

Accurate Automatic Extraction of Signal Components from Noisy Radio Spectrograms

Miguel López-Benítez, *Senior Member, IEEE*, and Mohammed M. Alammar, *Member, IEEE*

Abstract—In some radio communication scenarios it is useful to extract the bandwidth and start/end times of each transmission in a spectrogram. However, few methods in the literature are able to extract this information without human manual intervention, and the few existing ones have a limited accuracy. In this context, this work overcomes these limitations by proposing a novel fully automated method that can provide this information with high accuracy. The results obtained from simulations and hardware experiments show that the proposed method outperforms other methods available in the literature, achieving a virtually perfect accuracy for signal-to-noise ratio values as low as -10 dB.

Index Terms—Spectrogram, clustering algorithms, signal area estimation, spectrum awareness, autonomous wireless systems.

I. INTRODUCTION

RADIO spectrograms describe the temporal evolution of the power spectral density of wireless communication signals and provide essential information such as the received signal strength, carrier frequency, occupied bandwidth, spectral mask and transmission pattern. Spectrograms have been used to address various problems in wireless communications such as automatic blind modulation classification (based on heuristic algorithms [1]–[3] and convolutional neural networks [4]–[6]), radio technology identification [7], interference detection and mitigation [8], detection and localisation of radio events [9], radio signal denoising [10], extraction of frequency hopping signal parameters [11], [12], spectrum sensing [13], detection of radar signals [14] and characterisation of the Signal-to-Noise Ratio (SNR) and Doppler shift [15].

This work addresses the problem of automatically extracting the signal components present in a spectrogram. This entails not only detecting the signal components but also determining the time-frequency area that each of them occupy within the spectrogram, which in this work is referred to as Signal Area (SA). An SA is a rectangular cluster of spectrogram points where a signal transmission is present. As such, an SA indicates precisely the occupied bandwidth and start/end time of each radio transmission. The ability to extract this information automatically from a spectrogram opens the possibility to make intelligent decisions for context-aware autonomous wireless spectrum monitoring systems. This kind of technique

can find a broad range of applications in wireless communication systems, including for instance spectrum surveillance for enforcement of spectrum regulations or gathering of signal intelligence in military applications, signal/emitter interception and identification, electronic warfare and radio environment spectral awareness (e.g., databases for spectrum sharing).

The automatic extraction (i.e., without human manual intervention) of communication signals from a radio spectrogram is a challenging problem due to the degrading effect of radio propagation (which leads to random signal missed detections) and the unavoidable presence of noise in any radio receiver (which introduces random false alarms in the spectrogram). Despite its practical relevance, this problem has not been explored extensively in the literature. In [16], a region growing algorithm controlled by the spectrogram’s first- and second-order statistics is proposed for spectrogram segmentation, which can differentiate deterministic signal components from background noise and classify them into separate regions, however without estimating the SA of each component. The work presented in [17] proposes a computer vision approach that applies a fixed threshold to the spectrogram to produce a binary image along with morphological operations as an adaptive threshold to remove extraneous detections and finally extracts the image blobs by grouping connected components and calculating their bounding boxes. This method is modified in [18] by introducing an auto-thresholding technique and a bi-directional self-organising neural network to reduce noise after thresholding, which is reported in [18] to achieve a probability of detection $P_d = 98.27\%$ and a probability of false alarm $P_{fa} = 5.4\%$, however at an unspecified SNR. In [19], the authors propose the use of a single shot multibox detector network [20], which is a classical deep learning based object detector, for signal component extraction, however admitting that the desired performance is not attained. This limitation is addressed in [21] by introducing convolutional layers, which provides an accurate detection (for SNR values above 0 dB) at the expense of an increased complexity and the requirement of training. A more computationally efficient Simple Signal Area (SSA) estimation method is proposed in [22], which performs a raster scan to find the first corner of each SA, followed by horizontal scanning to estimate the SA width and coarse/fine vertical scanning to estimate the SA height. Some variants to reduce the impact of false alarms are proposed in [23]–[25].

The approach proposed in this work exploits the clustered structure of radio signals in the time-frequency domain and the fact that the density of data points per unit area is higher inside SAs than outside, even at low SNR. A two stage approach is employed in this work to exploit these observations. First, a

M. López-Benítez is with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, L69 3GJ, United Kingdom, and also with the ARIES Research Centre, Antonio de Nebrija University, 28040 Madrid, Spain (email: m.lopez-benitez@liverpool.ac.uk).

M. M. Alammar is with the Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, L69 3GJ, United Kingdom, and also with the Department of Electrical Engineering, King Khalid University, Abha, Asir 61421, Saudi Arabia (email: m.m.alammar@liverpool.ac.uk).

Final author version (September 2022).

spectrogram clustering stage groups data points belonging to the same SA based on the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [26]. Afterwards, a Signal Area Estimation (SAE) stage estimates the SA of each cluster with a simple and efficient algorithm that can remove false alarms accidentally included in the cluster.

To the best of the authors' knowledge, the most closely related methods available in the literature are those proposed in [27] and [28], which rely on a similar two stage approach as the one considered in this work. The Transmission Encapsulation based on the Connected Component Labelling (TECCL) method proposed in [27] performs clustering based on connected component labelling [29] and estimates the SA of each cluster as its extreme dimensions (bounding box). This method can be implemented using standard contour/boundary tracing techniques [30] (see [22] for instance). In [28], a Mean-Shift Clustering (MSC) algorithm is employed for clustering while the SAE stage relies on a scanning window whose dimensions need to be adjusted manually to each empirical scenario according to the expected bandwidth and transmission duration of the signal components to be detected. As a result, the method proposed in [28] needs to be applied multiple times to the same spectrogram, one for each expected SA dimension, which not only incurs in a high computational cost but also introduces the problem of multiple detections of the same signal component. On the other hand, the method here proposed is based on DBSCAN. The worst-case computational complexity of DBSCAN is $\mathcal{O}(n^2)$, the same as MSC, however the proposed method needs to be applied only once to each spectrogram, thus having a much lower computational cost. Moreover, the proposed method is fully automated and does not require signal-dependent, manual parameter tuning. Furthermore, DBSCAN can detect and isolate outliers (false alarms), while MSC clusters all spectrogram points, including false alarms. As a result, the proposed method can achieve a better performance. In particular, the method proposed in [28] requires a minimum SNR of 5 dB to achieve $P_d = 90\%$ and $P_{fa} = 12\%$, while the method proposed in this work can provide a virtually perfect detection performance ($P_d \approx 100\%$ and $P_{fa} \approx 0\%$) for SNR as low as -10 dB.

In summary, while some methods are available to detect SAs in radio spectrograms, most are unable to automatically provide information about the characteristics of the detected SAs, and the few that are able to do so have a rather limited accuracy. In this context, the main contribution of this work is a novel method that overcomes these limitations. The proposed method can extract the features of each SA present in a radio spectrogram (number of SAs and their dimensions in the time and frequency domains) in a completely automated manner, without requiring any form of manual intervention from the user. Moreover, the proposed method provides high detection accuracy, outperforming existing methods both in terms of sensitivity and maximum attainable accuracy.

The remainder of this work is organised as follows. First, Section II presents the considered system model and a formal definition of the problem addressed in this work. The algorithms utilised for spectrogram clustering and SAE along with the design of their configuration parameters are dis-

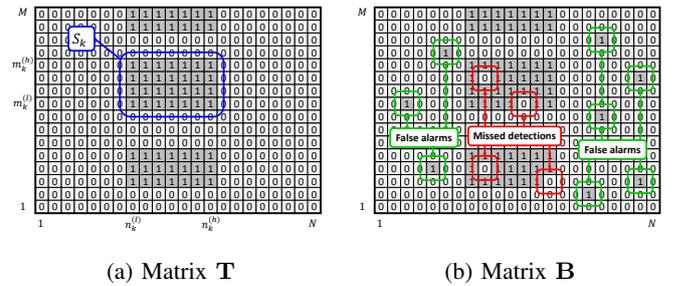


Fig. 1: Matrix model for radio spectrograms: (a) ground truth at the transmitter (matrix **T**), (b) states observed at the receiver (matrix **B**).

cussed in Sections III and IV, respectively. Section V then describes the methodology to assess the performance of the proposed method, followed by the analysis and discussion of the obtained simulation and experimental results in Section VI. Finally, Section VII summarises and concludes this work.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A radio spectrogram can be mathematically represented as a matrix $\mathbf{P} = [p_{m,n}] \in \mathbb{R}^{M \times N}$ of $M \times N$ power values. The convention adopted in this work is that the horizontal axis of \mathbf{P} is associated with frequency while the vertical axis corresponds to time. Each element $p_{m,n} \in \mathbb{R}$ ($m = 1, \dots, M$ and $n = 1, \dots, N$) is the power level measured at the m th time instant and n th frequency point, the vertical size M is the number of temporal samples (i.e., number of frequency sweeps taken over a given time interval) and the horizontal size N is the number of frequency points over the target frequency span.

