electrosense project is to use low-cost hardware, high-end SDR devices can also be part of the network. Open source software modules for low-end and high-end sensors are provided by Electrosense for easy setup (<https://github.com/electrosense>).

Crowdsourcing initiatives have been successfully applied in other sensing contexts, for example, distributed sensor networks collecting information about temperature, air quality, pollution or air traffic are now in widespread use. Yet, radio spectrum data collection poses novel challenges because of the sheer volume of information in the spectrum, which is several orders of

magnitude higher than the data collected in typical sensing contexts as mentioned above.

The amount of data produced by a single sensor typically varies between 50 Kb/s and 50 Mb/s, depending on the sensor platform and configuration. Storing spectrum data received from each sensor can swiftly use up the storage space of any server machine, for instance data from 60 sensors for a single month needs storage space of 1 terabyte in 50 Kb/s mode. In order to handle this large volume of data in the backend and extract meaningful information over the entire spectrum, a flexible big data architecture is designed. This spectrum management architecture is responsible for the sensors' control, data storage, and algorithm deployment in the back-end for further processing.

Electrosense also includes a complete framework for identifying, locating and deploying sensors securely through a consistent registration process and remote control framework of the sensors. The sensed spectrum data from different sensors can be retrieved from the electrosense backend using an open API (<https://electrosense.org/open-api-spec.html>) over the Internet. Data privacy concerns are addressed by enabling public, private and restricted data access permissions that restrict user data access with suitable time resolutions. Readily available spectrum aggrega-

In order to make the most out of the data, we introduce an Open Spectrum Data as a Service (OSDaaS) model in which the spectrum data can be used by several applications, each of them with unique requirements. This approach differs from a classic “Infrastructure as a Service” model used in other contexts where different applications cannot run at the same time due to conflicting requirements.

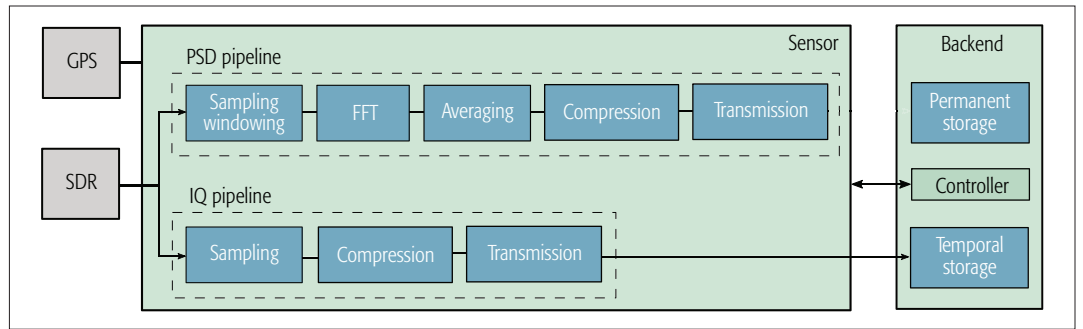


Figure 2. Sensor architecture: sensor software contains two different pipelines for retrieving spectrum information.

tion tools in the back-end help the network users; performing easy spectrum data analysis as an additional incentive. The data access permissions and other applications are detailed below.

The rest of this article is organized as follows. A brief overview of the service model and the open API are presented in the next section. Following that we define Electrosense system design considerations and the final architecture. Then we detail the spectrum data analysis tools and applications. Finally, conclusions and future work are presented.

## SPECTRUM DATA AS A SERVICE

### SERVICE MODEL

In order to make the most out of the data, we introduce an Open Spectrum Data as a Service (OSDaaS) model in which the spectrum data can be used by several applications, each of them with unique requirements. This approach differs from a classic “Infrastructure as a Service” model used in other contexts where different applications cannot run at the same time due to conflicting requirements. The spectrum data inherently changes over time and thus allocating the node to a specific application would reduce the information and the interest of several users interested in the same spectrum data. Therefore, the current framework prevents the users from launching their own measurement campaigns as they only need to access gathered data from one of the two pipelines of the sensor: PSD (power spectral density) and IQ (in-phase and quadrature components of raw signals). In contrast, the flexibility of the OSDaaS model is an incentive to join and use the network and an opportunity for a wide engagement and participation by citizens. In addition, the spectrum data gathered in the past can be re-processed with a new application in mind, or new spectrum data can be compared with older spectrum data. Users can join the Electrosense network easily by adding their own sensor through the web interface (<https://electrosense.org/join.html>). Specifics about the sensor such as antenna details and sensor location can be specified during the registration process. The sensor status and location are readily visible in a global sensor map once it is registered.

