IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 10, 2020, accepted January 26, 2020, date of publication January 31, 2020, date of current version February 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2970840

Automatic Detection of Wireless Transmissions

TIMOTEJ GALE[1,2], TOMAŽ ŠOLC[1,3], RAREŞ-ANDREI MOŞOI[1,4], MIHAEL MOHORČIČ[1,3], (Senior Member, IEEE), AND CAROLINA FORTUNA[1,2]

[1]Department of Communication Systems, Jožef Stefan Institute, 1000 Ljubljana, Slovenia
[2]ComSensus LLC, 1233 Dob, Slovenia
[3]Jožef Stefan International Postgraduate School, 1000 Ljubljana, Slovenia
[4]Faculty of Electronics, Telecommunications and Information Technology, Technical University of Cluj-Napoca, 400027 Cluj-Napoca, Romania

Corresponding author: Timotej Gale (timotej.gale@ijs.si)

This work was supported in part by the Slovenian Research Agency under Grant P2-0016 and Grant J2-9232, in part by the European Commission through the H2020 WiSHFUL under Grant 645 274, in part by the NRG5 under Grant 762 013, and in part by the eWINE Project under Grant 688 116.

ABSTRACT The current understanding of activity in the wireless spectrum is limited to mostly punctual studies of aggregated energy values. However, there is a need and increasing technological means for a better understanding of spectrum usage by automatically detecting and recognizing wireless transmissions in an unlicensed or shared frequency band. In this paper we propose, implement and evaluate a framework for automatic detection of wireless transmissions. Our framework includes a manual component as our assessment suggests manual labor has a paramount impact on tuning and maintaining good performance of an automatic transmission detection system. However, a considerable problem in this aspect is represented by the disagreement amongst human annotations which is a universally recognized issue. To this end, we discuss and evaluate challenges in generating labeled datasets that can then be used as ground truth for evaluating and possibly training automatic transmission detection systems. We also propose two methods for automatic transmission detection that are not based on machine learning and therefore do not need training data and evaluate their performance against each other and manually labeled data. Our results show that generating human-labeled ground truth data is an expensive and imperfect process. Humans on average require 90 minutes to label 56 minutes of unlicensed European narrowband spectrum. The experts that generate the ground truth sometimes only agree on as little as 40.18% of the labeled cases.

INDEX TERMS Automatic transmission detection, wireless networks, machine vision, labeling, annotation reliability, crowdsourcing, radio spectrum measurements, radio spectrum monitoring, radio spectrum management.

I. INTRODUCTION

The increased penetration of data-driven knowledge technologies, which are complementary to the existing analytical approaches, is already changing the modern information society. For instance, in wireless networks, devices were foreseen to use the *information* provided by the spectrum sensing algorithms for dynamic spectrum access [1]. Spectrum sensing algorithms and low-cost hardware enabled conducting long term spectrum usage studies around the world [2]. Such studies generated additional knowledge, on a larger scale than previously possible. More recently, broadband multi-GHz real-time spectrum analytics enables *fast* generation of information by guiding the sensing devices [3]. Furthermore, real-time wideband spectrum sensing systems able to monitor a larger portion of the spectrum are being proposed [4].

In spite of increased spectrum sensing capabilities, existing long term wider area spectrum studies are based on averaged energy levels from the spectrum [5]–[7]. However, extracting more in-depth knowledge about how the spectrum is used and what kind of signals are present in the air may have a significant impact on the technology and policy design. The need for such knowledge is perhaps best illustrated by SigIdWiki,[1] a Wikipedia-like encyclopedia for signals where known and unknown signals are posted. The effort within

The associate editor coordinating the review of this manuscript and approving it for publication was Ding Xu.

[1]https://www.sigidwiki.com/wiki/Signal_Identification_Guide

SigIdWiki is largely manual: collecting spectrum sensing data using various low-cost devices, human analysis of the data, computer-aided human detection and recognition of transmissions, and finally knowledge generation and publication in a wiki structure.

To automate a large part of the manual efforts that go into systems such as SigIdWiki and thus evolve existing spectrum measurement systems that are not optimized to detect specific transmissions [2], several options exist. One way is to automatically detect the signal using the large body of existing algorithms [8], then filter the signal by time-frequency filtering and recognize the signal type. Another way is to use machine learning by employing a separate machine learning model for each of the two steps, however, if the end goal is to perform recognition, the machine learning problem formulation is able to combine the detection step with the recognition step provided suitable labeled data are available. For good performance, a detection and recognition system likely needs to rely on recent developments from artificial intelligence, particularly deep learning for computer vision [9]. Therefore, as a pre-requisite, sufficient good quality training and evaluation data have to be available. Spectrum data collection can be now scaled in time and space using crowdsourcing infrastructure such as proposed in [10]. However, the research community has yet to consider generating and curating high quality real-world, non-simulated datasets. As shown in other scientific communities [11], [12], there are a number of challenges associated with this process. The quality of the labeled data does not only affect the performance of the learning algorithms they train [13], but, together with the evaluation metrics, also the performance evaluation of the final system. Therefore, even the performance evaluation of unsupervised machine learning systems, such as proposed in [14], depends on the labels of the *ground truth* - i.e. on the convention used to discriminate anomaly from normality.

In this paper, we propose a framework that enables the design and development of automatic detection of wireless transmissions. The proposed framework includes a manual processing component as well as the inherent automatic processing component. We then implement and evaluate the proposed framework on transmission detection from power spectral density (PSD) data. We show that 1) generating human-labeled data is an expensive and imperfect process but necessary as 2) the performance of an automatic detection system depends on the available ground truth data. In this respect, our contributions are as follows:

- We propose and evaluate a manual PSD spectrum labeling approach.
- We propose and evaluate two automatic transmission detection techniques that do not require training data, therefore are not based on machine learning, against the manually generated data.
- We adapt a set of existing evaluation metrics to make them relevant for studying the quality of manually labeled data as well as the performance of the automatic detection.
- We make all the datasets generated for this work publicly available. We also publish the source code for the labeling and machine vision tools as open-source software under the terms of the GNU General Public License v3.0.

The paper is organized as follows. Section II surveys the related work, Section III describes the proposed framework for automatic detection of wireless transmissions and Section IV provides the design and implementation details of the processing pipelines for transmission detection. Section V introduces several performance metrics and discusses performance aspects of the manual and automatic detection processes before Section VI concludes the paper.

II. RELATED WORK

A very comprehensive and critical overview on spectrum occupancy sensing is provided in [2]. The authors show that existing spectrum usage studies generated additional knowledge, on a larger scale than previously possible, however still not sufficient to draw strong conclusions on the topic. They also propose a methodology on how to perform spectrum occupancy analysis for improving spectrum management. A system designed using the framework proposed in this paper can be used to realize all five phases of the methodology proposed in [2].

Various big data spectrum sensing and management systems such as the Microsoft Spectrum Observatory,[2] the Google spectrum[3] for measurements on TV white-spaces, the IBM Horizon[4] project, and, more recently, RadioHound [15] and Electrosense[5] have been proposed over time. Their aim is to enable collecting and analyzing spectrum usage. For instance, the Electrosense architecture [10] is designed for crowdsourcing spectrum sensing and processing large volumes of batch and streaming data. Additionally, efforts addressing punctual problems, such as compressing spectrum scanning to reduce the amount of data are considered in [16].

The labeling method and tool proposed in this paper can extend existing data acquisition infrastructures, eventually enabling the generation of crowdsourcing based wireless labeled datasets. In this endeavor, existing findings with low-cost crowdsourced dataset generation efforts could be considered. For instance, [11] investigated the quality of labeling natural language using platforms such as Mechanical Turk. Similarly as we do in this paper for manually labeled wireless PSD data, [12] investigated how well different humans agree on labels for images. As shown in [13], the quality of the labels in the training data impacts the performance of the machine learning models, therefore generating good quality training data should be a priority as it is a pre-requisite for building an automatic transmission detection system. An anomaly detection system such as the

[2]http://observatory.microsoftspectrum.com
[3]https://www.google.com/get/spectrumdatabase
[4]https://bluehorizon.network/documentation/sdr-radio-spectrum-analysis
[5]https://electrosense.org