In order to decide on the presence of signal components in the spectrogram, a frequency-dependent decision threshold $\boldsymbol{\lambda} = [\lambda_n] \in \mathbb{R}^{1 \times N}$ is calculated. The use of a threshold value for each frequency point is preferred when processing experimental power measurements due to the frequency-dependent nature of noise in practical radio systems. The threshold can be calculated based on different methods as discussed in [31].

The selected threshold is employed to convert the matrix of continuous power values \mathbf{P} observed at the receiver into a binary matrix $\mathbf{B} = [b_{m,n}] \in \mathbb{B}^{M \times N}$ (with $\mathbb{B} = \{0, 1\}$) where each element is calculated as:

$$b_{m,n} = \begin{cases} 0, & p_{m,n} < \lambda_n \\ 1, & p_{m,n} \geq \lambda_n \end{cases} \quad (1a)$$

$$b_{m,n} = \begin{cases} 0, & p_{m,n} < \lambda_n \\ 1, & p_{m,n} \geq \lambda_n \end{cases} \quad (1b)$$

indicating whether each spectrogram element is believed to contain a signal component ($b_{m,n} = 1$) or not ($b_{m,n} = 0$).

Let $\mathbf{T} = [t_{m,n}] \in \mathbb{B}^{M \times N}$ be a matrix whose elements $t_{m,n}$ contain the true states of the elements $b_{m,n}$ of matrix \mathbf{B} at the receiver (see Fig. 1). Matrix \mathbf{T} represents the ground truth for the spectrogram observed by the receiver. Notice that, in a practical system implementation, \mathbf{T} is only known by the transmitter(s); the receiver does not know \mathbf{T} but can attempt to estimate it based on the information available from matrix \mathbf{B} . However, some of the elements of \mathbf{B} will unavoidably be incorrect due to errors in the signal transmission and detection process, which can be characterised in terms of the false alarm probability $P_{fa} = P(b_{m,n} = 1 | t_{m,n} = 0)$ and detection probability $P_d = P(b_{m,n} = 1 | t_{m,n} = 1)$. Matrix \mathbf{B} can be

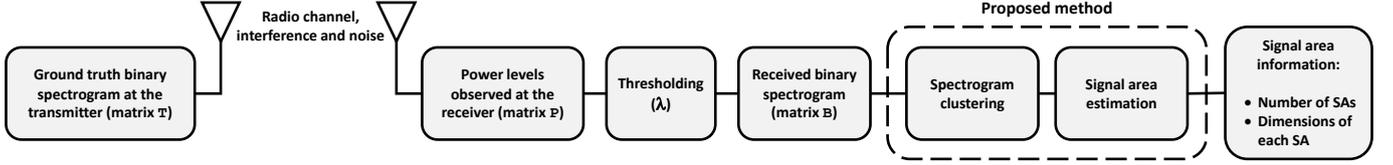


Fig. 2: Overall signal process from the transmitter to the final estimated information at the receiver.

seen as a degraded version of \mathbf{T} where random errors are introduced with probabilities P_{fa} and $1 - P_d$. In an ideal scenario of infinite (sufficiently high) SNR it is possible to set a threshold λ such that $P_{fa} = 0$ and $P_d = 1$ so that $\mathbf{B} = \mathbf{T}$. However, in many practical scenarios $\mathbf{B} \neq \mathbf{T}$ in general.

Let $K \in \mathbb{N}$ denote the number of unique signal transmissions present in a spectrogram, with the k th transmission taking place between low/high time indices $m_k^{(l)} \leq m \leq m_k^{(h)}$ and low/high frequency indices $n_k^{(l)} \leq n \leq n_k^{(h)}$, respectively ($1 \leq m_k^{(l)} \leq m_k^{(h)} \leq M$ and $1 \leq n_k^{(l)} \leq n_k^{(h)} \leq N \forall k \in \{1, \dots, K\}$). Every unique signal transmission will lead to a submatrix within \mathbf{T} with all its elements equal to one (see Fig. 1a). Each of these submatrices can be visualised as a rectangular cluster of points whose height $m_k^{(h)} - m_k^{(l)} + 1$ corresponds to the k th transmission duration and its width $n_k^{(h)} - n_k^{(l)} + 1$ corresponds to the k th signal bandwidth. Each of these clusters is referred to as an SA in this work. The k th SA can be formally defined as the subset of points $\mathcal{S}_k = \{(m, n) : m_k^{(l)} \leq m \leq m_k^{(h)}, n_k^{(l)} \leq n \leq n_k^{(h)}\}$. All SAs in a spectrogram are assumed not to overlap ($\bigcap_{k=1}^K \mathcal{S}_k = \emptyset$). Notice that, by definition, $t_{m,n} = 0 \forall (m, n) \notin \mathcal{S}_k \forall k$ and $t_{m,n} = 1 \forall (m, n) \in \mathcal{S}_k \forall k$, however this is not true in general for the corresponding elements $b_{m,n}$ of matrix \mathbf{B} due to errors in the signal transmission and detection process (see Fig. 1b).

The purpose of this work is to determine automatically and as accurately as possible the number K of SAs present in \mathbf{T} and the dimensions of each of them (given by $m_k^{(l)}, m_k^{(h)}, n_k^{(l)}, n_k^{(h)} \forall k \in \{1, \dots, K\}$) based on the binary spectrogram \mathbf{B} resulting from thresholding a radio spectrogram \mathbf{P} obtained from experimental power measurements. The complete signal process is illustrated in Fig. 2.

To address this problem, a two stage approach is proposed in this work (dashed box in Fig. 2). The first stage applies a clustering algorithm to matrix \mathbf{B} to produce a clustered matrix $\mathbf{C} = [c_{m,n}] \in \mathbb{Z}^{M \times N}$ with elements:

$$c_{m,n} = \begin{cases} 0, & b_{m,n} = 0 \\ -1, & b_{m,n} = 1 \text{ and outlier (false alarm)} \\ \tilde{k}, & b_{m,n} = 1 \text{ and member of } \tilde{k}\text{th SA} \end{cases} \quad (2a)$$

where $\tilde{k} \in \{1, \dots, \tilde{K}\}$. This will implicitly provide an estimation of the number of SAs as $\tilde{K} = \max_{m,n}(c_{m,n})$ along with a set of \tilde{K} clusters that can be extracted from matrix \mathbf{C} as the subsets $\mathcal{C}_{\tilde{k}} = \{(m, n) : c_{m,n} = \tilde{k}\} \forall \tilde{k} \in \{1, \dots, \tilde{K}\}$. Notice that the subsets $\mathcal{C}_{\tilde{k}}$ do not have in general a perfectly rectangular, solid shape and therefore do not meet the definition of SA given by \mathcal{S}_k . However, if the clustering stage performs well ($\tilde{K} = K$), each $\mathcal{C}_{\tilde{k}}$ should be the *noisy* version of a true SA \mathcal{S}_k and therefore can be used to estimate the SA parameters. Thus,

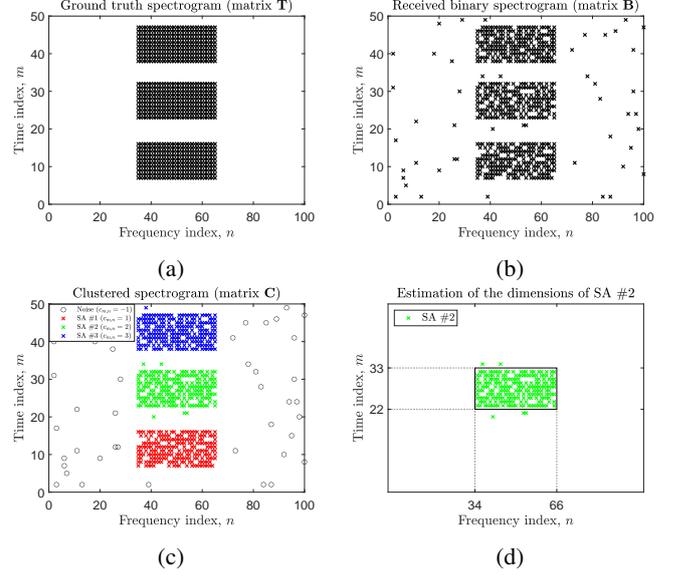


Fig. 3: Illustration of the operation of the proposed method: (a) ground truth spectrogram at transmitter (matrix \mathbf{T}), (b) binary spectrogram observed at receiver (matrix \mathbf{B}), (c) output of spectrogram clustering (matrix \mathbf{C}), and (d) estimation of SA dimensions.

the second stage of the proposed method estimates the dimensions of each detected cluster (SA) by providing estimates of the parameters $\tilde{m}_k^{(l)}, \tilde{m}_k^{(h)}, \tilde{n}_k^{(l)}, \tilde{n}_k^{(h)} \forall k \in \{1, \dots, \tilde{K}\}$ through an appropriate algorithm that can fit rectangles to noisy data.