### APPLICATION PROGRAMMING INTERFACE (API)

In addition to data processing, it is important to provide the community with easy access to the data. An open API serves this purpose for bulk or streaming data retrieval. The API also

allows access to the algorithms’ output running in the backend, e.g., results of the modulation classification algorithms or the anomaly detector. Electrosense users can retrieve magnitude data along with the sensor details using the API. The API allows two data retrieval modes: aggregated and raw Fast Fourier Transform (FFT) data. Aggregated query type allows the user to request a bulk of data with specified frequency and time resolutions, after applying a predefined aggregation function such as averaging or max-value. With raw requests, it is possible to retrieve magnitude FFT data acquired by a sensor as such without any modifications. Raw FFT data from a sensor can be accessed only by its owner. In addition, as shown in Fig. 2, there is an IQ data pipeline that is enabled on demand for data testing or for algorithms that work on IQ data in the backend. The IQ data from a sensor is only directly available to its owner through the API, but conclusions from algorithms related to IQ data will be made available to all users. Furthermore, real-time spectrum monitoring is possible through a streaming API in the backend. A few examples of data retrieval can be found in the repository (<https://github.com/electrosense/api-examples>).

### SECURITY AND PRIVACY

Wideband spectrum monitoring on a large scale typically raises critical security and privacy concerns. Security concerns include secure data transmissions from the sensor and proper sensor identification. Data privacy concerns in terms of data access restrictions should also be addressed. In addition, capture and storage of IQ data, which can be decoded to reveal content, is not advisable, especially in the military bands. Electrosense’s design addresses sensor identification via a proper registration process. The data from Electrosense sensors are sent over secure TLS channels which guarantees data privacy and integrity and prevents the data being modified by an attacker. The IQ pipeline is only accessible by the Electrosense backend and the data is deleted once the backend algorithms are done processing it. In addition, IQ data is only made available to the users for their own sensors through the API. Furthermore, the highest time and frequency resolution that users can get from other sensors over the API are limited to 60 seconds and 100 kHz, respectively. Electrosense also makes it possible to obfuscate a sensor’s publicly displayed location within a range of a

few kilometers in the sensor map, thus protecting users' locational privacy.

## SYSTEM ARCHITECTURE

Electrosense was designed with a clear vision in mind that translates into a list of design goals as described below. We present the actual architecture chosen to meet these requirements and give a more detailed explanation of its components.

### DESIGN GOALS AND ELECTROSENSE VISION

Electrosense is a crowdsourcing approach to spectrum monitoring where volunteers play a central role. To keep the entry barrier as low as possible, large-scale deployment can only be achieved with *low-cost sensors*. The backend should be flexible, to enable *wide-band monitoring* from sweeping low-end sensors, to *real-time scanning* with more capable SDR platforms or spectrum analyzers. In addition, the backend should be horizontally *scalable*, enabling the network to continuously grow. Scalability introduces even more complexity into the system, which makes it prone to failures. As such, the architecture must also be *fault-tolerant*. Finally, the backend should be able to process a large amount of data from any sensor with *low latency*.

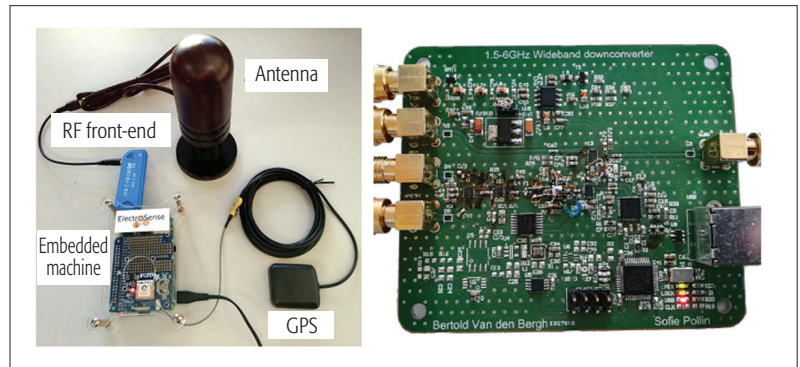
With its main goal to serve researchers as a platform for spectrum analysis, the system should have a *central component to control* the sensors for specific measurement campaigns. Some applications might need to give immediate response while others generate insights by inspecting the data set as a whole. Following these requirements, the system should support *low-latency stream processing* and *large-scale batch analyses* at the same time. In order to address these goals and requirements, the Electrosense architecture consists of three main components:

- The sensors deployed by users collecting spectrum measurements.
- A centralized controller infrastructure to interact with the sensors and to administrate measurement campaigns.
- The backend, which is responsible for collecting data from all the sensors and provides insights by applying algorithms. The following subsections describe these components in detail.

### SENSOR

The sensing nodes used in the Electrosense network consist of small-sized, low-cost, software-defined embedded computing devices connected to a simplistic RF front-end and a general purpose antenna [3], as shown in Fig. 3 (left). The sensors can measure the spectrum ranging from 20 MHz up to 6 GHz using an optional down-converter (Fig. 3, right). The sensors include an optional low-cost GPS device to synchronize the time among them, which in turn helps enable collaborative spectrum scanning and detection algorithms. For instance, in [1] the feasibility to recombine signals collaboratively from different sensors, using GPS as a reference clock, is analyzed in detail.

Two signal pre-processing pipelines are enabled on the sensor, as illustrated in Fig. 2. Each sensor can be configured in order to work with both the PSD or IQ pipeline. When the sensor is



**Figure 3.** [Left] Sensor based on a single-board computer (RPi-2) and RF front-end (RTL-SDR) device responsible for signal acquisition. [Right] First prototype of custom RF converter to enable spectrum scanning in 0-6 GHz range.

configured in the PSD mode, the spectrum sensed by the RF front-end is converted to the frequency domain using a Fast Fourier Transform (FFT). In this mode only the averaged squared magnitude FFTs are sent to the backend. Depending on the averaging and compression factors, the produced data is in the order of 50–100 Kb/s. This pipeline saves bandwidth and storage over the IQ pipeline, and thus contributes to the cost-efficiency of the whole network. In addition, the on-board GPU on the Raspberry Pi is used for computing FFTs, reducing the processing time. The PSD mode aids spectrum data retrieval using a restricted network bandwidth, which enables applications where phase information is not crucial to analyze data in terms of signal power. In addition, wide-band scanning techniques are applied that prioritize frequency bands of bursty activities over bands that are stationary [3] to overcome the hardware-limitations of low-cost radios, e.g., the limited sampling bandwidth of the low-end ADCs (e.g., 2.4 MHz for RTL-SDR).

For applications that require phase and non-averaged information, the sensor implements an IQ pipeline. The IQ mode retrieves raw measurements (IQ samples) from the RF front-end that are then compressed and sent to the backend. This pipeline easily produces up to 50 Mb/s and hence the data is only stored temporarily in the backend. In addition to spectrum measurements in both pipelines, the sensor also provides timing information together with RF-specific values such as antenna gain. The data structure used to transport results over the Internet is flexible and can be extended in the future.

The software that runs in the electrosense nodes is released as open source (<https://github.com/electrosense/es-sensor>). In addition, a GNU Radio based software module, *gr-electrosense* (<https://github.com/electrosense/gr-electrosense-sample>), is provided to allow high-end SDR users to be a part of the Electrosense network.

### CONTROLLER

Apart from continuous wide-band monitoring, it is interesting to have a more detailed look at particular parts of the frequency spectrum. This is accomplished with the help of a command-and-control layer, as depicted in Fig. 4. In order to start and control such in-depth measurements, the controller communicates with sensors