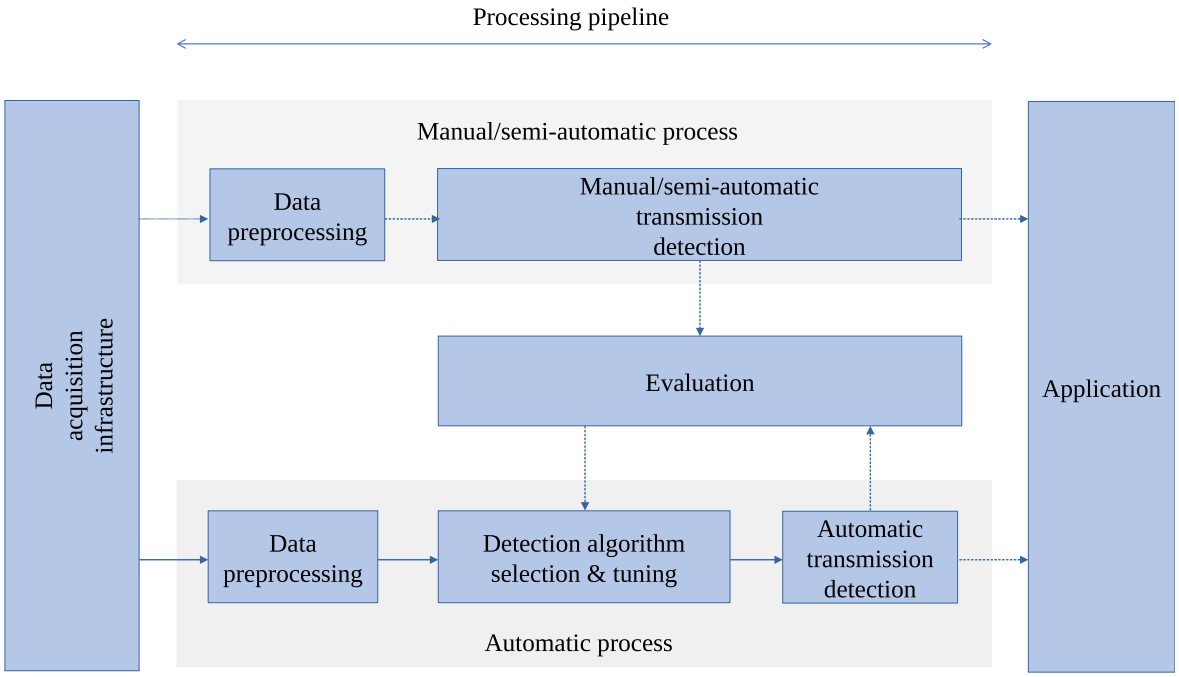


FIGURE 1. Wireless transmission detection framework.

one proposed in [14] could be used to improve the efficiency of the human labeler by selecting transmissions that seem anomalous or have not been seen before.

The generated labeled datasets by the proposed labeling methods and techniques could subsequently be used as training datasets for deep learning models or input to modulation/signal classifiers [17], enabling additional knowledge extraction.

The evaluation metrics adapted in this paper for evaluating the agreement between human-labeled data as well as for evaluating the performance of the systems are similar to the ones used in machine vision communities [9], however they better focus on the needs of the wireless community.

III. AUTOMATIC TRANSMISSION DETECTION FRAMEWORK

In this section, we propose a framework that can be used for automatic detection of transmissions in wireless spectrum. This framework is generic and can be realized in various ways, using fully automatic, manual or mixed rule-based detection systems, supervised classification systems as well as unsupervised systems and is depicted in Fig. 1. It starts with the infrastructure that senses the spectrum, continues with a processing pipeline and ends with an application that utilizes the information on the detected transmissions. In the following, we briefly describe each component of the framework whereas in the rest of the paper we showcase a possible realization and discuss design trade-offs, alternatives and the performance.

A. DATA ACQUISITION INFRASTRUCTURE

The data acquisition infrastructure represents the entry point of the proposed framework. It is responsible for collecting representative data that can be used by the processing pipeline to produce information that the application needs. For instance, the infrastructure could collect I/Q samples for a subsequent modulation and coding classification [18], [19] or PSD data for transmission detection/recognition as found on SigIdWiki and further discussed in this paper. Such infrastructure could be custom made with software-defined radios such as USRPs or signal analyzers, crowdsourced such as in Electrosense [10] or proprietary such as Microsoft Spectrum Observatory or Google spectrum (see Section II). The infrastructure can provide streaming data [20] or enable batch processing by querying traditional relational databases.

Large publicly available datasets for wireless networks are becoming available with the increase of data-driven research initiatives and efforts to democratize spectrum data such as Electrosense. When recording spectrum data suitable for designing automatic transmission detection systems, the relevant tunings of the sensing equipment and possibly environmental features have to be recorded in the meta-data of the actual traces. Documenting relevant meta-data is necessary for gaining an insight into the recording process and understanding the related parameters. Acquiring and publishing a dataset according to best practices is also suitable for better indexing and finding by dedicated dataset search engines, such as Google Dataset Search.[6]

B. PROCESSING PIPELINE

The processing pipeline refers to all the components that are needed to detect transmissions in the data provided by the acquisition infrastructure. Depending on the design, the

[6]https://toolbox.google.com/datasetsearch

transmission detection system may include one or both of the two processing pipelines illustrated in Fig. 1. It may include the manual process in which humans clean and detect transmissions manually using a visual tool or some scripts. This is the current state of the art in the area, and the process is required before any automation is designed and implemented. For designing a good transmission detection system, in addition to having controlled transmissions (i.e. simulated/emulated data), manual data labeling is necessary at least for the ground truth used to evaluate the performance of the automated process [11]–[13].

The pipeline can also be completely automated by employing a set of techniques such as knowledge rules or machine learning algorithms. Automated pipelines, even though designed to detect as wide range of transmissions as possible, will need periodic tuning and updating that is inherently manual. Considering the state of the art in the area, it is not clear what are the design trade-offs of such pipelines and what their relative performance is. However, the accuracy of automated systems can always be improved by adding additional, better quality (labeled) data [13]. Thus, a pipeline that has built-in improvement mechanisms is probably a good way to ensure durability and scalability of an automatic transmission detection system. The automatic process would detect transmissions that it has high confidence in, while (selected) low confidence transmissions [14] would be sent to a semi-automatic or manual process for subsequent characterization (i.e. labeling). The system would then use the resulting labeled data to improve its model of existing transmissions. Depending on the design and implementation of the system, approaches such as manual model update, active learning or online learning could be used. As a result, the performance of the system for automatic detection can be evaluated at design time and it can be improved during run time.

C. APPLICATION

The application refers to the way the detected transmissions are subsequently used. These applications can be dedicated to human or machine end-users. For instance, a signal encyclopedia such as SigIdWiki or a tool summarizing the activity in the spectrum for the regulators would fall under the first category whereas a dynamic radio access system that relies on the detected transmissions would fall under the second category. From the implementation perspective, they could be classified into non-real-time applications that rely on a query-response model for statistical reports or real-time applications that imply a streaming approach useful for agile or dynamic spectrum and network management scenarios. Fig. 2 presents a real-time transmission recognition application that relies on transmission detection.

IV. DESIGN AND IMPLEMENTATION OF THE PROCESSING PIPELINES

In this section, we design and implement three processing pipelines for transmission detection to illustrate the proposed framework. The goal is to detect transmissions in wireless radio spectrum using data as it comes from available off-the-shelf or advanced spectrum sensing devices [4], [21] in the form of PSD values. We do not assume any prior knowledge of how these transmissions might look like.

As we aim to detect wireless transmissions on PSD data, our data resembles the form of a two-dimensional matrix, where one dimension represents time, the other frequency and measured power is color coded. When visualized, the matrix forms the well-known waterfall plot of the spectrum as depicted in Fig. 3. Brighter colors correspond to higher energy levels that represent transmissions, whereas darker colors stand for lower energy levels that represent noise or weak transmissions.

The PSD matrix P, denoted as

$$P = p_{t,f} \in \mathbb{R}^{m \times n}, \tag{1}$$

contains power levels $p_{t,f}$ (we also refer to these as spectrum points), where t is a discrete time point determined by the sampling frequency and f corresponds to each frequency bin (i.e. the smallest resolution in frequency domain obtained from FFT). Formally, given a set of transmissions $X = \{x_1, \ldots, x_n\}$, each transmission x_n is characterized by a rectangle bounding box described as

$$x_n = (t_{start}, t_{stop}, f_{start}, f_{stop})_n \tag{2}$$

where $t_{stop} > t_{start}$ and $f_{stop} > f_{start}$. A rectangle is used as a boundary because it characterizes the behavior of transmissions in PSD data well and is also frequently used in the machine vision community for detecting faces and objects in images [9].

A. PIPELINE FOR MANUAL TRANSMISSION DETECTION

As discussed in Section III, the manual process of detecting transmissions is useful for understanding how to design an automated system and subsequently evaluate the automated system. As spectrum monitoring data are generated at high speeds and in high volume, a human is able to manually detect only a small number of transmissions. Therefore, to maximize the value of manual labeling, the pipeline has to be designed in such a way to:

- Enable fast visual perception of transmissions in the spectrogram. The spectrogram rendering should display transmissions and hide noise, it should use a color scheme that is suitable for human perception so that the transmissions immediately stand out.
- Enable fast bounding box creation. The human user should be able to quickly draw accurate rectangles around transmissions from the spectrograms and associate time/frequency values with the transmission inside the bounding box.
- Enable labeling of relevant transmissions. The spectrograms rendered for labeling purposes should contain transmissions that vary in time, space and type to achieve the highest labeling coverage possible with minimal effort and little repetition.