The operation principle of the proposed method is illustrated in Fig. 3 with some sample spectrograms. Fig. 3a shows an example of the ground truth at the transmitter (matrix \mathbf{T}), which the receiver attempts to estimate based on a degraded version (matrix \mathbf{B}) shown in Fig. 3b. To this end, the points of \mathbf{B} are first clustered as illustrated in Fig. 3c so that points that are likely to belong to the same SA are grouped together while points that are likely to be caused by noise (i.e., false alarms) are isolated and not processed further. This clustering stage meets several important purposes. First, it removes most (but not all) spectrogram points caused by the receiver's noise (false alarms) in order to facilitate a more accurate estimation of the dimensions of each SA in the subsequent stage. Second, by clustering the remaining data points into a number of prospective SAs, it allows each cluster to be processed independently so that its dimensions can be estimated more precisely by means of an appropriate algorithm that can fit rectangles to noisy data, which is illustrated for an individual SA in Fig. 3d. The dimensions of the fitted rectangle will provide the dimensions of the SA (frequency bandwidth and start/end times). The process can be repeated individually

for each detected cluster (SA). The presence of errors in the received spectrogram \mathbf{B} requires a careful consideration in the design of the spectrum clustering and SAE steps, which are discussed in more detail in Sections III and IV, respectively.

III. SPECTROGRAM CLUSTERING

A. Selection of the Clustering Algorithm

The DBSCAN algorithm [26] is selected for SA clustering. DBSCAN is a density-based clustering algorithm with some desirable features that make of it a suitable choice for SA clustering in radio spectrograms. Concretely, DBSCAN can:

- Determine automatically the number of clusters in a data set, while other methods require a pre-set number of clusters as an input parameter (e.g., k -means). This feature is appealing as the number of SAs that may be present in a spectrogram is not known beforehand.
- Identify outliers not belonging to any clusters, while some other clustering methods classify all data points into a cluster. This feature is useful to isolate false alarms.
- Recognise arbitrarily sized and shaped clusters. Other methods implicitly assume or are more sensitive to certain cluster shapes, such as circular shapes for k -means or elliptical shapes for Gaussian mixture models, which are not compatible with the rectangular shape of SAs.

It is worth noting that, in its most basic form, DBSCAN has a worst-case run-time complexity of $\mathcal{O}(n^2)$ and an associated memory complexity of $\mathcal{O}(n)$. Nevertheless, a distance matrix of size $(n^2 - n)/2$ can be employed to avoid distance recalculations. Then, using an index structure that executes a neighbourhood search in logarithmic time, the overall time complexity can be reduced to $\mathcal{O}(n \log n)$ at the expense of an increased memory complexity of $\mathcal{O}(n^2)$ [32].

B. Configuration of the Clustering Parameters

DBSCAN has two parameters. The first parameter is the *neighbourhood radius* ε , which defines the neighbourhood of a data point as the set of points that are within a distance ε of such point. The second parameter is the *clustering threshold* δ , which determines the minimum number of points in the ε -neighbourhood of a data point (including the point itself) for it to be considered as a *core point* of a cluster; points whose neighbourhood size is below this threshold are labelled either as *noise* (outliers not belonging to any clusters) if the neighbourhood size is one (i.e., only the point itself) or *border points* otherwise. The algorithm iterates over all data points ($b_{m,n} = 1$), labelling them as noise/outliers ($c_{m,n} = -1$) or cluster members ($c_{m,n} = \tilde{k}$), until all points are labelled.

A critical aspect in the performance of DBSCAN is the configuration of its parameters $\varepsilon, \delta \in \mathbb{N}^+$. Ester et al. proposed in their original work [26] some generic strategies where the selection of ε is based on the generation of k -distance graphs for the dataset while the value for δ is lower bounded by the number of dimensions of the input data plus one. This section presents design criteria for ε and δ specifically envisaged for the clustering of SA points in noisy radio spectrograms.

Regarding the configuration of the neighbourhood radius ε , the first relevant aspect is the metric employed to measure

distances between spectrogram data points. The commonly used Euclidean distance evaluates points within circular neighbourhoods around each data point. Since SAs are by definition rectangularly shaped, a more natural metric choice is the Chebyshev distance (or Minkowski distance of infinite order), which calculates the distance between two Cartesian points $A(m_a, n_a)$ and $B(m_b, n_b)$ in the spectrogram as the maximum coordinate difference: $D(A, B) = \max(|m_a - m_b|, |n_a - n_b|)$. In a discrete two-dimensional space as the one defined for spectrograms, a Chebyshev circle of radius ε corresponds to an Euclidean square with total side length $2\varepsilon + 1$. This distance metric therefore provides a morphological match between the shapes of the ε -neighbourhoods and those of the SAs, which ensures that corner regions in SAs are given due consideration.

Another relevant aspect about ε is the criterion employed to select its value. In an ideal scenario without noise (no false alarms) and with infinite SNR (no signal missed detections), ε could take any value up to an upper bound given by the minimum possible geometric distance between SAs in the spectrogram. Such upper bound is given by either the minimum off/idle period duration of the transmitters in the time axis of the spectrogram (which, if unknown, can be estimated as discussed in [33]) or the guard band between adjacent channels in the frequency axis. For any ε beyond this limit, the ε -neighbourhood of a point in an SA could contain points from other SAs. In practice, however, the presence of false alarms and missed detections in the spectrogram points may impose more strict constraints on the selection of ε . The individual impact of false alarms and missed detections on the appropriate choice of ε is discussed separately below.

1) *Impact of False Alarms*: False alarms occur in spectrogram points where a signal component is not present but the employed signal detection method outputs a busy decision, implying that a signal component is believed to be present. False alarms are caused by noise, which is an uncorrelated process, and therefore appear in independent points (i.e., not clustered) which can in general be isolated by DBSCAN. However, false alarms can be problematic when they appear in the neighbourhood of SAs, where they can potentially be included in the clustering process, thus being incorrectly classified as members of an SA. An excessive number of false alarms clustered into an SA may affect the ability of the subsequent SAE stage (discussed in Section IV) to correctly identify the true dimensions of the SA. Thus, the radius ε should be selected to guarantee, with a sufficiently high probability, that the maximum number of false alarms that can be tolerated per neighbourhood area is not exceeded.

The number of false alarms within an ε -neighbourhood is random and can be modelled with a binomial distribution. Accordingly, the selected radius ε should meet the condition:

$$P(X \leq N_{fa}^{\text{tol}}) = I_{1-P_{fa}}(N_{fa}^{\text{max}} - N_{fa}^{\text{tol}}, N_{fa}^{\text{tol}} + 1) > \rho_{fa}, \quad (3)$$

where X represents the (random) number of observed false alarms, $I_x(a, b)$ is the regularised lower incomplete beta function [34, eq. 6.6.2] (which characterises the cumulative distribution function of binomial random variables), P_{fa} is the false alarm probability of the signal detection process at

every point of the spectrogram (*success* probability in the binomial distribution), N_{fa}^{tol} is the (maximum) number of false alarms within a neighbourhood of radius ε that can be tolerated by the SAE process (number of *successes* in the binomial distribution), N_{fa}^{max} is the maximum possible number of false alarms that can exist within the selected ε -neighbourhood (number of Bernoulli trials) and ρ_{fa} is the desired minimum probability that N_{fa}^{tol} is not exceeded in any ε -neighbourhood.

Notice that, by definition, false alarms can only occur outside SAs. Therefore, the value of N_{fa}^{max} corresponds to the number of points within the ε -neighbourhood that fall outside the SA, which depends not only on the radius ε but also on the particular location of the considered point within the SA. Clustering errors are likely to occur in regions close to the border between cluster/SA points and noise/false alarm points. Thus, special attention needs to be paid to edge and corner points. As illustrated in Fig. 4, for points in the edge and corner of an SA the maximum number of false alarms is (intermediate values are possible for points at other locations, e.g., halfway between edge/corner points):

$$N_{fa}^{max} = \begin{cases} 2\varepsilon^2 + \varepsilon, & \text{for edge points,} \\ 3\varepsilon^2 + 2\varepsilon, & \text{for corner points.} \end{cases} \quad (4a)$$

The worst-case scenario corresponds to corner points as here is where more false alarms may be found in the ε -neighbourhood of a point that belongs to an SA. Thus, a worst-case scenario design should select ε so as to meet the condition in (3) for corner points (this design criterion for the neighbourhood radius will be referred to as *corner point design*). However, there are only four corner points in every SA so their relevance in the SA clustering process may not be as significant as that of edge points, whose number can in general be expected to be much larger. Hence, the radius ε may alternatively be chosen to meet the condition (3) for edge points, which will provide a less restrictive constraint on the allowable radius ε (this will be referred to as *edge point design*).