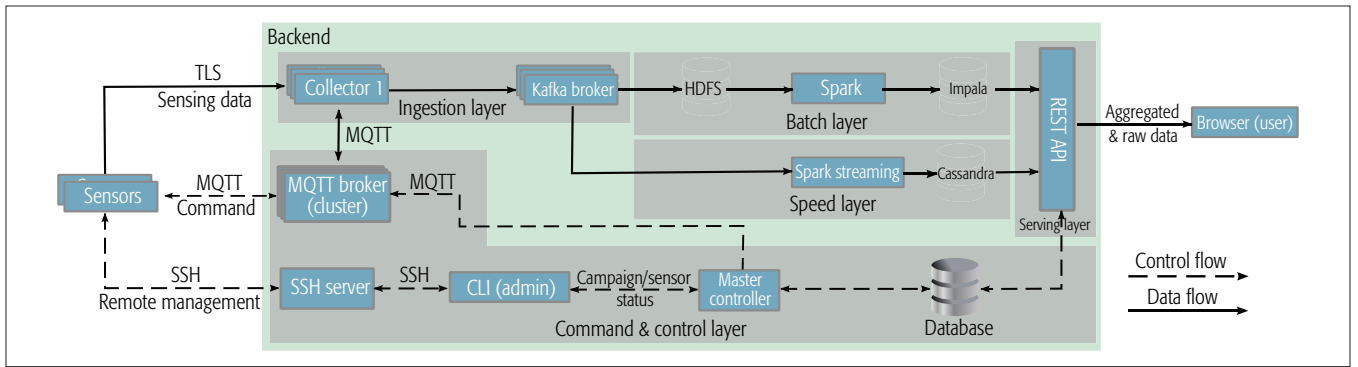


Figure 4. Sensors are managed by the control layer which sets the scanning parameters for a specific campaign. The configured sensors send the spectrum information to the backend, where the data is processed and distributed over the API to the users.

directly and influences their scanning strategy; some examples include the frequency range the sensor should scan, the sensor frequency hopping strategy or the sensor sampling rate.

The core of the controller infrastructure is a Message Queue Telemetry Transport (MQTT) broker system that consists of a set of several independent machines forming a cluster. This ensures scalability and fault-tolerance in case of a hardware failure. MQTT is a publish-subscribe-based messaging protocol and has a wide adoption in IoT applications [4]. The brokers have a permanent connection to the sensors over secure TLS channels as well as to a machine called the *master controller*. The master offers an interface where administrators can start and stop *measurement campaigns*. It is responsible for executing these commands on different sensors using the MQTT connection. Spectrum measurements collected by the sensors are passed to the backend over secure TLS channels, as shown in Fig. 4. The control layer is a part of the Electrosense backend and works in a closed loop with the data processing path for easy response validation.

#### BACKEND

All data collected by the sensors is sent to the backend. In order to fulfill the design goals, in particular scalability, the system is built upon a Lambda architecture [5] with exclusively horizontally scalable system components. The Lambda architecture is a three-layered architecture consisting of *batch*, *speed*, and *serving* layers. An overview of the whole backend architecture can be found in Fig. 4. The batch layer is responsible for executing bulk analysis tasks on large data sets. These tasks mostly involve complex long-running algorithms, whose results might take several minutes to hours to complete. To overcome this gap and, furthermore, provide the capability for real-time monitoring with sub-second delay, there is the need for a speed layer. It is possible to run different or identical applications on each of these layers simultaneously.

In addition, the Electrosense backend has an *ingestion* layer to support high availability. It consists of a distributed message queue that stores multiple replicas of each incoming measurement. Implementation and functionality of each layer is detailed in the subsequent sections.

**Ingestion Layer:** Sensor measurements eventually reach the *collector*, as shown in Fig. 4. The collector inserts received data as messages into

a distributed queuing system. Using a distributed queuing system, we decouple data ingestion from processing in the batch and speed layers, allowing an asynchronous mode of operation. Furthermore, the ingestion layer serves as an intermediate buffer for incoming data, allowing maintenance of the batch and speed layers without data loss.

Apache Kafka [6], a distributed messaging system that features replication, partitioning and data retention, is used as the queuing system. Following the ingestion layer, spectrum data is processed on two different paths, in the batch and speed layers simultaneously.

**Batch Layer:** The batch layer is responsible for storing all incoming raw data as an immutable master dataset and executing long-running workloads. The batch layer facilitates processing of huge amounts of historical data, ensuring no loss of information in case of algorithmic errors. The master dataset is stored in a binary format on a distributed file system. The Hadoop Distributed File System (HDFS) [7] is employed, allowing data to be stored in files that are distributed and replicated over a cluster of servers. Apache Spark [8], which allows parallel computation by evenly distributing workload over several computing nodes, is employed for data processing. For accessing data stored on HDFS, Cloudera Impala [9], a fast and scalable query engine that understands a subset of the SQL language, is deployed. The latency of the batch layer as a whole is in the order of minutes or even hours.