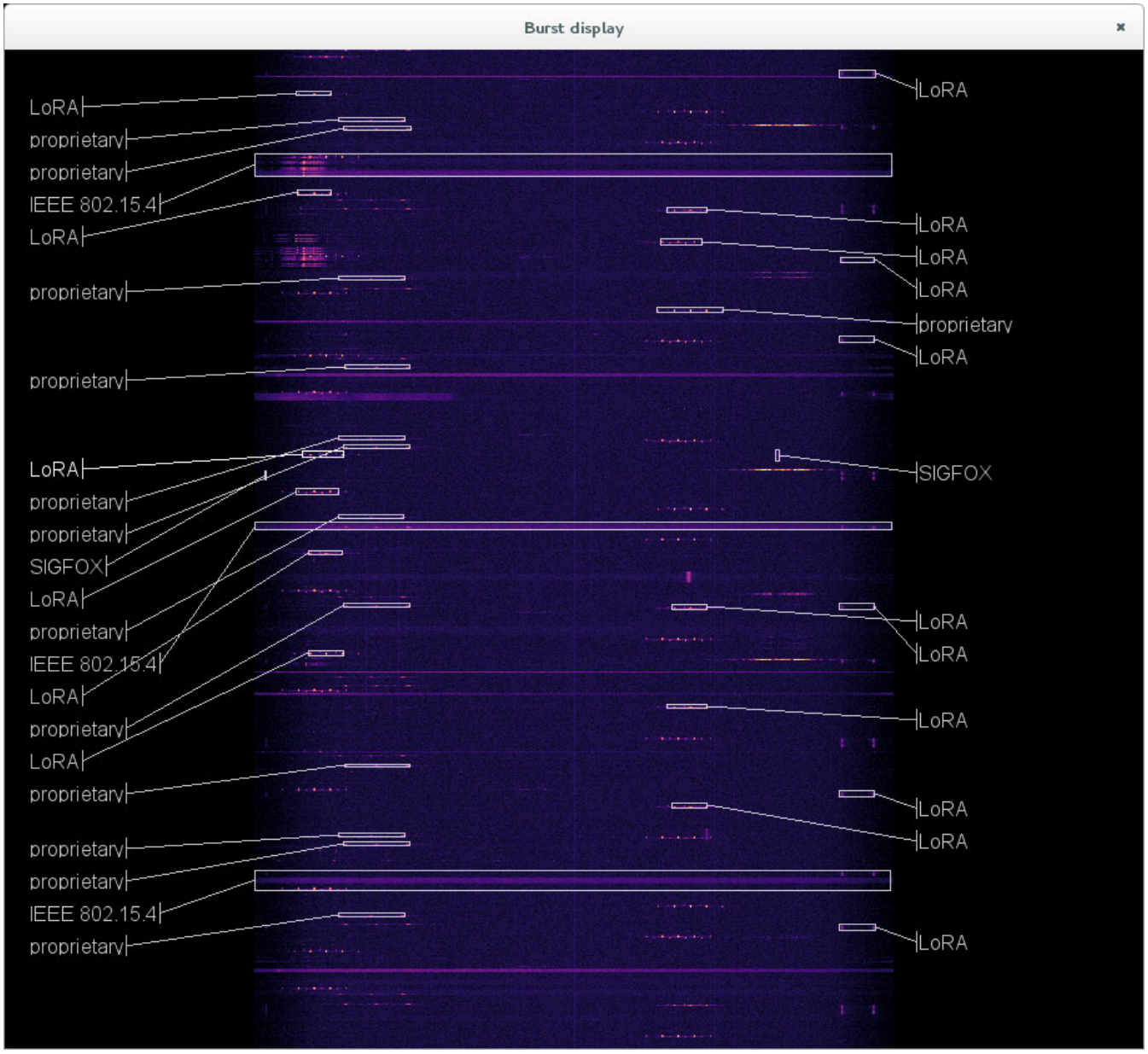


FIGURE 2. Example transmission recognition application that relies on automatically detected transmissions.

- Can be scaled to enable more users to label with controlled overlap, eventually leading to crowdsourcing of spectrum event detection.

In the data preprocessing step, the manual pipeline proposed in this paper is able to remove values from the PSD that are not considered high enough. It can be configured to display 1) all the PSD points - this corresponds to the noise floor + 0 dB, 2) noise floor + 3 dB as used in a study made at four locations in Guangdong, China [5], 3) noise floor + 6 dB corresponding to a study made in Singapore [22] and 4) noise floor + 10 dB corresponding to a recommendation from ITU [23]. For user rendering, the pipeline uses dark colors for low values of the PSD matrix and bright colors for transmissions for increased contrast. To further improve the contrast and therefore improve the efficiency of the pipeline, findings from human-computer interaction studies could be further considered [24].

The manual interaction with the pipeline requires three different actions from the user: 1) input the start corner of a transmission, 2) input the end corner of a transmission and 3) move to the next waterfall plot. By inputting the start and end corners of the transmission (see the white dots in Fig. 3), the pipeline will automatically map the two inputs on the image to the corresponding values in time and frequency. Then it will associate the matrix of pixels within the bounding box with the time/frequency values thus creating

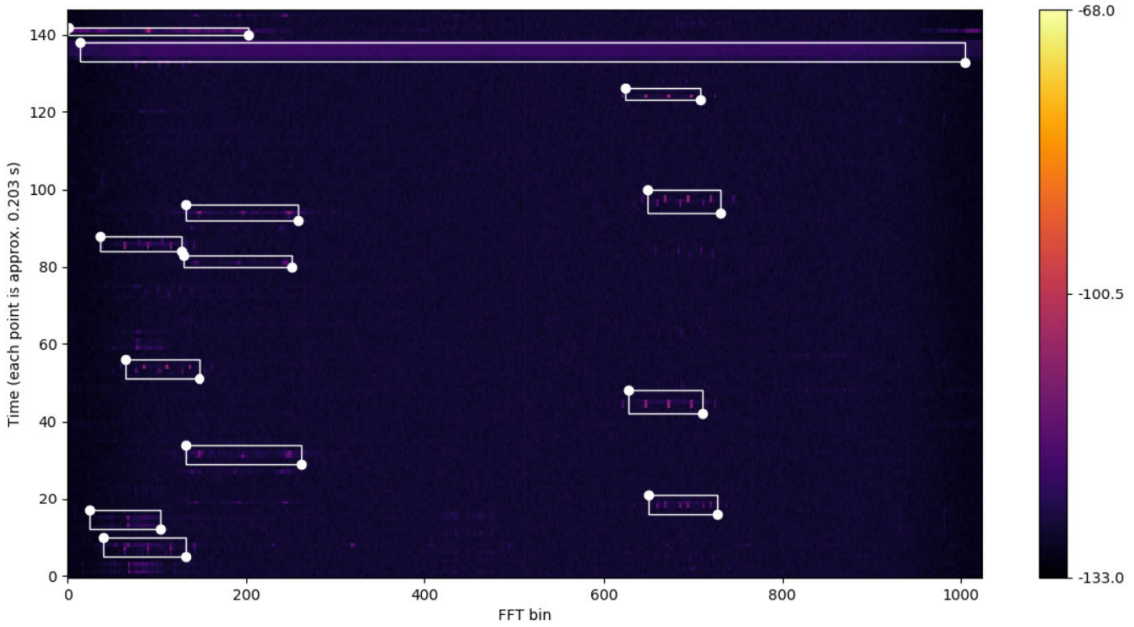


FIGURE 3. Screen with the labeling tool presenting the user with a random excerpt of the waterfall plot. The white bounding boxes represent the manually labeled transmissions.

a time/frequency labeled transmission. The inputs can be provided using standard peripherals such as a mouse or a touch screen.

A standard approach in machine learning communities for creating relevant manually labeled datasets is to present random excerpts of the data to the user. We employed this approach in our manual pipeline by extracting random slices of the spectrum of specified duration d that are x seconds apart, where x is randomly selected from an interval $[xmin, xmax]$. The two parameters, d and x, are configurable. This approach could be improved along several dimensions. First, by scaling the labeling process to a high number of human labelers and then using majority voting for label creation. Second, by only displaying slices that are sufficiently different from previously displayed ones or by displaying slices that contain transmissions detected with low confidence by the automatic system. For generating high quality manually labeled datasets, also findings from other scientific fields can be considered [11], [12].

The proposed pipeline for manual transmission detection is implemented in Python and is available as open-source software[7] to be used by the community with the aim to stimulate labeled dataset generation and sharing.

B. PIPELINE FOR AUTOMATIC TRANSMISSION DETECTION

The automatic process attempts to abstract the detection procedure of a human and automate it. Apart from having the clear advantage of being able to process vast amounts of data, automatic detection can also leverage the option of inexpensively running multiple algorithms on the same data. An agreement between the algorithms could be computed that would give a higher confidence in results.

For designing an automatic process, the following steps are required: 1) selecting and tuning an appropriate detection algorithm and 2) implementing and executing the algorithm for detection of wireless transmissions. Typically, an evaluation step follows that may re-trigger parts of the automatic process for fine-tuning purposes. This evaluation step uses selected performance criteria that measure the performance of the system according to some metrics.

The main effort in automatic transmission detection in wireless spectrum is carried out by the detection algorithms. We identify two main groups of detection algorithms, which contrast with respect to conceptual differences in the strategy of data processing and transmission detection: rule-based algorithms and computer vision algorithms. In the next subsections, we further explain and provide an example for each group of algorithms. The source code of our machine vision algorithm implementation is available on GitHub.[8]

1) RULE-BASED ALGORITHMS

This group of algorithms processes spectrum and detects transmissions in agglomerative fashion from the standpoint of every single spectrum point (measurement). As each transmission in the time-frequency spectrum representation can be decomposed into a set of points with a predefined resolution, a legitimate assumption is that this process is reversible if we can correctly identify these points and join them. We refer to these points as activity points.

[7]https://github.com/sensorlab/spectrum-labeling-tool

[8]https://github.com/sensorlab/sigfox-toolbox

Rule-based algorithms consist of two parts: detection of activity points and rule-based grouping. Detection of activity points, also called transmission candidate points, may be carried out by very simple approaches, such as thresholding. The goal of this step is to either accept or reject a hypothesis that a point is a part of some transmission. Once the candidate points have been detected, the points are grouped by consecutively applying a set of predefined grouping rules. One such rule may, for example, trivially group adjacent activity points in the frequency dimension. Whereas the rules generally only group the points, it is possible for a rule to also ignore already detected candidate transmission points. This allows for a higher degree of rule flexibility. Final groups of points represent the detected transmissions.