Given a desired minimum ρ_{fa} , an upper bound for ε can be obtained by inverting numerically the LHS of the inequality in (3) with respect to N_{fa}^{max} and then solving (4), which yields:

$$\varepsilon \leq \begin{cases} \frac{1}{4} \left(\sqrt{1 + 8N_{fa}^{max}} - 1 \right), & \text{for edge points,} \\ \frac{1}{3} \left(\sqrt{1 + 3N_{fa}^{max}} - 1 \right), & \text{for corner points.} \end{cases} \quad (5a)$$

It is worth noting that the calculation of (5) requires the knowledge of the false alarm probability of the signal detection process at every spectrogram point (P_{fa}). False alarms are solely caused by the receiver's noise, which can be estimated accurately by using appropriate methods [35], [36] and can thus be assumed to be known in practical implementations.

2) *Impact of Missed Detections*: The choice of radius ε not only affects the rejection of false alarms but also the correct clustering of SA points. As the SNR decreases, the probability of detection will decrease as well and, as a result, the density of busy points observed within the existing SAs will decrease, thus making the clustering process more challenging. This can be compensated to some extent by increasing ε in order to increase the probability of having sufficient SA points within

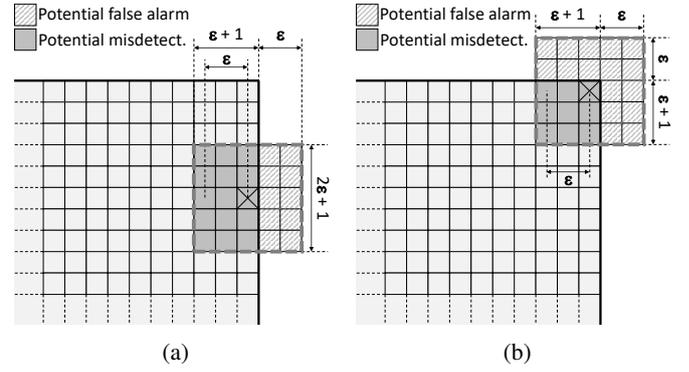


Fig. 4: Neighbourhood of radius ε for spectrogram data points (\times) at different locations within the SA: (a) edge, and (b) corner.

the ε -neighbourhood (thus improving the sensitivity of the clustering process) and then adjusting δ accordingly as needed. To correctly classify busy points that belong to an SA as core points, the radius ε should be selected to guarantee, with a sufficiently high probability, that the number of SA points observed within a neighbourhood of radius ε is greater than or equal to the employed clustering threshold δ .

The number of true SA points within an ε -neighbourhood can be characterised with a binomial distribution and therefore the probability to observe a minimum of δ points within that ε -neighbourhood is determined by its complementary cumulative distribution function. Accordingly, the selected radius ε should meet the condition [34, eq. 6.6.3]:

$$P(Y \geq \delta) = 1 - I_{1-P_d}(N_{sd}^{max} - \delta, \delta + 1) = I_{P_d}(\delta + 1, N_{sd}^{max} - \delta) > \rho_{sd}, \quad (6)$$

where Y is the (random) number of true SA points, P_d is the detection probability of the signal detection process at every point of the spectrogram (*success* probability in the binomial distribution), δ is the (minimum) number of signal detections within a neighbourhood of radius ε required to declare the central point as a core point of a cluster/SA (number of *successes* in the binomial distribution), N_{sd}^{max} is the maximum possible number of signal detections that can exist within the ε -neighbourhood (number of Bernoulli trials in the binomial distribution) and ρ_{sd} represents the desired minimum probability that δ is exceeded in every ε -neighbourhood.

The value of N_{sd}^{max} corresponds to the number of points within the ε -neighbourhood that fall inside the SA. As illustrated in Fig. 4, for points in the edge and corner of an SA the values are (intermediate values are possible for points at other locations as discussed earlier):

$$N_{sd}^{max} = \begin{cases} (2\varepsilon + 1)(\varepsilon + 1), & \text{for edge points,} \\ (\varepsilon + 1)^2, & \text{for corner points.} \end{cases} \quad (7a)$$

Corner points represent again the worst-case scenario since here is where less signal detections may be found within the ε -neighbourhood of a point that belongs to an SA. Consequently, a worst-case scenario design should select ε so as to meet the condition in (6) for corner points. As discussed earlier, the radius ε may alternatively be chosen so as to meet the

condition in (6) for edge points, which will in general provide a less restrictive constraint on the allowable radius ε .

Given a desired minimum ρ_{sd} , a lower bound for ε can be obtained by inverting numerically the LHS of (6) with respect to N_{sd}^{\max} and then solving (7), which yields:

$$\varepsilon \geq \begin{cases} \frac{1}{4} \left(\sqrt{1 + 8N_{sd}^{\max}} - 3 \right), & \text{for edge points,} \\ \sqrt{N_{sd}^{\max}} - 1, & \text{for corner points.} \end{cases} \quad (8a)$$

The calculation of (8) requires the knowledge of the detection probability of the signal detection process at every spectrogram point (P_d), which in turn requires the knowledge of the SNR for almost every signal detection method. For instance, for an energy detector, which is one of the most commonly used signal detection methods [37], [38], the detection probability P_d for a constant false alarm rate P_{fa} can be obtained as [39, eqs. (1) and (3)]:

$$P_d = Q \left(\frac{Q^{-1}(P_{fa}) - \sqrt{L}\gamma}{1 + \gamma} \right), \quad (9)$$

where $Q(\cdot)$ and $Q^{-1}(\cdot)$ represent the Gaussian Q-function [34, eq. 26.2.3] and its inverse, respectively, L is the sample size employed in the signal detection process at every spectrogram point and γ is the SNR. The knowledge of the actual SNR may or may not be available (or not to a sufficient level of accuracy) in practical implementations. If it is available, then the lower bound in (8) can be optimised for every experienced SNR; otherwise, the calculation can be based on optimistic (e.g., $P_d \approx 0.9$) and pessimistic (e.g., $P_d \approx 0.5$) assumptions to determine a practical range of suitable values for ε . This aspect will be explored in more detail in Section VI.

3) *Proposed Design Approach*: From the analysis presented above, it is evident that reducing ε decreases the number of false alarms that can potentially fall within the neighbourhood of an SA and enhances the robustness of the clustering process against false alarms at the expense of a worse sensitivity to true SA points under degraded SNR conditions. Conversely, increasing the radius ε improves clustering of true SA points but also increases the probability that false alarms are included in an SA cluster. The expressions shown in (5) and (8) provide an interval of values for ε where the selected radius should lie to ensure a satisfactory performance. These expressions require the previous evaluation of (3) and (6), which in turn requires two pairs of input parameters to be specified, namely $(\rho_{fa}, N_{fa}^{\text{tol}})$ and (ρ_{sd}, δ) , respectively.

A reasonable choice for the probability parameters ρ_{fa} and ρ_{sd} can be any sufficiently high value (e.g., 0.90, 0.95 or 0.99).

The choice of N_{fa}^{tol} is relatively simple as its value should always be kept low to ensure that the subsequent stage of the proposed SAE method (discussed in Section IV) can correctly identify the true dimensions of the SA. Fig. 5 shows the maximum value of the radius ε as a function of the desired N_{fa}^{tol} . The curves are shown for $\rho_{fa} \in \{0.90, 0.95, 0.99\}$ when the radius ε is calculated assuming that N_{fa}^{\max} corresponds to the maximum number of SA points within the neighbourhood of edge/corner SA points (i.e., edge/corner point designs, respectively). It is worth noting that small variations of N_{fa}^{tol}

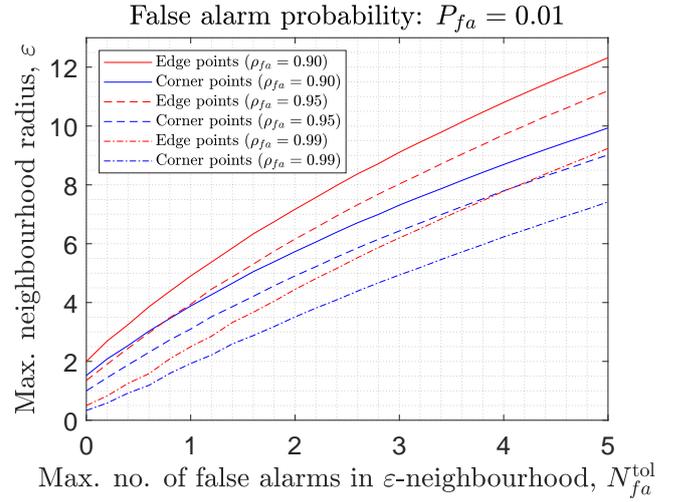


Fig. 5: Maximum neighbourhood radius ε as a function of the maximum desired number of false alarms N_{fa}^{tol} ($P_{fa} = 0.01$).

do not result in large variations of ε (for instance, moving from $N_{fa}^{\text{tol}} = 2$ to $N_{fa}^{\text{tol}} = 3$ increases ε by only two units). This suggests that, for spectrograms with sufficiently high time and frequency resolutions [39], the particular choice of N_{fa}^{tol} should not have a significant impact on the false alarm rejection performance as long as it is kept relatively low (e.g., around the range of values shown in Fig. 5).