**Speed Layer:** The Electrosense speed layer is based on Spark Streaming. It is an extension of Spark that computes its' results continuously on a parallel cluster over a small window of recent data, with the current default window length set to 5 seconds. Spark Streaming meets the design requirements as the spectrum data is high in volume and algorithms do not demand sub-second latency. As Impala is not suitable for near real-time storage requirements, the speed layer persists its results in another database. We use Apache Cassandra [10], a hybrid between a key-value and column-oriented distributed storage system, for this real-time database, as it represents well the data structure of spectrum data.

**Serving Layer:** As seen before, the batch and speed layers store their results differently. The serving layer takes care of data fusion to provide query results to the user while hiding the complexity of the multi-layer architecture. It offers the

open API, which serves as the endpoint for any user or application to retrieve data. The serving layer handles queries on different views and combines results from the batch and speed layers if necessary. In case results are available in both layers, batch layer results are preferred as they provide more accurate data. In Electrosense, the serving layer is a custom component that offers a RESTful web service over HTTP.

## SPECTRUM DATA PROCESSING AND ANALYSIS

To substantiate the value of the data generated by the Electrosense framework, a few applications are presented in this section. We envision some of these applications to be integrated in Electrosense's batch and speed layers in the backend, but the goal of Electrosense is that users can start implementing their own applications by using the open API.

### LIVE AND HISTORICAL SPECTRUM VISUALIZATION

We have developed a web front-end to enable easy visualization of the spectrum data (<https://electrosense.org/app.html>). The spectrum visualizer allows interactive sensor selection and varying frequency resolution for detailed spectrum viewing. With the visualization tool, users can look at historical spectrum data from the sensors, since the sensor registration. For this purpose, the batch layer pre-computes spectrum data at various temporal and frequency aggregation levels and stores this information in queryable tables for low-latency access over the web. For live spectrum updates with 5 second delays, a streaming display is incorporated into the web interface directly from the speed layer.

### DYNAMIC SPECTRUM ACCESS

Dynamic Spectrum Access (DSA) enables a technology model to manage spectrum availability and needs in terms of frequency, time, geography, quality and cost of the service. Cooperative and distributed spectrum sensing approaches for cognitive radios can solve hidden primary user problems and significantly reduce non-detection probability and false alarms [11]. Advanced collaborative sensing strategies along with detailed sensor density analysis is required to address the complete cognitive cycle. A white space detection study is done to validate the feasibility of the framework in this direction.

The term *white spaces* usually refers to the unused broadcasting frequencies in the TV band (400-800 MHz). The availability of white spaces is highly region dependent, e.g., the white spaces in Europe in the 470-790 MHz band is found to be less than in the U.S. [12]. Using the Electrosense infrastructure and the sensors configured in the PSD pipeline, it is feasible to detect the white spaces in a certain region. Figure 5 shows the spectrum occupancy in the broadcasting TV band (400-800 MHz) of two different cities in Europe with a frequency resolution of 1 MHz and 60 second time resolution. Similar analysis can be easily performed by retrieving the aggregated data from the Electrosense backend through the API.

### SPECTRUM COP

Spectrum enforcement is challenging mainly due to the effort involved in analyzing, detecting and locating anomalies occurring in the spectrum. Even with a large scale deployment of