The correctness of rule-based algorithms heavily depends on the correct identification of candidate transmission points. These algorithms allow domain knowledge to be applied in a semantically straightforward manner, as the grouping process is transparent and directly reflects the natural visual grouping operations of a human.

As an example of a rule-based detection algorithm, we propose the *TX grouping algorithm*. The proposed algorithm is shown in Alg. 1. The input to the algorithm is a two-dimensional matrix of PSD values P, as defined in (1). The matrix can be infinite in time dimension as the algorithm can be easily adapted to also deal with streaming spectrum data by processing each row consecutively, beginning with the lowest (least recent) row. The algorithm has several configurable parameters, namely: 1) power threshold thr, 2) frequency grouping threshold df, 3) frequency pruning threshold p, 4) time grouping threshold dt and 5) frequency distance k. The output of the algorithm is represented by a list of bounding boxes, as defined in (2).

The algorithm works as follows. First, activity points are detected using a fixed threshold thr by passing through all elements and comparing them to the threshold as shown in lines 1-8 of Alg. 1. The effect of this code is visually depicted in Fig. 4a. Next, a series of simple grouping rules is applied:

- Frequency grouping rule (Alg. 1, lines 9-25): Group points in frequency that are not more than df FFT bins apart (see Fig. 4b). Function *fQuery* is defined as $fQuery(point, points, df) = \{p : p \in points \wedge p_t = point_t \wedge |p_f - point_f| \leq df\}$.
- Pruning rule (Alg. 1, lines 26-30): Remove groups that contain less than p points (see Fig. 4c). This step reduces incorrect detections attributed to noise.
- Time grouping rule (Alg. 1, lines 31-47): Join groups in time that are less than dt samples apart and the left-most frequency point of one group is less than k FFT bins apart from the left-most frequency point of the second group; equivalently for the right-most frequency points (see Fig. 4d). Function *tQuery* is defined as $tQuery(group, groups, dt, k) = \{g : g \in groups \wedge |g_t - group_t| \leq dt \wedge |max_f(g) - max_f(group)| \leq k \wedge |min_f(g) - min_f(group)| \leq k\}$.

Algorithm 1 TX Grouping Algorithm

Input: $P = p_{t,f} \in \mathbb{R}^{m \times n}$
Parameters: thr, df, p, dt, k
Output: $X = \{x_1, \ldots, x_n\}, x_n = (t_{start}, t_{stop}, f_{start}, f_{stop})_n$

1: $points \leftarrow \{\}$
2: **for** $t = 1$ to m **do**
3: **for** $f = 1$ to n **do**
4: **if** $p_{t,f} > thr$ **then**
5: $points \leftarrow points \cup \{(t, f)\}$
6: **end if**
7: **end for**
8: **end for**
9: $fGroups \leftarrow \{\}$
10: $visitedPoints \leftarrow \{\}$
11: **for** each $point \in points$ **do**
12: **if** $point \notin visitedPoints$ **then**
13: $visitedPoints \leftarrow visitedPoints \cup \{point\}$
14: $fNeighbors \leftarrow fQuery(point, points, df)$
15: $fGroup \leftarrow \{point\}$
16: **for** each $fNeighbor \in fNeighbors$ **do**
17: **if** $fNeighbor \notin visitedPoints$ **then**
18: $visitedPoints \leftarrow visitedPoints \cup \{fNeighbor\}$
19: $fNeighbors \leftarrow fNeighbors \cup fQuery(fNeighbor, points, df)$
20: $fGroup \leftarrow fGroup \cup \{fNeighbor\}$
21: **end if**
22: **end for**
23: $fGroups \leftarrow fGroups \cup \{fGroup\}$
24: **end if**
25: **end for**
26: **for** each $fGroup \in fGroups$ **do**
27: **if** $|fGroup| < p$ **then**
28: $fGroups \leftarrow fGroups \setminus fGroup$
29: **end if**
30: **end for**
31: $tGroups \leftarrow \{\}$
32: $visitedFGroups \leftarrow \{\}$
33: **for** each $fGroup \in fGroups$ **do**
34: **if** $fGroup \notin visitedFGroups$ **then**
35: $visitedFGroups \leftarrow visitedFGroups \cup \{fGroup\}$
36: $tNeighbors \leftarrow tQuery(fGroup, fGroups, dt, k)$
37: $tGroup \leftarrow \{fGroup\}$
38: **for** each $tNeighbor \in tNeighbors$ **do**
39: **if** $tNeighbor \notin visitedFGroups$ **then**
40: $visitedFGroups \leftarrow visitedFGroups \cup \{tNeighbor\}$
41: $tNeighbors \leftarrow tNeighbors \cup tQuery(tNeighbor, fGroups, dt, k)$
42: $tGroup \leftarrow tGroup \cup \{tNeighbor\}$
43: **end if**
44: **end for**
45: $tGroups \leftarrow tGroups \cup \{tGroup\}$
46: **end if**
47: **end for**
48: $X \leftarrow getBoundingBoxes(tGroups)$

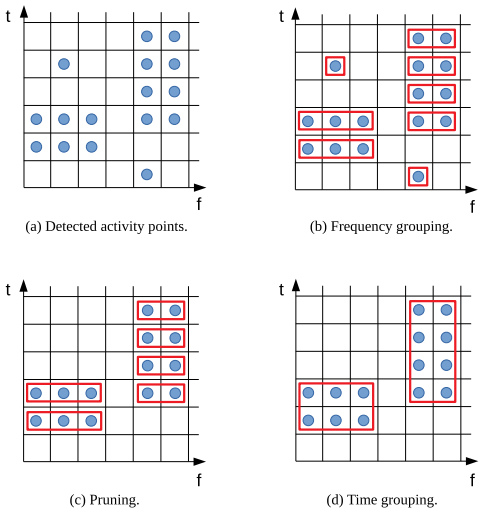


FIGURE 4. Application of grouping rules when $df = 1$, $p = 2$, $dt = 1$ and $k = 1$.

Algorithm 2 Dilate/Erode Algorithm

Input: $P = p_{t,f} \in \mathbb{R}^{m \times n}$
Parameters: $rThr$, $kSize$, d, s
Output: $X = \{x_1, \ldots, x_n\}, x_n = (t_{start}, t_{stop}, f_{start}, f_{stop})_n$

1: $G \leftarrow gaussianBlur(P, kSize)$
2: $B \leftarrow adaptiveThresholding(G, rThr)$
3: **for** $i = 0$ to d **do**
4: $\quad B \leftarrow dilate(B)$
5: **end for**
6: **for** $i = 0$ to d **do**
7: $\quad B \leftarrow erode(B)$
8: **end for**
9: $C \leftarrow findContours(B)$
10: $X \leftarrow getBoundingBoxes(C)$
11: **for** each $x \in X$ **do**
12: $\quad$ **if** $(x_{t_{stop}} - x_{t_{start}}) \times (x_{f_{stop}} - x_{f_{start}}) < s$ **then**
13: $\quad\quad X \leftarrow X \setminus x$
14: $\quad$ **end if**
15: **end for**

Final groups of points represent detected wireless transmissions. To obtain bounding boxes, minimum and maximum frequency as well as time of all points in each group are computed (Alg. 1, line 48). A list of these values for all detected wireless transmissions forms the output of the algorithm.

While the detection of activity points can be carried out by a simple thresholding approach such as presented, a possible enhancement may introduce more advanced techniques over one- and two-dimensional windows, for instance statistical modeling using distributions of energy levels and statistical hypothesis testing.

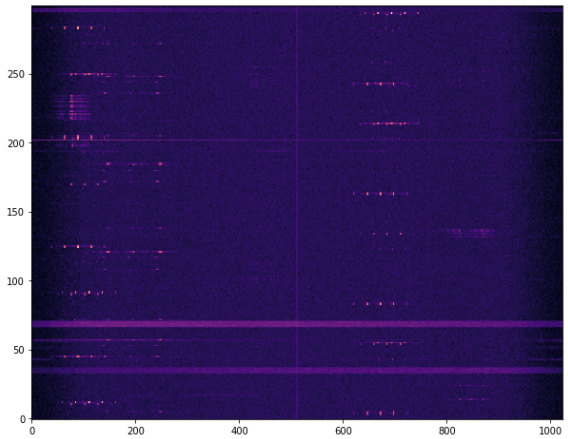
2) COMPUTER VISION ALGORITHMS

This group of algorithms processes spectrum as a whole and reduces it to a set of detected transmissions. These algorithms are based on the observation that a person can often immediately recognize individual transmissions based on their visual representations on the waterfall plot and determine their start and stop times and the occupied range of frequencies. Furthermore, with some experience a person can often tell the type of transmission and wireless technology in use, even though the waterfall plot is a severely reduced representation of the radio spectrum; it is typically greatly undersampled and contains no phase information.