The choice of δ requires a more careful consideration given that this is one of the DBSCAN configuration parameters and it can determine critically the performance and accuracy of the clustering process. As mentioned above, the value of N_{sd}^{\max} denotes the maximum possible number of signal detections within the considered ε -neighbourhood and therefore represents a practical upper bound to the clustering threshold, i.e. $\delta \leq N_{sd}^{\max}$. If the clustering threshold is set at the extreme value $\delta = N_{sd}^{\max}$, then the clustering process requires that every single SA point within an ε -neighbourhood that can be detected must be detected in order to meet the clustering requirements, which would only occur in an ideal scenario with no signal missed detections (i.e., infinite or sufficiently high SNR). In a practical context, a more reasonable choice would be $\delta = \lfloor P_d N_{sd}^{\max} \rfloor$, which corresponds to the average number of SA points that would actually be observed within the ε -neighbourhood taking into account that some SA points will be missing due to signal missed detections, which can occur with probability $1 - P_d$. Notice that this latter choice of δ requires the knowledge of the SNR to calculate P_d .

Taking into account the observations above, the following design approach is proposed for the configuration of the DBSCAN parameters. First, select the desired value of N_{fa}^{tol} and calculate the corresponding radius according to (3)–(5). This is the maximum radius that will provide the desired false alarm rejection performance. Since the sensitivity to true SA points increases with the employed radius, select the maximum possible value allowed by the upper bound given by (5) as the neighbourhood radius ε (notice that this procedure does not require the knowledge of the SNR to set the neighbourhood

radius). Once the radius ε is set, the clustering threshold can be obtained following two possible methods:

- Method 1: Calculate N_{sd}^{\max} based on (7), introduce this value in (6) and invert numerically the LHS of (6) with respect to δ obtain the clustering threshold.
- Method 2: Calculate the clustering threshold directly as $\delta = \lfloor P_d N_{sd}^{\max} \rfloor$, where N_{sd}^{\max} is obtained from (7).

In both cases, the calculation of δ requires a value for P_d , which can be either its true value if it can be calculated (i.e., if the SNR is known) or a set of pessimistic/optimistic values (e.g., $P_d \approx 0.5/0.9$) if the SNR is unknown. In summary, this configuration procedure first sets the neighbourhood radius ε so as to achieve the desired false alarm rejection performance and then tunes the clustering threshold δ to provide a suitable level of sensitivity to true SA points for the selected radius ε .

Fig. 6 compares the clustering threshold obtained with both methods as a function of the SNR (assuming SNR known) for various N_{fa}^{tol} and edge/corner point designs. The largest variations in the calculated thresholds are due to the consideration of the edge or corner point designs. The selected method also has, to a lesser extent, an impact on the resulting clustering threshold. Method 1 yields lower thresholds than Method 2, which can be explained by the fact that Method 1 sets the threshold to guarantee a high probability that the threshold is exceeded (ρ_{sd}), while Method 2 sets the threshold based on the average number of SA points (disregarding ρ_{sd}) and thus can be expected to embrace a lower proportion of the distribution. An interesting feature of both methods is the ability to adjust the threshold according to the experienced SNR. When the SNR decreases, the density of points within SAs decreases as a result of more frequent signal missed detections, which is compensated by reducing the threshold, thus requiring a lower number of points in the neighbourhood to satisfy the clustering requirements, hence improving the sensitivity. When the SNR increases, the density of SA points increases as well and the threshold is then increased to make the clustering process more robust to outliers. For sufficiently large SNR, the threshold converges to $\delta = \lfloor P_d N_{sd}^{\max} \rfloor = N_{sd}^{\max} - 1$ for both methods, which is the highest practical value ($P_d < 1$). Both methods will be analysed in more detail in Section VI.

IV. SIGNAL AREA ESTIMATION

According to the system model presented in Section II, the clustering stage discussed in Section III outputs a set of \tilde{K} clusters found in the spectrogram, each of them comprising a set of points $\mathcal{C}_{\tilde{k}} = \{(m, n) : c_{m,n} = \tilde{k}\} \forall \tilde{k} \in \{1, \dots, \tilde{K}\}$. Assuming that the clustering stage performed well and the true number of clusters/SAs is correctly recognised ($\tilde{K} = K$), which, as it will be shown later on, is true as long as the SNR is greater than a certain *sensitivity* level, then each set $\mathcal{C}_{\tilde{k}}$ can be seen as a *noisy* version of a true SA \mathcal{S}_k and therefore can be used to produce estimates of the SA's parameters $(\tilde{m}_k^{(l)}, \tilde{m}_k^{(h)}, \tilde{n}_k^{(l)}, \tilde{n}_k^{(h)}) \forall k \in \{1, \dots, \tilde{K}\}$ by using an appropriate algorithm that can fit rectangles to noisy data.

A common approach to fit models to noisy data points is by minimising the sum of squared errors. When fitting geometric models (e.g., rectangles), the error is usually quantified in

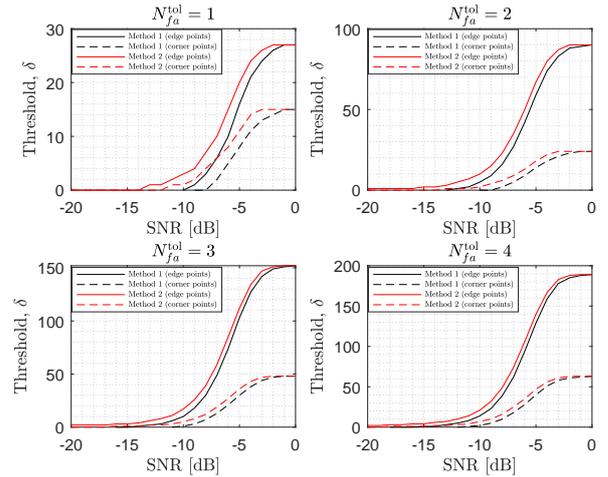


Fig. 6: Clustering threshold δ as a function of the SNR for $N_{fa}^{\text{tol}} \in \{1, 2, 3, 4\}$ ($N=100$, $P_{fa}=0.01$, $\rho_{fa}=\rho_{sd}=0.95$).

terms of the Euclidean distance from the points to the model (more sophisticated error metrics can be obtained from a more rigorous application of Bayesian principles, see for instance [40, Sect. 2.5] for the case of axis-aligned rectangles). Unconstrained minimisation can provide good results if data points are distributed along the edges of a rectangle, but fails when data points are scattered over the whole area of the rectangle as it is the case of the problem here considered. To resolve this issue, [41] introduces some *interior requirement* constraints to the optimisation problem to ensure that points are contained inside the fitted rectangle. However, this usually leads to larger rectangles than the desired one when false alarms in the proximity of the SA are clustered along with true SA points. To make the optimisation robust to noise, [41] softens the interior requirement constraints by introducing slack variables (one for each data point) along with an additional slackness parameter. The resulting optimisation problem [41, eq. (8)] is in general computationally expensive and needs some manual parameter tuning (e.g., for the slackness parameter).

In contrast, this section proposes a significantly simpler heuristic method that exploits several features unique to the SAE problem here considered: 1) true SA points are contained within axis-aligned rectangles (i.e., rectangles whose edges are parallel to the horizontal and vertical axes of the spectrogram); 2) the coordinate system is discrete and therefore each true SA point can only be placed in a finite set of locations within the true SA defined by a finite range of row/column indices; 3) the number of data points in any row/column that belongs to a true SA should be greater (or at least not lower) than in any row/column that falls outside. Based on these observations, a simple method is proposed, which is described below.