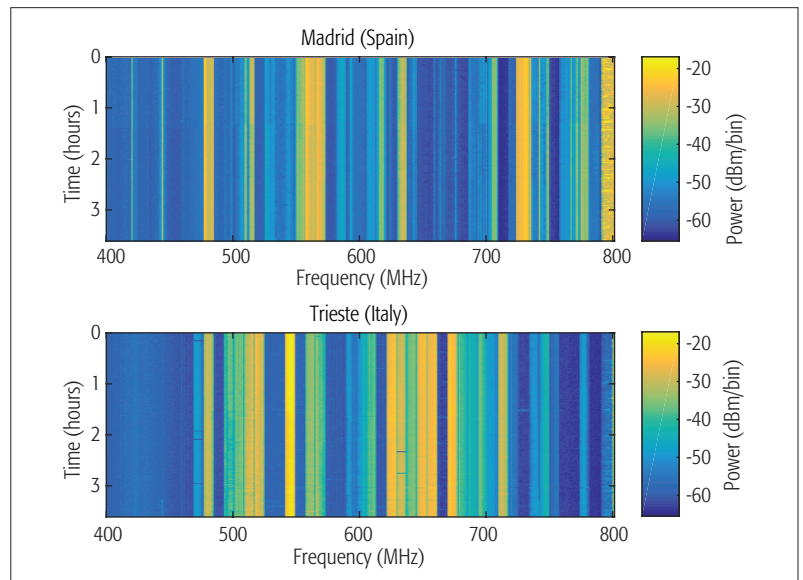


Figure 5. TV band occupancy in Madrid and Trieste.

spectrum scanners, the anomaly detection process is demanding. Automated systems to detect pirate FM stations, fake GSM towers, unauthorized transmissions hindering normal functioning of meteorological radars, and even transmissions at higher power levels than the permitted levels of operation is fundamental for effective enforcement. A good anomaly detector should first detect whether a particular transmission type is allowed in the frequency of interest. Moreover, it should fingerprint normal spectrum transmissions by analyzing the temporal and spatial features of the transmitters in terms of power levels and spectral occupancy. The former can be achieved to some extent using a modulation classifier, and the latter by learning spectral occupancy distributions over time. A drastic variation of any of the aforementioned features can be classified as an anomaly.

To validate modulation classification using Electrosense, a time domain deep learning modulation classifier is used in the backend which works effectively with gr-electrosense using the IQ pipeline [13, 14]. The deep learning model takes IQ samples as input giving out the probability of the data belonging to a particular modulation class. The model is trained to learn from the modulation schemes' time domain amplitude and phase information, without requiring expert features, such as higher order cyclic moments. Analyses show that the proposed model yields an average classification accuracy of over 90 percent at varying SNR conditions ranging from 0 dB to 20 dB, independent of channel characteristics. In the future a computationally efficient binarized version of the model will be deployed on low-cost sensors, reducing the data overhead of sending IQ samples to the backend [14].

### LOCALIZATION

Transmitter localization is a promising application used for generating both automated transmitter maps and transmitter fingerprinting. Research on received signal strength indicator (RSSI)-based indoor and outdoor localization has gained interest in recent years [15]. However, the accuracy of

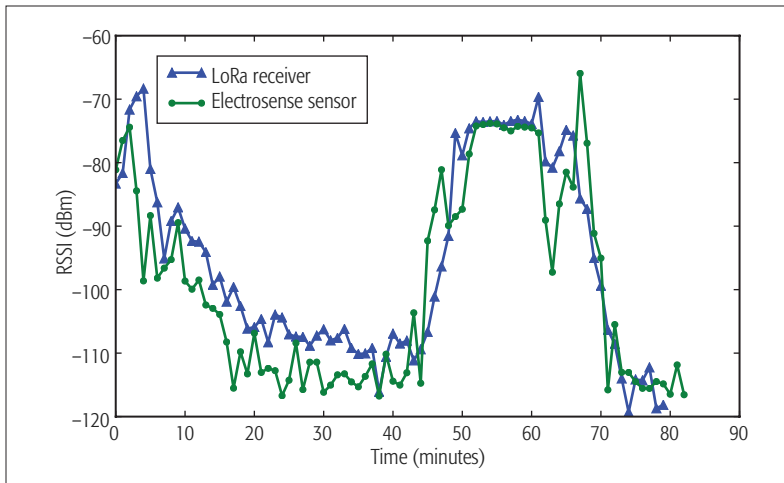


Figure 6. RSSI measured using a LoRa and Electrosense sensor.

the RSSI information drawn from the Electrosense sensors must be verified to validate localization. Such a sensor is calibrated and then the system gain, consisting of the antenna gain, cable losses and front-end gain, is measured for absolute power measurements. RSSI measurements are made in the 435 MHz ISM band using standard LoRa sensors. A splitter is employed to ensure an equal antenna signal reaches both a LoRa receiver and a low-cost Electrosense sensor. A mobile LoRa transmitter is configured to send packets every three second over a period of 80 minutes. Their GPS logged and time-stamped locations are sent to the receiver. The collected RSS measurements from both the LoRa and Electrosense sensors are depicted in Fig. 6. This plot shows that accurate RSSI information can be obtained from properly calibrated Electrosense sensors. The RSSI dips in the sensor data are attributed to the lengthy frequency scan of the sensor, which is over 40 seconds, causing it to miss some packets. Existing algorithms presented in the literature [15] can be deployed using the sensed spectrum data retrieved through the open API or implemented directly in the backend.