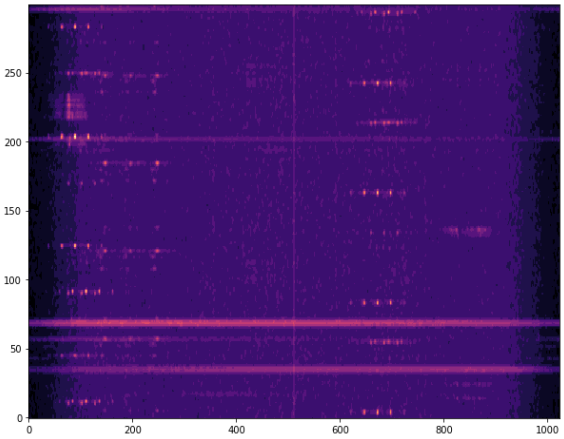
As an example of a computer vision detection algorithm, we outline our proposed *dilate/erode algorithm*. The proposed algorithm is shown in Alg. 2. The input to the algorithm is a two-dimensional matrix of PSD values P, as defined in (1). With streaming input data, the algorithm can operate on sub-matrices. The algorithm has several configurable parameters, namely: 1) relative power threshold $rThr$, 2) Gaussian kernel size $kSize$, 3) number of dilate/erode operations d and 4) pruning area s. The output of the algorithm is represented by a list of bounding boxes, as defined in (2).

The algorithm works as follows. After receiving a matrix of PSD values (see Fig. 5a), the algorithm first performs a two-dimensional Gaussian filter on the matrix (Alg. 2, line 1) using kernel of size $kSize$. This effectively blurs the waterfall plot and decreases the effect of noise by trading off some time and frequency resolution. Fig. 5b shows the waterfall plot after the filtering step. The next step performs adaptive thresholding (Alg. 2, line 2). Each element of the matrix is compared to a threshold value and replaced with the binary result of the comparison. The threshold is calculated for each individual matrix element separately. A weighted sum of neighboring values minus $rThr$ is used as a threshold. The weights are defined by a Gaussian window. Fig. 5c shows the result of the thresholding step. To further reduce the effect of noise, the binary matrix is then consecutively subjected to d dilate and d erode operations (Alg. 2, lines 3-8). A dilate operation replaces a matrix element with a logical OR function over a region. An erode operation similarly replaces a matrix element with a logical AND over a region. These steps are designed to remove single, isolated matrix elements that were above the threshold and most commonly represent noise. They also merge neighboring regions of the plot that appeared above the threshold into a single contiguous region. The matrix resulting from successive dilate and erode operations is shown in Fig. 5d. Finally, the data are converted from a binary matrix form to a list of bounding boxes, encircling the individual transmissions where only bounding boxes that have an area of at least s are kept (see Fig. 5e). These steps are shown in lines 9-15 of Alg. 2.

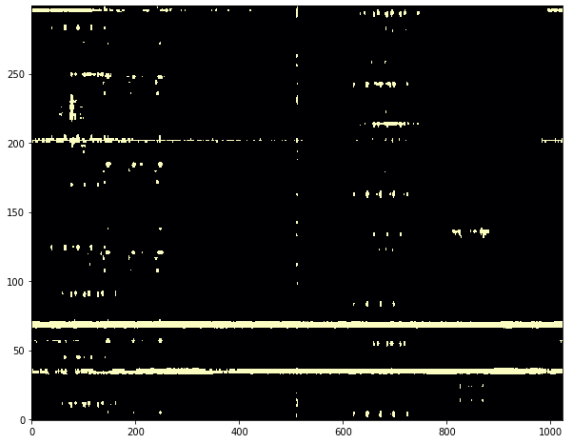
The most promising approach to improving the detection using machine vision algorithms is by using deep learning [9]. However, for this work, we selected algorithms that do not depend on training data. This way we are able to better focus and understand the effect of the labeled data on the evaluation of an automatic system.



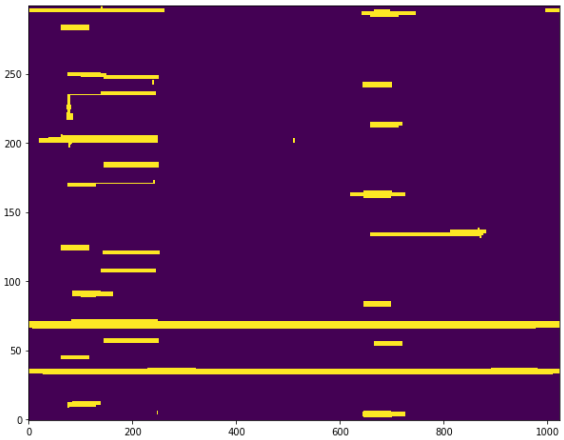
(a) Input matrix containing raw PSD data.



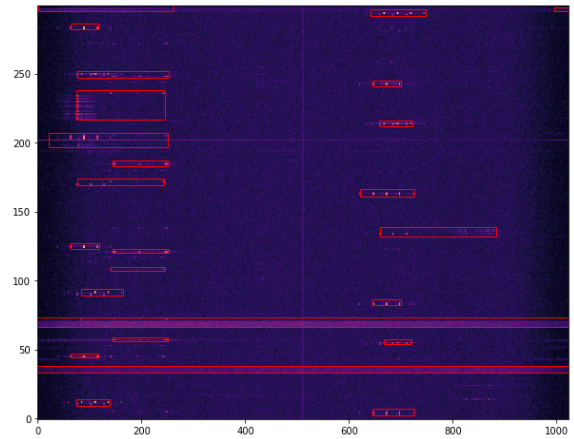
(b) Matrix after Gaussian filtering.



(c) Binary matrix after thresholding.



(d) Binary matrix after dilate and erode.



(e) Generated bounding boxes overlaid over raw PSD data.

FIGURE 5. The processing steps of the dilate/erode algorithm.

V. EVALUATION

This section covers various aspects of the evaluation. We first adapt existing metrics from machine learning for our purpose. Then we describe the used dataset and discuss performance aspects of the manual and automatic detection process.

A. METRICS

Before we can use an evaluation metric, we define the way two detected wireless transmissions are compared by introducing the concepts of transmission *span*, *overlap* and *intersection*. Even in the case of two clear transmissions that are

obviously the same, there might be small differences in their detection due to slight errors in the manual placement of the bounding boxes (see Section IV-A) or errors in the automatic bounding box creation (see Section IV-B). Therefore there might be deviations in their time/frequency parameters as defined in (2).

In machine learning, the most commonly used metrics for evaluating the performance of classification tasks are the *precision* and *recall* [25]. We adapt the *precision* and *recall* terms to our problem and we also introduce the *agreement precision* and *agreement recall* metrics that are somewhat similar to some of the metrics used in [9].

1) TRANSMISSION SPAN

We define the *span* of a transmission bounded by a box x as $span_m(x) = m_{stop} - m_{start}$ where m_{start} and m_{stop} can be time or frequency metrics ($m \in \{t, f\}$) as defined in (2). In other words, the time or frequency span of transmission x refers to the dimensions of that transmission as defined by the time-frequency bounding box - a visual illustration is presented in Fig. 6.

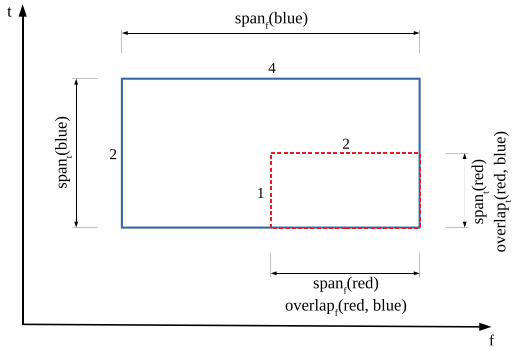


FIGURE 6. Visualization of span and overlap for two different transmissions. Frequency span is equal to 4 and 2 units while time span is 2 and 1 units for blue and red bounding boxes respectively. Frequency overlap between the two transmissions is 2 and time overlap is 1.

2) TRANSMISSION OVERLAP

We define the *overlap* of two transmissions bounded by boxes x_1 and x_2 as $overlap_m(x_1, x_2) = min(m_{stop_1}, m_{stop_2}) - max(m_{start_1}, m_{start_2})$ where m_{start_i} and m_{stop_i} can be time or frequency metrics ($m \in \{t, f\}$) as defined in (2). The *overlap* calculates the length of overlap or, in case there is no overlap, the minimal negative distance needed for the overlap to occur between one-dimensional projections of two transmissions in time on y-axis or frequency on x-axis.

Two transmissions x_1 and x_2 perfectly overlap when the spans of all metrics of one transmission are equal to the spans of all metrics of the other transmission and the overlaps are equal to the spans for all metrics: $span_m(x_1) = span_m(x_2) = overlap_m(x_1, x_2)$; $\forall m, m \in \{t, f\}$. In this case, the red and blue rectangles in Fig. 6 have the same area and overlap perfectly. However, in most cases, there is only partial overlap between transmissions. In the case of the example given in Fig. 6, the red dashed transmission overlaps on half time and half frequency span of the blue solid transmission.

3) TRANSMISSION INTERSECTION

We define transmission intersection, an extended set intersection operation, on two sets of transmissions X_1 and X_2 as a set of all transmissions from set X_1 whose bounding boxes overlap with a bounding box of at least one transmission from set X_2: $X_1 \cap X_2 = \{x_1 : x_1 \in X_1 \wedge \exists x_2 \in X_2 : overlap_m(x_1, x_2) > 0; \forall m, m \in \{t, f\}\}$. This definition implies that a single transmission from one set can intersect with multiple different transmissions from the other set. A regularization factor that alleviates the ramifications of such an assumption can be introduced. The transmission intersection operation and symbol are used synonymously to regular intersection operation in the rest of this section.