First, calculate for each cluster set $\mathcal{C}_{\tilde{k}}$ the Minimum-Area Bounding Rectangle (MABR) that contains all its points. Several sophisticated methods have been proposed in the literature to calculate MABRs [42]–[44]. However, based on observation 1 above, the calculation in this case is trivial since the MABR is obtained by simply calculating the minimum and maximum

horizontal and vertical coordinates of the data set $\mathcal{C}_{\tilde{k}}$, hence: $m_{\tilde{k}}^{\min} = \min_m(\mathcal{C}_{\tilde{k}})$, $m_{\tilde{k}}^{\max} = \max_m(\mathcal{C}_{\tilde{k}})$, $n_{\tilde{k}}^{\min} = \min_n(\mathcal{C}_{\tilde{k}})$, $n_{\tilde{k}}^{\max} = \max_n(\mathcal{C}_{\tilde{k}})$. The MABR determined by these points will in general be larger than the true SA due to false alarms in the proximity of the SA that have been included in the set during the clustering stage. In order to determine the correct edges of the SA, the elements of two vectors $\alpha_{\tilde{k}} = [\alpha_{\tilde{k},m}] \in \mathbb{N}^{(m_{\tilde{k}}^{\max}-m_{\tilde{k}}^{\min}+1) \times 1}$ and $\beta_{\tilde{k}} = [\beta_{\tilde{k},n}] \in \mathbb{N}^{1 \times (n_{\tilde{k}}^{\max}-n_{\tilde{k}}^{\min}+1)}$ are calculated (based on observation 2 above) as follows:

$$\alpha_{\tilde{k},m} = \sum_{n=n_{\tilde{k}}^{\min}}^{n_{\tilde{k}}^{\max}} \mathbf{1}_{\mathcal{C}_{\tilde{k}}}\{(m,n)\}, \quad \forall m \in [m_{\tilde{k}}^{\min}, m_{\tilde{k}}^{\max}] \quad (10)$$

$$\beta_{\tilde{k},n} = \sum_{m=m_{\tilde{k}}^{\min}}^{m_{\tilde{k}}^{\max}} \mathbf{1}_{\mathcal{C}_{\tilde{k}}}\{(m,n)\}, \quad \forall n \in [n_{\tilde{k}}^{\min}, n_{\tilde{k}}^{\max}] \quad (11)$$

where $\mathbf{1}_{\mathcal{A}}\{x\}$ is the indicator function of subset \mathcal{A} , which is equal to one for the elements $x \in \mathcal{A}$ and zero otherwise. The elements of these vectors indicate how many data points are contained in each of the rows/columns within the calculated MABR. Based on observation 3 above, the value of the central elements of vectors $\alpha_{\tilde{k}}$ and $\beta_{\tilde{k}}$ should be noticeably greater than the first and last few elements, since the central elements will correspond to rows/columns within the body of the true SA (where the density of points should be higher), while the first and last elements will correspond to rows/columns outside the SA (where the density of points should be lower) and their value is related to the number of false alarms observed in the vicinity of the true SA. Thus, the edges of the true SA can be estimated by setting a proper threshold such that only the central elements are greater than the threshold and discarding the elements below the threshold, which should be those in the few first/last positions of vectors $\alpha_{\tilde{k}}$ and $\beta_{\tilde{k}}$. This is equivalent to reduce the area of the calculated MABR by discarding those outer points that are likely to be just false alarms and not true SA points. Therefore, the SA edges can be estimated as:

$$\begin{aligned} \tilde{m}_{\tilde{k}}^{(l)} &= \min m : \alpha_{\tilde{k},m} \geq \eta_{\alpha}, & \tilde{m}_{\tilde{k}}^{(h)} &= \max m : \alpha_{\tilde{k},m} \geq \eta_{\alpha}, \\ \tilde{n}_{\tilde{k}}^{(l)} &= \min n : \beta_{\tilde{k},n} \geq \eta_{\beta}, & \tilde{n}_{\tilde{k}}^{(h)} &= \max n : \beta_{\tilde{k},n} \geq \eta_{\beta}. \end{aligned}$$

where η_{α} and η_{β} can be calculated with Otsu's algorithm [45].

Note that the operations involved in the SAE stage (searching the minimum and maximum values of arbitrary arrays and counting their numbers of elements) have a run-time complexity of $\mathcal{O}(n)$. Therefore, the overall run-time complexity of the proposed method is dominated by the complexity of the DBSCAN algorithm used in the spectrogram clustering stage, which was discussed in Section III-A. Moreover, notice that the proposed SAE method does not require any manual parameter adjustment to the data set and, owing to its heuristic nature, is significantly less computationally costly than optimisation-based approaches. Despite not being based on optimisation formulations, the proposed method provides virtually perfect accuracy as it will be shown in Section VI.

TABLE I: Simulation parameters.

Parameter	Value
Spectrogram resolution ($M \times N$)	250×500 points
Number of RF channels	1
Channel bandwidth	1/3 of spectrogram frequency span
Channel guard band	5% of channel bandwidth
ON / OFF model	Exponential durations ON periods rate parameter = 0.1 OFF periods rate parameter = 0.1
Minimum ON / OFF duration	50 / 25 spectrogram points
P_{fa} / P_d probabilities	$P_{fa} = 0.01 / P_d$ as in eqn. (9)
Signal detection sample size (L)	100 samples
SNR range ($10 \log_{10} \gamma$)	[-20, 0] dB in 0.5 dB increments
No. spectrograms per SNR value	100
ρ_{fa} / ρ_{sd} probabilities	$\rho_{fa} = \rho_{sd} = 0.95$

V. PERFORMANCE EVALUATION METHODOLOGY

A. Software Simulator

Monte Carlo simulations were carried out with the parameters shown in Table I and following a procedure similar to that employed in [39], [46], which is summarised below.

Step 1. Generate ground truth spectrograms: For each simulated SNR, a set of 100 random spectrograms \mathbf{T} were generated with a resolution of 250×500 points. Each spectrogram was generated to include the equivalent to three RF channels with 5% guard bands, where only the central channel was in use and carried traffic characterised by a set of ON/OFF transmissions randomly drawn from exponential distributions with the parameters shown in Table I.

Step 2. Generate received spectrograms: For each spectrogram generated in Step 1, random errors were introduced to produce the corresponding matrix \mathbf{B} observed at the receiver. Concretely, idle (busy) elements in \mathbf{T} were changed to busy (idle) points in \mathbf{B} with probability P_{fa} (or $1-P_d$), respectively. These probabilities were calculated assuming that the decision threshold λ was set for a constant false alarm rate [31] equal to $P_{fa} = 0.01$, such that P_d depends on the experienced SNR as given by (9), where signal detection was assumed to be performed based on blocks of $L = 100$ signal samples.

Step 3. Apply the proposed method and assess its accuracy: The proposed method was applied to each spectrogram \mathbf{B} obtained in Step 2 and the output was compared to the original ground truth spectrogram \mathbf{T} to assess the estimation accuracy.

B. Experimental Prototype

The experimental prototype shown in Fig. 7 was employed in this work to corroborate the simulation results and provide a realistic performance validation. The platform was composed of a Signal Hound VSG25A vector signal generator (1) as the signal transmitter, a 4-inch long, 0.141-inch diameter Mini-Circuits HandFlex 141-4SM+ coaxial cable (2) along with a Mini-Circuits VAT-20+ 20 dB attenuator (3) as the propagation channel, and a Tektronix RSA306B real-time spectrum analyser (4) as the signal receiver or spectrum monitoring device. A wired connection was employed to avoid unwanted interference to/from other wireless devices operating in the vicinity of

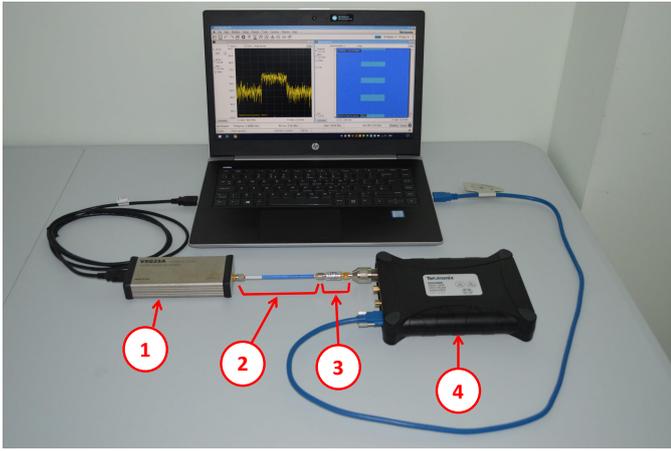


Fig. 7: Experimental prototype used in this work: (1) vector signal generator, (2) coaxial cable, (3) attenuator, and (4) spectrum analyser.

the prototype. The transmitter and receiver were connected via USB to a single computer running a Matlab-tailored control program communicating simultaneously with both. A single centralised control program/computer was required to enable adequate time synchronisation between the transmitter and the receiver operation, which is essential to accurately evaluate the performance of the proposed method by comparing the result of processing spectrogram \mathbf{B} observed at the receiver to the ground truth \mathbf{T} at the transmitter (i.e., each row of the spectrogram at the transmitter and the receiver must correspond to the same time-instant/time-sweep to allow a correct comparison). To communicate with the transmitter, a separate C-tailored program was developed based on the library and Application Programming Interface (API) provided by the manufacturer and invoked from the Matlab control program to turn the transmitter's RF stage on and off at appropriate times so as to emulate an intermittent transmission with the traffic statistical properties shown in Table I. To communicate with the receiver, the control program used Matlab's Instrument Control Toolbox along with the manufacturer's provided library and API.