## CONCLUSION AND FUTURE WORK

We have introduced Electrosense, a global initiative that seeks a more efficient, safe and reliable use of the electromagnetic space by improving the accessibility of spectrum data for the general public. Electrosense as a crowdsourced network provides solutions to the problems associated with large scale spectrum monitoring and the resulting data deluge. The network users receive incentives in terms of free data storage, readily available applications and an open API.

As the first version of the Electrosense network is fully functional, the next step will be in the direction of enabling usable applications in the backend. The feasibility of various applications was already presented above. Research will be continued on the application space, to develop new algorithms for spectrum prediction and anomaly detection. We also plan to implement temporal and spatial interpolation in the backend to produce spectrum geomaps. In addition, in the future the PSD pipeline in the sensor will be enhanced with advanced spectrum estimation techniques

such as Welch's method. Furthermore, algorithms to extract useful analytics from the spectrum data to overcome data storage constraints will also be explored in the future. Detecting fake or forged data from malicious users, by enabling temporal spectrum comparisons among co-located sensors and user sensor ranking, is another active area for future research. Extensive studies will be done to also understand how complex algorithms can be disintegrated for large scale deployment on these cheap sensors. Electrosense can be the first step to democratize access to the spectrum data to everyone. The age of spectrum data democratization has arrived and it could help increase the transparency and knowledge about spectrum usage.

## REFERENCES

- [1] R. Calvo-Palomino et al., "Crowdsourcing Spectrum Data Decoding," *Proc. IEEE Int'l. Conf. Computer Communications (INFOCOM)*, May 2017, pp. 1–4.
- [2] A. Nika et al., "Empirical Validation of Commodity Spectrum Monitoring," *Proc. SenSys*, 2016, pp. 96–108.
- [3] D. Pfammatter, D. Giustiniano, and V. Lenders, "A Software-Defined Sensor Architecture for Large-Scale Wideband Spectrum Monitoring," *Proc. 14th Int'l. Conf. Information Processing in Sensor Networks*, ser. IPSN '15, 2015, pp. 71–82.
- [4] A. Al-Fuqaha et al., "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surveys & Tutorials*, vol. 17, no. 4, 2015, pp. 2347–76.
- [5] N. Marz and J. Warren, *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*, Manning Publications Co., April 2015.
- [6] J. Kreps et al., "Kafka: A Distributed Messaging System for Log Processing," *Proc. NetDB*, 2011, pp. 1–7.
- [7] K. Shvachko et al., "The Hadoop Distributed File System," *Proc. 2010 IEEE 26th Symposium Mass Storage Systems Technologies (MSST)*, IEEE, 2010, pp. 1–10.
- [8] M. Zaharia et al., "Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing," *Proc. 9th USENIX Conf. Networked Systems Design and Implementation*, USENIX Association, 2012, pp. 2–2.
- [9] M. Kornacker et al., "Impala: A Modern, Open-Source SQL Engine for Hadoop," *Proc. 7th Biennial Conf. Innovative Data Systems Research*, 2015.
- [10] A. Lakshman and P. Malik, "Cassandra: A Decentralized Structured Storage System," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 2, 2010, pp. 35–40.
- [11] T. Yucek and H. Arslan, "A Survey of Spectrum Sensing Algorithms for Cognitive Radio Applications," *IEEE Commun. Surveys & Tutorials*, vol. 11, no. 1, pp. 116–130, 2009.
- [12] J. van de Beek et al., "TV White Space in Europe," *IEEE Trans. Mobile Computing*, vol. 11, no. 2, Feb. 2012, pp. 178–88.
- [13] N. E. West and T. O'Shea, "Deep Architectures for Modulation Recognition," *Proc. 2017 IEEE Int'l. Symposium Dynamic Spectrum Access Networks (DySPAN)*, March 2017, pp. 1–6.
- [14] S. Rajendran et al., "Distributed Deep Learning Models for Wireless Signal Classification with Low-Cost Spectrum Sensors," arXiv preprint arXiv:1707.08908, 2017.
- [15] A. Zanella, "Best Practice in RSS Measurements and Ranging," *IEEE Commun. Surveys & Tutorials*, vol. 18, no. 4, Fourth Quarter 2016, pp. 2662–86.