4) PRECISION AND RECALL

Precision is defined as the fraction of evaluated detected wireless transmissions that *intersect* the detected wireless transmissions attributed to ground truth, i.e. the fraction of evaluated detected transmissions that are correct with respect to the detected ground truth transmissions.

$$precision = \frac{|X_e \cap X_{gt}|}{|X_e|} \tag{3}$$

where X ($X = X_e \cup X_{gt}$) is the set of all detected transmissions, X_e is the set of evaluated transmissions detected using a manual or automatic pipeline and X_{gt} is the set of transmissions according to the ground truth detected using a manual or automatic pipeline.

Recall is defined as the fraction of detected wireless transmissions attributed to ground truth that *intersect* the evaluated detected wireless transmissions.

$$recall = \frac{|X_{gt} \cap X_e|}{|X_{gt}|} \tag{4}$$

5) AGREEMENT PRECISION AND AGREEMENT RECALL

These metrics measure the accuracy of overlaps (i.e. agreement) only on the transmissions that are in the intersection of both sets: 1) the evaluated detected transmissions and 2) the detected ground truth transmissions. In other words, agreement metrics are computed only for wireless transmissions from both sets that have overlapping bounding boxes. We define $\widetilde{X}_e \subseteq X_e$ as a set of evaluated detected transmissions that overlap with at least one detected ground truth transmission and $\widetilde{X}_{gt} \subseteq X_{gt}$ to be the set of detected ground truth transmissions that overlap with at least one evaluated detected transmission.

Agreement precision is defined as the average fraction of evaluated detected transmissions' bounding boxes that overlap with the detected ground truth transmissions' bounding boxes. For two overlapping transmissions $x_e \in \widetilde{X}_e$ and $x_{gt} \in \widetilde{X}_{gt}$ we can calculate the *agreement precision* as

$$precision_A(x_e, x_{gt}) = \frac{overlap_t(x_e, x_{gt}) + overlap_f(x_e, x_{gt})}{span_t(x_e) + span_f(x_e)}. \tag{5}$$

If $\widetilde{X}_{gt_{x_e}}$ is a set of all detected ground truth transmissions overlapping with an evaluated detected transmission x_e, the *average agreement precision* is then given by

$$precision_A = \frac{\sum_{x_e \in \widetilde{X}_e} \sum_{x_{gt} \in \widetilde{X}_{gt_{x_e}}} precision_A(x_e, x_{gt})}{|\widetilde{X}_e| + \sum_{x_e \in \widetilde{X}_e} |\widetilde{X}_{gt_{x_e}}|}. \quad (6)$$

Similarly, *agreement recall* is defined as an average fraction of detected ground truth transmissions' bounding boxes that overlap with evaluated detected transmissions' bounding boxes. It can be defined for two overlapping transmissions as

$$recall_A(x_e, x_{gt}) = \frac{overlap_t(x_e, x_{gt}) + overlap_f(x_e, x_{gt})}{span_t(x_{gt}) + span_f(x_{gt})}. \quad (7)$$

If $\widetilde{X}_{e_{x_{gt}}}$ is a set of all evaluated detected wireless transmissions overlapping with a detected ground truth transmission x_{gt}, the *average agreement recall* is given by

$$recall_A = \frac{\sum_{x_{gt} \in \widetilde{X}_{gt}} \sum_{x_e \in \widetilde{X}_{e_{x_{gt}}}} recall_A(x_e, x_{gt})}{|\widetilde{X}_{gt}| + \sum_{x_{gt} \in \widetilde{X}_{gt}} |\widetilde{X}_{e_{x_{gt}}}|}. \quad (8)$$

As time and frequency dimensions have different units, a weight function may be applied to each dimension in (5) and (7) to balance the discrepancies.

The first intuition when calculating *agreement precision* and *agreement recall* would be to use the area of the bounding boxes. However, area-based metrics do not aptly measure real performance. From Fig. 6, it can be seen that in the case of the two transmissions bounded by the blue solid rectangle and the red dashed rectangle, the area of correctly detected transmission would correspond to 25% whereas using the averaged time/frequency overlap the overlap is 50%.

B. DATASET

In this evaluation, we use 24 hours of continuous spectrum measurements in a 192 kHz wide band inside the unlicensed European 868 MHz SRD band recorded using a proprietary spectrum sensing device. The device was placed on top of a building in a mid-sized European city. We recorded 5 PSD measurements per second using 1024 FFT bins. During the measurement collection, we observed radio traffic common for this frequency band, such as IEEE 802.15.4, LoRA, Sigfox, and some proprietary transmissions. We also observed many transmissions of unknown origin and technology. We make the described unlabeled 24 hours worth of spectrum as well as random subsets with manual labels publicly available.[9]

C. MANUAL PROCESS

For evaluating the manual process, two experts first manually detected transmissions on pseudo-random 30 seconds long spectrum excerpts from the dataset, as described in Section IV. The excerpts were uniformly randomly extracted and were 10-15 minutes apart. Identical excerpts were presented to both experts to allow for cross-comparison. The two experts each labeled 4 runs consisting of 112 windows of 30-second spectrum excerpts totaling 56 minutes of labeled spectrum per run and needed on average approximately 90 minutes to label. The required time for labeling was larger by a factor of 1.6. From this experience, it seems that for detecting all transmissions on the complete 24-hour dataset, more than 38 hours of human labor would be required.

Expert 1 detected 1290, 1544, 1037 and 661 transmissions whereas expert 2 detected 1353, 1834, 1986 and 1844 transmissions for 0 dB, 3 dB, 6 dB and 10 dB data preprocessing configurations respectively (see Section IV-A), totaling 11549 manually detected transmissions.

1) THE INFLUENCE OF THRESHOLDS ON SELF-AGREEMENT

Tables 1 and 2 evaluate the influence of data preprocessing thresholds on the manual annotations of experts 1 and 2, measured by the *precision/recall* metrics. The columns of the table indicate the manually detected transmissions for various preprocessing thresholds that we fix as ground truth. The rows indicate the same detected transmissions for various thresholds that are subject to evaluation. For instance, the notation x dB (e) means that the dataset was generated at an x dB threshold used by the labeling tool and is subject to evaluation. x dB (gt) means that the dataset was generated at an x dB threshold used by the labeling tool and is considered as ground truth. As the two tables evaluate self-agreement at different preprocessing thresholds, they are symmetric,

TABLE 1. The influence of thresholds on the manual labels of expert 1. Evaluates self-agreement using precision/recall.

Thr./Thr.	0 dB (gt)	3 dB (gt)	6 dB (gt)	10 dB (gt)
	P/R	P/R	P/R	P/R
0 dB (e)	100% 100%	82.45% 97.21%	95.08% 80.08%	99.09% 53.57%
3 dB (e)	97.21% 82.45%	100% 100%	97.97% 69.43%	99.55% 45.73%
6 dB (e)	80.08% 95.08%	69.43% 97.97%	100% 100%	99.24% 63.93%
10 dB (e)	53.57% 99.09%	45.73% 99.55%	63.93% 99.24%	100% 100%

TABLE 2. The influence of thresholds on the manual labels of expert 2. Evaluates self-agreement using precision/recall.

Thr./Thr.	0 dB (gt)	3 dB (gt)	6 dB (gt)	10 dB (gt)
	P/R	P/R	P/R	P/R
0 dB (e)	100% 100%	78.19% 99.26%	75.18% 99.19%	77.01% 97.63%
3 dB (e)	99.26% 78.19%	100% 100%	91.89% 94.82%	93.76% 92.69%
6 dB (e)	99.19% 75.18%	94.82% 91.89%	100% 100%	94.47% 89.83%
10 dB (e)	97.63% 77.01%	92.69% 93.76%	89.83% 94.47%	100% 100%

[9] http://log-a-tec.eu/datasets

therefore reading only the part below the diagonal is sufficient.

From the results in Table 1, we can see that the higher the threshold set at preprocessing time, the lower the *precision* of the self-agreement for expert 1. Particularly, the *precision* between the 3 dB (e) evaluated set and the 0 dB (gt) ground truth set is 97% and it drops to 53% for the 10 dB (e) evaluated set. The *precision* between the 10 dB (e) set and the 6 dB (gt) drops to 63%. As expected, the *recall* undertakes the opposite trend and is the highest between the 10 dB (e) and the 6 dB (gt) sets. Similar observations hold for labeled datasets created by the second expert and listed in Table 2, but in general this expert achieved much better results in terms of *precision*.

For brevity, the results according to the *agreement precision* and *agreement recall* metrics are omitted from this section, however, the same conclusions can be drawn. In order to diminish the human bias and reduce the labeling error it is desirable that not only several independent labels are produced by multiple individuals, but also multiple labels are produced by the same expert.

2) THE INFLUENCE OF THRESHOLDS ON INTER-EXPERT AGREEMENT

Tables 3 and 4 evaluate the agreement between the two experts. The columns indicate the preprocessing thresholds at which the manual labels of both experts are compared. The rows specify which expert's labels are under evaluation as the labels of the other expert are fixed as ground truth. The results in Table 3 show that with the increase of the preprocessing threshold, the *precision* of the labels generated by the second expert is dropping. In this case, the drop can be explained by expert 1 detecting fewer transmissions than expert 2 due to transmissions attributed to background noise by the preprocessing step. However, once a transmission is labeled by both experts, it can be seen from Table 4 that the inter-expert agreement between overlapping detected transmissions remains almost constant and is not influenced by the threshold.