The experimental platform was configured to replicate the simulation configuration shown in Table I as closely as possible and according to the capabilities of the employed equipment. The transmitted signal was a multi-tone signal¹ composed of 1001 unmodulated tones with random phase spaced at 10 kHz around a central frequency of 1 GHz, with a total signal bandwidth of 10 MHz. The centre frequency of the receiver was also set to 1 GHz with a frequency span of 30 MHz (so that the signal bandwidth is 1/3 of the spectrogram frequency span) and the resolution bandwidth was set to 10 kHz following the recommendations provided in [31]. Due to configuration restrictions of the spectrum analyser, the spectrogram resolution in the experiments was

¹The power spectral density of the transmitted multi-tone signal has a square shape (similar to that of an OFDM signal) with abrupt signal passband transitions, which facilitates an accurate identification of the signal bandwidth in the experimental data. This feature is also important to set the ground truth \mathbf{T} against which the received spectrograms \mathbf{B} are compared in order to evaluate experimentally the SAE accuracy. Concretely, this ensures that the estimated SA *widths* are compared to the correct signal bandwidth values.

set to 500×901 points. Increasing the resolution employed in the simulations to match these values was not possible due to excessive computational workload. To solve this problem, the experimental spectrograms were decimated to achieve the same resolution used in simulations (i.e., 250×500 points).

Based on noise power measurements, the decision threshold employed to binarise the power levels of the received spectrogram was calculated individually for each frequency point as indicated by [31, eq. (3)] to ensure a constant false alarm probability of 0.01 as in Table I.

The relation between the transmission power configured at the signal generator and the SNR observed at the spectrum analyser was carefully calibrated to enable a fair comparison between simulation and experimental results.

C. Performance Metrics

The resulting estimation accuracy was assessed by comparing the output of the proposed method to the ground truth spectrogram \mathbf{T} , both in simulations and experiments. The comparison was made based on the F1 score, defined as [47]:

$$\text{F1 score} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \in [0, 1] \quad (12)$$

where TP, FP and FN represent the number of true positives, false positives and false negatives, respectively. All $M \times N$ spectrogram elements were included in the calculation of (12).

The performance of the clustering stage was also assessed based on the ratio of the number of estimated SAs, \tilde{K} , to the true number of SAs in the ground truth spectrogram, K , where a value equal to one indicates a perfect estimation.

VI. PERFORMANCE EVALUATION RESULTS

A. Performance of the Spectrogram Clustering Stage Alone

This subsection evaluates the performance of the clustering stage in the proposed method. The interest of the analysis here presented is in the optimisation of the proposed method to achieve the best possible sensitivity in the lower SNR regime.

The ability of the DBSCAN clustering algorithm to discriminate true SA points from false alarms is illustrated in Fig. 8 assuming $N_{fa}^{tol} = 3$ (the impact of N_{fa}^{tol} will be discussed later on). This figure has been obtained by comparing the ground truth, noise-free spectrogram at the transmitter (containing only true SA points) with the output of DBSCAN at the receiver after discarding points classified as outliers/false alarms (containing only those points that DBSCAN believes to be true SA points) and calculating the resulting F1 score as a function of the SNR. Therefore, this result indicates how accurately the DBSCAN algorithm can discriminate each data point detected in the spectrogram in the busy state either as a member of any SA or as a false alarm. As expected, the false alarm rejection performance degrades as the SNR decreases. When the SNR is unknown (i.e., the configuration of the algorithm is based on a guess of the detection probability P_d instead of its true value), there exists a certain SNR threshold below which the accuracy degrades faster. As it can be observed in Fig. 8, the particular value of such SNR threshold depends on the value assumed for P_d as well as the considered design criterion (i.e.,

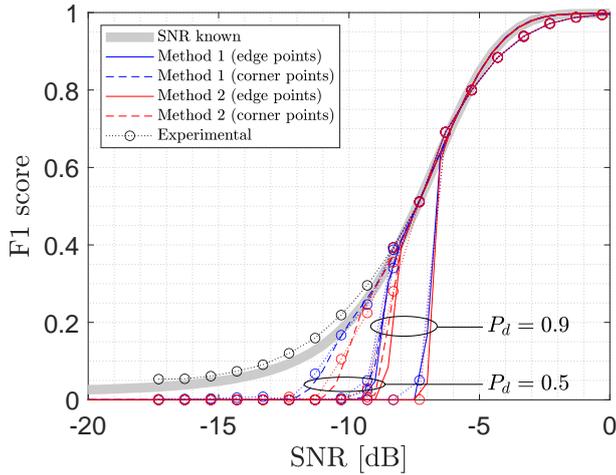


Fig. 8: DBSCAN false alarm rejection performance ($N_{fa}^{tol} = 3$).

neighbourhood radius based on edge or corner point designs and clustering threshold based on Method 1 or 2). On the other hand, when the SNR is known, this knowledge can be exploited to calculate the true value of the detection probability for the actual SNR, which provides an SNR-adaptive value for the clustering threshold (recall from Section III-B that the calculation of the neighbourhood radius is independent of the SNR); this results in the best attainable discrimination accuracy over the whole SNR range as observed in Fig. 8. Moreover, when the SNR is known, the considered design criterion (edge/corner points or Method 1/2) does not affect the F1 score, which in all cases is as shown by the thick line of Fig. 8. This suggests the need of a metric that can provide further insights into the clustering performance of DBSCAN.

As pointed out above, Fig. 8 illustrates how accurately the DBSCAN algorithm can discriminate true SA points from false alarms. However, it does not provide any information on whether the algorithm groups a data point into the correct SA when it is classified as a member of an SA. As a matter of fact, if the SNR is sufficiently low so that the detection and false alarm probabilities are comparable, then the density of true SA points and false alarm points per unit area will be similar and it will not be possible to unambiguously distinguish true SA points from false alarms. As a result, in such case a set of points originally belonging to the same SA may be grouped into a number of smaller adjacent SAs, which would result in high F1 scores even though the clustering is actually incorrect. Therefore, to correctly assess the clustering performance, it is necessary to also compare the number of SAs detected by DBSCAN to the true number of SAs in the spectrogram.

Fig. 9 illustrates the performance of the DBSCAN algorithm in terms of the number of detected SAs (\tilde{K}) normalised by the true number of SAs present in each spectrogram (K) when the SNR is assumed to be known (the case of unknown SNR will be discussed later on). A value equal to one (represented by horizontal reference lines) indicates a perfect SA classification. The minimum SNR at which a value equal to one is observed can be defined as the *sensitivity* of the clustering algorithm for a particular configuration. As it can be observed, if the SNR is

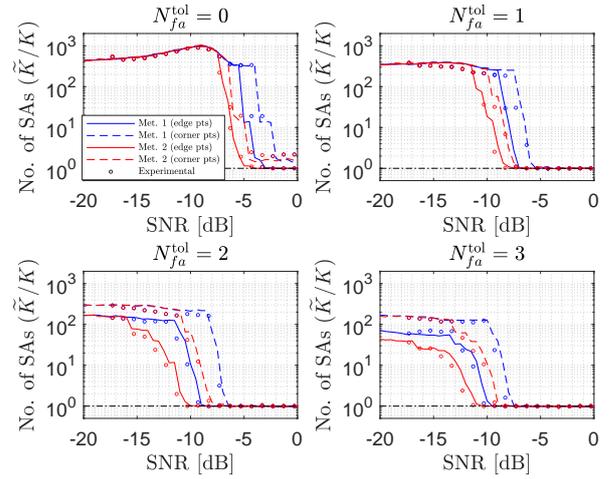


Fig. 9: Number of SAs detected by DBSCAN as a function of the SNR for various design criteria (SNR assumed to be known).

known the best sensitivity of the clustering process is always obtained when the neighbourhood radius is calculated based on edge points and the clustering threshold is calculated based on Method 2. As discussed in Section III-B, a design based on corner points assumes a worst-case scenario that leads to a smaller neighbourhood radius, while a design based on edge points leads to a larger radius. When the SNR decreases and the density of SA points per unit area decreases as well, a larger neighbourhood radius increases the chances of finding true SA points for clustering. The challenge in this case is to accurately tune the clustering threshold in order to ensure that the clustering condition is satisfied mostly when enough true SA points (rather than false alarms) are found. If the SNR is known, then this is possible with Method 2, which calculates a value for the clustering threshold closer to the true number of SA points that, in average, will be actually found within the selected neighbourhood radius. Consequently, this design criterion (based on edge points² and Method 2) results in the best sensitivity when the SNR is known.

The results shown in Fig. 9 correspond to various values of the N_{fa}^{tol} parameter, which represents the maximum number of false alarms that can be tolerated within the selected neighbourhood radius. The selection of a value for this parameter is the first step in the design approach proposed in Section III-B, therefore a natural question is what is the optimum choice for this parameter. One may feel tempted to choose $N_{fa}^{tol} = 0$ with the hope of minimising the number of false alarms involved in the clustering process. However, while this would be true, this choice does not result in the best sensitivity. The reason is that by setting very low values of N_{fa}^{tol} , the proposed design approach will select equally low neighbourhood radii (see Fig. 5), which will not only reduce the number of clustered false

²It is worth noting that corner points may still be correctly clustered even if the design is based on edge points. In such case, they may be first classified as (in DBSCAN terminology) *border points* or *noise* since their neighbourhood size may not exceed the (higher) clustering threshold set for edge points and later on be reclassified as points belonging to a cluster if they satisfy the clustering requirements through some other data point within the SA.