## BIOGRAPHIES

SREERAJ RAJENDRAN received his Masters degree in communication and signal processing from the Indian Institute of Technology, Bombay, in 2013. He is currently pursuing the Ph.D. degree in the Department of Electrical Engineering, KU Leuven, Belgium. Before joining KU Leuven, he worked as a senior design engineer on the baseband team of Cadence and as an ASIC verification engineer at Wipro Technologies. His main research interests include machine learning algorithms for wireless and low power wireless sensor networks.

ROBERTO CALVO-PALOMINO is a Ph.D. student at IMDEA Networks Institute, associated with the Universidad Carlos III of Madrid (UC3M). He received his M.Sc. degree in telematic engineering from UC3M in 2014 and his M.Sc. degree in telematics systems and computer science from the Universidad Rey Juan Carlos in 2014. His main research is focused on building a smart

crowdsourcing platform for monitoring and decoding wideband spectrum collaboratively, with the main view of democratizing access and knowledge of the radio signals.

MARKUS FUCHS is a data engineer at SeRo Systems, Germany. His work is research-oriented and focuses on scalable data processing systems. He is the lead developer of the OpenSky Network and responsible for the ElectroSense backend. Before joining SeRo Systems, he received his M.Sc. in computer science from the University of Kaiserslautern in 2014.

BERTOLD VAN DEN BERGH is a Ph.D. student in the KU Leuven Electrical Engineering Department. His research is focused on reliable communication systems for UAVs and other applications. Before starting his Ph.D. he obtained his masters degree in microelectronics at KU Leuven in 2013 and did an internship at Flanders Make on wireless localization and tracking of mechatronic systems. He has also worked at ETH Zurich on the Swarmix project.

HÉCTOR CORDOBÉS obtained his M.Sc. in telecommunications in 2003, and an M.Sc. in multimedia and communications research in 2014, both from the Universidad Carlos III. Starting in 2003 we worked at Motorola as a system engineer and developer on international projects such as the Motorola Soft Switch IMS solution, and the Telefonica/Kreatel IPTV platforms. In 2013 he joined IMDEA Networks as a research engineer in machine learning, IoT, and indoor wireless localization projects.

DOMENICO GIUSTINIANO is a research associate professor at IMDEA Networks Institute and leader of the Pervasive Wire-

less Systems Group. He was formerly a senior researcher and lecturer at ETH Zurich and a post-doctoral researcher at Disney Research Zurich and at Telefonica Research Barcelona. He holds a Ph.D. from the University of Rome Tor Vergata (2008). He devotes most of his current research to visible light communication, mobile indoor localization, and collaborative spectrum sensing systems. He is an IEEE senior member, author of more than 70 international papers, leader of the OpenVLC project and co-founder of the non-profit Electrosense Association.

SOFIE POLLIN obtained her Ph.D. degree at KU Leuven with honors in 2006. From 2006 to 2008 she continued her research on wireless communication, energy-efficient networks, cross-layer design, coexistence and cognitive radio at UC Berkeley. In November 2008 she returned to IMEC to become a principal scientist on the green radio team. Since 2012, she has been a tenure track assistant professor in the Electrical Engineering Department at KU Leuven. Her research centers around networked systems that require networks that are ever more dense, heterogeneous, battery powered and spectrum constrained. She is a BAEF and Marie Curie fellow, and an IEEE senior member.

VINCENT LENDERS is a research director at Armasuisse, where he leads the cyber and information sciences research of the Swiss Federal Department of Defense. He received the M.Sc. and Ph.D. degrees in electrical engineering from ETH Zurich. He was a postdoctoral research fellow at Princeton University. He is the cofounder and on the board of the OpenSky Network and Electrosense Associations. His current research interests are in the fields of cyber security, information management, big data, and crowdsourcing.