TABLE 3. The influence of thresholds on the manual labels. Evaluates inter-expert agreement using precision/recall.

Exp./Thr.	0 dB	3 dB	6 dB	10 dB
	P/R	P/R	P/R	P/R
Expert 1	94.19% 79.75%	98.64% 72.36%	99.23% 57.6%	99.55% 40.18%
Expert 2	79.75% 94.19%	72.36% 98.64%	57.6% 99.23%	40.18% 99.55%

TABLE 4. The influence of thresholds on the manual labels. Evaluates inter-expert agreement using agreement precision/recall.

Exp./Thr.	0 dB	3 dB	6 dB	10 dB
	AP/AR	AP/AR	AP/AR	AP/AR
Expert 1	80.80% 85.93%	79.90% 85.42%	74.61% 85.29%	80.98% 84.83%
Expert 2	85.93% 80.80%	85.42% 79.90%	85.29% 74.61%	84.83% 80.98%

D. AUTOMATIC PROCESS

For the evaluation of the automatic process, the algorithms discussed in Section IV-B were executed on the complete 24-hour dataset and thus automatically detected transmissions were obtained. The parameters for each algorithm were manually selected and no optimization was performed. The *TX grouping algorithm* detected 57,152 transmissions over 24 hours of spectrum sensing data. The *dilate/erode algorithm* detected only 36,204 transmissions.

1) INTER-ALGORITHM AGREEMENT

Table 5 presents the results of evaluating the two automatic algorithms against each other. The first row of the tables assumes that the ground truth is the set of labels generated by the *TX grouping algorithm* and evaluates the labels generated by the *dilate/erode* algorithm. The second row presents the opposite evaluation. The *precision* and *recall* are relatively high, signifying the algorithms identify similar phenomena in the spectrum as transmissions. Higher *precision* and lower *recall* of the *dilate/erode algorithm* as compared to the *TX grouping algorithm* may be attributed to discrepancies in the number of detected transmissions between the two algorithms. The algorithms disagree to some extent regarding the transmission bounding box placement, as indicated by a lower score of *agreement precision* and *agreement recall*.

TABLE 5. Inter-algorithm agreement for automatically detected transmissions.

	P	R	AP	AR
dilate/erode algorithm	93.35%	78.04%	61.26%	82.43%
TX grouping algorithm	78.04%	93.35%	82.43%	61.26%

2) DETECTION ALGORITHM EVALUATION

Tables 6 and 7 evaluate the automatically detected transmissions considering manually created labels of both experts as ground truth for different preprocessing thresholds. The results show that the *dilate/erode algorithm* exhibits slightly better *precision* (up to 85%) and *agreement recall*

TABLE 6. Precision and recall for the dilate/erode and the TX grouping algorithms with various manual detection thresholds.

Alg./Thr.	0 dB (gt)	3 dB (gt)	6 dB (gt)	10 dB (gt)
	P/R	P/R	P/R	P/R
dilate/erode algorithm	77.7% 86.3%	85.0% 79.3%	76.6% 78.4%	66.9% 79.9%
TX grouping algorithm	68.2% 94.7%	78.1% 92.5%	72.9% 90.4%	62.4% 94.5%

TABLE 7. Agreement precision and agreement recall for the dilate/erode and the TX grouping algorithms with various manual detection thresholds.

Alg./Thr.	0 dB (gt)	3 dB (gt)	6 dB (gt)	10 dB (gt)
	AP/AR	AP/AR	AP/AR	AP/AR
dilate/erode algorithm	74.7% 80.4%	73.1% 80.1%	73.2% 79.4%	68.5% 83.9%
TX grouping algorithm	80.2% 71.0%	80.2% 72.1%	80.5% 68.6%	76.4% 79.8%

(up to 83%). On the other hand, the *TX grouping algorithm* exhibits higher *recall* (up to 94%) and *agreement precision* (up to 80%). This means that, on unfiltered spectrum scans where noise is present, the *dilate/erode algorithm* detects transmissions more as a human would compared to the *TX grouping algorithm*. However, once a transmission is detected by both human and automatic processes, the overlap between the automatically generated bounding box compared to the manually generated bounding box is higher for the *TX grouping algorithm*.

Different preprocessing thresholds have a negligible impact on *agreement precision* and *agreement recall*, the difference is only noticeable at 10 dB threshold where *agreement precision* drops and *agreement recall* increases. This can be explained by the fact that by showing less information in spectrograms during the manual process results in less manually detected transmissions. *Precision* and *recall* are more susceptible to a threshold change, high dependence on the manual process shows in the mutual trends when evaluating the two algorithms at different thresholds, i.e. *precision* increases as the *recall* decreases for both algorithms when we compare results for 0 dB and 3 dB thresholds.

3) DETECTION ALGORITHM TUNING

Detection algorithm tuning represents an essential task for ensuring satisfactory automatic system performance. Tuning is tightly coupled with 1) data characteristics, such as sampling rate and sensed radio band, and 2) specific objective, i.e. whether we give any preference with respect to what we aim to detect in the spectrum.

In this particular case, we can optimize along one or all four metrics we use for evaluation as defined in Section V-A. Precision and recall are well known to often follow opposite trends [26], i.e., when one increases, the other decreases and vice versa. For example, optimizing only *precision* can result in a system that would always identify the entire spectrum as one large transmission even though the optimization successfully found a global maxima. Most often, adjustment of parameters will exhibit compromises, such as an increase in *precision* but a drop in *recall* simultaneously. Parameters should be jointly optimized, optimizing each parameter separately will often result in finding only a local optimum.

Section IV-B1 identified thr, df, p, dt and k for Alg. 1 as the parameters that affect the outcome of the algorithm while in section IV-B2 $rThr$, $kSize$, d and s for Alg. 2 are identified as important. The evaluation metrics for various configurations of these parameters are calculated using the dataset presented in Section V-B and manual labels for all preprocessing thresholds (both experts) from Section V-C.

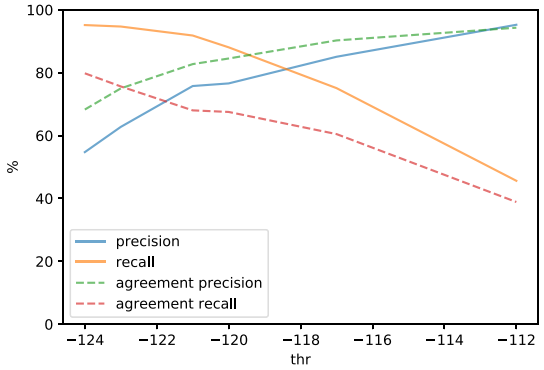


FIGURE 7. Evaluation metrics for the TX grouping algorithm for various values of *thr*.

Fig. 7 displays how different values of thr affect the performance of the *TX grouping algorithm*. We can see that there is an apparent *precision* versus *recall* and *agreement precision* versus *agreement recall* trade-off which corresponds to more detections at lower values of thr and less detections at higher values of thr. Considering our optimization goals, we can either decide to optimize *precision/recall* or *agreement precision/recall*. We select thr to be -121 as it maximizes the sum of all metrics. Similarly as thr, parameters p for the *TX grouping algorithm*, and $rThr$ and s for the *dilate/erode algorithm* influence the number of detections and consequently the detected transmissions' morphology. Parameters that more directly influence the morphology of detected transmissions are df, dt and k for the *TX grouping algorithm*, and $kSize$ and d for the *dilate/erode algorithm*. Whereas the number of detections is more closely related to *precision* and *recall*, the morphology mainly influences the *agreement precision* and *agreement recall* as these metrics measure the level of overlap between detected transmissions and ground truth.

As all parameters of both algorithms exhibit similar behavior as displayed in Fig. 7, we omit the figures, however, we list the optimal parameter values in Table 8 and Table 9.

TABLE 8. Optimal parameter selection for the TX grouping algorithm.

Parameter	thr	df	p	dt	k
Value	-121	50	20	4	5

TABLE 9. Optimal parameter selection for the dilate/erode algorithm.

Parameter	$rThr$	$kSize$	d	s
Value	-1	5	90	100

VI. CONCLUSION

In this paper, we proposed a framework that enables the design and development of automatic detection of wireless transmissions. Under this framework, we implemented and evaluated manual and automatic processes for transmission detection from PSD data. Our results confirm that generating human-labeled data is an expensive and imperfect process. Although limited in scale to over-generalize, the manual labeling of 8 sets of 56 minutes of continuous spectrum data took on average 90 minutes per set. Regardless of the inter-expert agreement on the labeled data reaching 99% in *precision* at 40% *recall* in particular cases, such agreement has a high variance by considering studies from larger manual labeling efforts conducted in other communities and more work should be conducted with respect to manual ground truth generation for wireless transmissions. The two automatic transmission detection algorithms evaluated in this study have a satisfactory performance of up to 93% *precision* with respect to each other and of up to 85% *precision* with respect to the manually generated labels used as ground truth. However, more work is necessary to understand the influence of the labeled data on detection algorithm evaluation and possible machine learning model training, especially now that significant efforts are being invested in solving various wireless communication problems using supervised methods.

REFERENCES

[1] Q. Zhao and B. M. Sadler, "A survey of dynamic spectrum access," *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 79–89, May 2007.

[2] M. Hoyhtya, A. Mammela, M. Eskola, M. Matinmikko, J. Kalliovaara, J. Ojaniemi, J. Suutala, R. Ekman, R. Bacchus, and D. Roberson, "Spectrum occupancy measurements: A survey and use of interference maps," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2386–2414, Apr. 2016.

[3] L. Shi, P. Bahl, and D. Katabi, "Beyond sensing: Multi-GHz realtime spectrum analytics.," in *Proc. NSDI*, 2015, pp. 159–172.

[4] Z. Qin, Y. Gao, M. D. Plumbley, and C. G. Parini, "Wideband spectrum sensing on real-time signals at sub-Nyquist sampling rates in single and cooperative multiple nodes," *IEEE Trans. Signal Process.*, vol. 64, no. 12, pp. 3106–3117, Jun. 2016.

[5] S. Yin, D. Chen, Q. Zhang, M. Liu, and S. Li, "Mining spectrum usage data: A large-scale spectrum measurement study," *IEEE Trans. Mobile Comput.*, vol. 11, no. 6, pp. 1033–1046, Jun. 2012.

[6] M. A. Mchenry, P. A. Tenhula, D. Mccloskey, D. A. Roberson, and C. S. Hood, "Chicago spectrum occupancy measurements & analysis and a long-term studies proposal," in *Proc. 1st Int. Workshop Technol. Policy Accessing Spectr. (TAPAS)*, 2006, p. 1.

[7] M. H. Islam, C. L. Koh, S. W. Oh, X. Qing, Y. Y. Lai, C. Wang, Y.-C. Liang, B. E. Toh, F. Chin, G. L. Tan, and W. Toh, "spectrum survey in singapore: Occupancy measurements and analyses," in *Proc. 3rd Int. Conf. Cognit. Radio Oriented Wireless Netw. Commun.*, May 2008, pp. 1–7.

[8] T. Šolc, M. Mohorčič, and C. Fortuna, "A methodology for experimental evaluation of signal detection methods in spectrum sensing," *PLoS ONE*, vol. 13, no. 6, Jun. 2018, Art. no. e0199550.

[9] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "From facial parts responses to face detection: A deep learning approach," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3676–3684.

[10] S. Rajendran, R. Calvo-Palomino, M. Fuchs, B. Van Den Bergh, H. Cordobes, D. Giustiniano, S. Pollin, and V. Lenders, "Electrosense: Open and big spectrum data," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 210–217, Jan. 2018.

[11] R. Snow, B. O'Connor, D. Jurafsky, and A. Y. Ng, "Cheap and fast–but is it good?: Evaluating non-expert annotations for natural language tasks," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2008, pp. 254–263.

[12] S. Nowak and S. Rüger, "How reliable are annotations via crowdsourcing: A study about inter-annotator agreement for multi-label image annotation," in *Proc. Int. Conf. Multimedia Inf. Retr. (MIR)*, 2010, pp. 557–566.

[13] A. A. A. Mansour, "Labeling Agreement Level and Classification Accuracy," in *Proc. 12th Int. Conf. Signal-Image Technol. Internet-Based Syst. (SITIS)*, 2016, pp. 271–274.

[14] S. Rajendran, W. Meert, V. Lenders, and S. Pollin, "SAIFE: Unsupervised wireless spectrum anomaly detection with interpretable features," in *Proc. IEEE Int. Symp. Dynamic Spectr. Access Netw. (DySPAN)*, Oct. 2018, pp. 1–9.

[15] N. Kleber, A. Termos, G. Martinez, J. Merritt, B. Hochwald, J. Chisum, A. Striegel, and J. N. Laneman, "RadioHound: A pervasive sensing platform for sub-6 GHz dynamic spectrum monitoring," in *Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw. (DySPAN)*, Mar. 2017, pp. 1–2.

[16] M. Zheleva, T. Larock, P. Schmitt, and P. Bogdanov, "Efficient spectrum summarization using compressed spectrum scans," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Apr. 2018, pp. 1–2.

[17] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 3, pp. 433–445, Sep. 2018.

[18] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Proc. Int. Conf. Eng. Appl. Neural Netw.* Cham, Switzerland: Springer, 2016, pp. 213–226.

[19] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, Jan. 2018.

[20] A. Kejariwal, S. Kulkarni, and K. Ramasamy, "Real time analytics: Algorithms and systems," Aug. 2017, *arXiv:1708.02621*. [Online]. Available: https://arxiv.org/abs/1708.02621

[21] A. Nafkha, M. Naoues, K. Cichon, and A. Kliks, "Experimental spectrum sensing measurements using USRP software radio platform and GNU-radio," in *Proc. 9th Int. Conf. Cognit. Radio Oriented Wireless Netw.*, 2014, pp. 429–434.

[22] M. H. Islam, C. L. Koh, S. W. Oh, X. Qing, Y. Y. Lai, C. Wang, Y.-C. Liang, B. E. Toh, F. Chin, G. L. Tan, and W. Toh, "Spectrum survey in singapore: Occupancy measurements and analyses," in *Proc. 3rd Int. Conf. Cognit. Radio Oriented Wireless Netw. Commun.*, May 2008, pp. 1–7.

[23] ITU Radiocommunication Bureau, (2002). *Handbook Spectrum Monitorin*. https://extranet.itu.int/brdocsearch/R-HDB/R-HDB-23/R-HDB-23-2002/R-HDB%-23-2002-OAS-PDF-E.pdf

[24] D. Albers, M. Correll, and M. Gleicher, "Task-driven evaluation of aggregation in time series visualization," in *Proc. 32nd Annu. ACM Conf. Human Factors Comput. Syst. (CHI)*, 2014, pp. 551–560.

[25] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 233–240.

[26] M. Buckland and F. Gey, "The relationship between Recall and precision," *J. Amer. Soc. Inf. Sci.*, vol. 45, no. 1, pp. 12–19, Jan. 1994.

TIMOTEJ GALE received the B.Sc. degree in computer and information science from the Faculty of Computer and Information Science, University of Ljubljana, in 2017, where he is currently pursuing the M.Sc. degree. He works as a Student Researcher with the Department of Communication Systems, Jožef Stefan Institute, and a Developer at multiple companies. His main research interests include application of data mining and machine learning methods and concepts to a wide range of problems in telecommunication systems and networks, with a focus on data-driven technologies for the Internet of Things.

TOMAŽ ŠOLC received the B.Sc. degree in electronics from the Faculty of Electrical Engineering, University of Ljubljana, in 2007. He is currently pursuing the Ph.D. degree with the Jožef Stefan International Postgraduate School. He is currently a Senior Research Assistant with the Department of Communication Systems, Jožef Stefan Institute. His work involves hardware design, measurements, and embedded software development with the Laboratory for Wireless Sensor Networks, and research in the field of spectrum sensing. He participated in various national and EU projects involving advanced radio technologies, including H2020 eWINE, FP7 CREW, FP7 Fed4Fire, and the U.K. Ofcom TV White-space trials. In FP7 CREW, he was a Technical Lead in development of an outdoor wireless testbed.

RAREŞ-ANDREI MOŞOI received the B.Sc. degree in telecommunication technologies and systems from the Faculty of Electronics, Telecommunications, and Information Technology, University Politehnica of Bucharest, in 2018. He is currently pursuing the M.Sc. degree in telecommunications with the Faculty of Electronics, Telecommunications, and Information Technology, Technical University of Cluj-Napoca. He works as a Visiting Student with the Department of Communication Systems, Jožef Stefan Institute. He gained experience in the industry by working with Vodafone Shared Services Romania and fme SRL. His research interests are focused on cognitive radio networks, telecommunication technologies, and spectral detection and identification of transmissions.

MIHAEL MOHORČIČ (Senior Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering from the University of Ljubljana, Ljubljana, in 1994, 1998, and 2002, respectively, and the M.Phil. degree in electrical engineering from the University of Bradford, Bradford, U.K., in 1998. He is currently the Head of the Department of Communication Systems and a Scientific Counselor with the Jožef Stefan Institute, and an Associate Professor with the Jožef Stefan International Postgraduate School. He has authored or coauthored over 180 refereed journal and conference papers, and coauthored three books and contributed to nine book chapters. His research interests include the development and performance evaluation of network protocols and architectures for mobile and wireless communication systems, and resource management in terrestrial, stratospheric, and satellite networks. His recent research interests are focused on cognitive radio networks, smart applications of wireless sensor networks, dynamic composition of communication services, and wireless experimental testbeds. He has participated in many EC co-funded research projects, since 1996. He is currently involved in H2020 projects DEFENDER, Fed4FIRE+, NRG-5, SAAM, and RESILOC.

CAROLINA FORTUNA received the B.Sc. and Ph.D. degrees, in 2006 and 2013, respectively. She was a Postdoctoral Research Associate with IBCN, Ghent University, from 2014 to 2015. She is currently a Research Fellow with the Department of Communication Systems, Jožef Stefan Institute and an Assistant with the Jožef Stefan International Postgraduate School. Her research is interdisciplinary, focusing on data and knowledge driven modeling of communication and sensor systems. She has participated in H2020, FP7, and FP6 projects. In H2020 WiSHFUL, she was the Technical Leader of the project on behalf of UGhent/iMinds, while in FP7 CREW, she was the Technical Leader of the JSI Team. She has coauthored over 50 peer-reviewed publications. She was a TPC Member at IEEE ICC 2011, 2012, 2013, 2014, 2016, ESWC 2012, IEEE Globecom 2011, 2016, VTC 2010, 2016, IEEE WCNC 2009, and gained industrial experience by working with Bloomberg LP and Siemens PSE.