TABLE II: Sensitivity of DBSCAN (SNR assumed to be known).

N_{fa}^{tol}	Corner point design		Edge point design	
	Method 1	Method 2	Method 1	Method 2
0	∞	∞	-3 dB	-4 dB
1	-5 dB	-7 dB	-7 dB	-8 dB
2	-6 dB	-8 dB	-9 dB	-10 dB
3	-7 dB	-8 dB	-9 dB	-11 dB
4	-8 dB	-9 dB	-10 dB	-11 dB
5	-8 dB	-9 dB	-10 dB	-11 dB

alarms but also affect the ability to correctly find and cluster true SA points under low SNR conditions. By allowing a moderate number of false alarms within the neighbourhood of SAs, a larger neighbourhood radius will be used, which will increase both the number of false alarms and true SA points within the neighbourhood radius. This will result in a better clustering performance as long as the density per unit area of true SA points is larger than that of false alarms, which will be the case as long as the SNR is not excessively low. This explains why the sensitivity observed in Fig. 9 improves as the value of N_{fa}^{tol} increases within the set $\{0, 1, 2, 3\}$. However, the sensitivity cannot be increased indefinitely by increasing the value of the parameter N_{fa}^{tol} . For every configuration there will be a certain SNR value below which the density of true SA points will be lower than that of false alarms and increasing the neighbourhood radius through higher N_{fa}^{tol} values will not provide any further improvements. This can be more clearly appreciated in Table II, which shows the approximated sensitivity (with 1 dB resolution) achieved with each design criterion for different values of N_{fa}^{tol} . For the best design criterion in this case (edge point design with Method 2), any increase beyond $N_{fa}^{tol} = 3$ will not improve the sensitivity of the clustering process. Notice that selecting higher values of N_{fa}^{tol} will increase the number and the distance of false alarms clustered with true SA points, which will make more challenging the subsequent task of estimating the correct dimensions of each detected SA. Therefore, N_{fa}^{tol} should be selected as the lowest value for which the best attainable sensitivity is observed (in the example of Table II, this is -11 dB for $N_{fa}^{tol} = 3$ with edge point design and Method 2).

The discussion so far has focused on the scenario of known SNR. However, in practical implementations the SNR may be unknown. In such a case, a guess of the detection probability must be used in order to configure the clustering algorithm following the proposed design approach. The clustering performance was evaluated assuming unknown SNR and $P_d \in \{0.10, 0.50, 0.90\}$. For $P_d = 0.10$, the clustering process was observed to fail completely because the associated clustering threshold was in all cases (edge/corner point designs and Methods 1/2) too low to successfully cluster any points (most individual points were classified as independent clusters). For $P_d = 0.50$ and $P_d = 0.90$, the obtained sensitivity levels are summarised in Table III. As it can be noticed, the choice $P_d = 0.50$ provides better sensitivity than $P_d = 0.90$, which can be explained based on the false alarm rejection performance shown in Fig. 8 for both probabilities. The best

TABLE III: Sensitivity of DBSCAN (SNR unknown).

P_d	N_{fa}^{tol}	Corner point design		Edge point design	
		Method 1	Method 2	Method 1	Method 2
0.5	0	∞	∞	∞	∞
	1	∞	-8 dB	-7 dB	-7 dB
	2	-9 dB	-9 dB	-8 dB	-7 dB
	3	-9 dB	-9 dB	-8 dB	-7 dB
0.9	0	∞	∞	∞	-5 dB
	1	-7 dB	-7 dB	-6 dB	-6 dB
	2	-7 dB	-7 dB	-6 dB	-6 dB

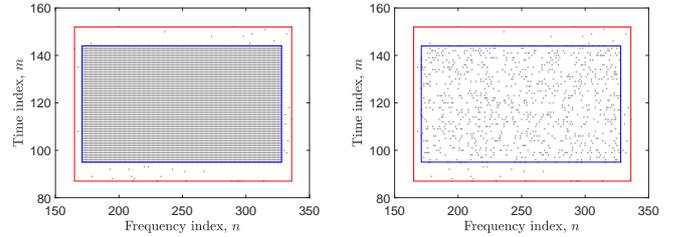
(a) SNR = 0 dB ($P_d = 0.99$) (b) SNR = -10 dB ($P_d = 0.11$)

Fig. 10: Illustrative example showing the initial MABR (red rectangle) and the SA estimated by the proposed SAE method (blue rectangle) when operating at an SNR of: (a) 0 dB, and (b) -10 dB.

sensitivity is obtained in this case for $P_d = 0.50$ and $N_{fa}^{tol} = 2$ with corner point design and both Methods 1 and 2 (Method 2 is slightly computationally simpler), which is 2 dB worse than the best sensitivity that can be achieved when the SNR is known. In this case, since the SNR is unknown, a fine tuning of the clustering threshold for larger neighbourhood radii is not possible so a corner point (worst-case) design, which uses a lower neighbourhood radius, with a middle point clustering threshold ($P_d \approx 0.5$) seems intuitively a reasonable choice and provides in this case the best attainable sensitivity. Notice that, despite the fact that the same P_d has been assumed for all the SAs (regardless of what their actual and unknown SNR may be), the overall sensitivity is reduced by only 2 dB with respect to the case of known SNR, which suggests that this is a suitable design approach to follow when the SNR is unknown.

B. Performance Including the Signal Area Estimation Stage

Fig. 10 illustrates the operation of the SAE method proposed in Section IV. As it can be noted, MABR overestimates the SA due to false alarms in the neighbourhood of the SA that are grouped along with true SA points by the clustering stage. On the other hand, the proposed SAE method effectively overcomes this issue by correctly discarding the outer points of the cluster (corresponding to false alarms) and accurately estimating the true SA dimensions, both at high (Fig. 10a) and low (Fig. 10b) SNR. It is worth noting that in the low SNR example only 11% of the SA points remain in the spectrogram and this is sufficient for the proposed SAE method to provide a highly accurate (virtually perfect) estimation of the SA.

The capability of the proposed SAE algorithm to accurately estimate the SA dimensions at such low SNR suggests that the main limiting factor in the sensitivity of the proposed method

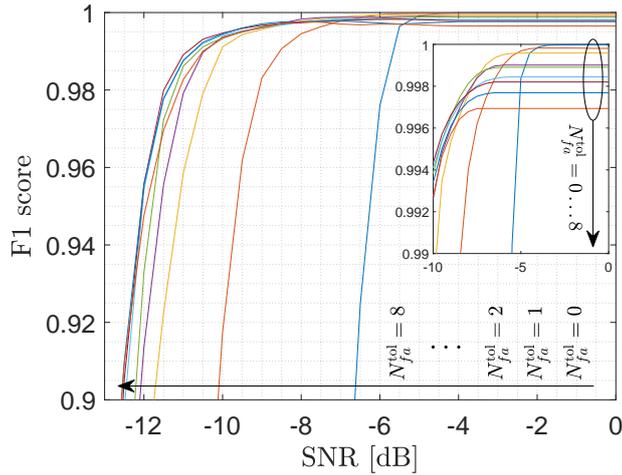


Fig. 11: Performance of the proposed method (including both stages) for various N_{fa}^{tol} (SNR known, edge point design, Method 2).

is the clustering stage. To corroborate this hypothesis, Fig. 11 shows the performance of the proposed method (including both clustering and SAE stages) for various values of N_{fa}^{tol} (these results assume that the SNR is known but the same conclusions are obtained for unknown SNR). It becomes apparent that the SNR values at which the F1 score starts to decrease in Fig. 11 for every N_{fa}^{tol} correspond very closely to the sensitivity levels of the clustering stage shown in Table II (for SNR known). The SAE stage simply estimates the likely true SA for each detected cluster, thus if the clustering stage fails to detect the correct set of clusters the subsequent SAE stage will provide incorrect results. Therefore, N_{fa}^{tol} should be selected according to the sensitivity levels of the clustering stage shown in Table II (if SNR is known) or Table III (if SNR is unknown). These tables provide an indication on the minimum N_{fa}^{tol} that should be selected but not the maximum value. To answer this question, Fig. 11 shows that the choice of N_{fa}^{tol} determines a trade-off between sensitivity at low SNR and accuracy at high SNR. However, the impact on the accuracy at high SNR is minimal (less than 0.3% as shown in the detail of Fig. 11) while the impact on the sensitivity at low SNR is much more significant as shown both in Fig. 11 and Table II. Therefore, as stated in Section VI-A, N_{fa}^{tol} should be selected as the lowest value for which the best attainable sensitivity is observed.

Fig. 12 compares the proposed method with the TECCL [27] and SSA [22] methods. For the SSA method, a false alarm reduction step was first applied and the configuration parameters were adjusted according to the guidelines provided in [23]. The F1 score for the output of the thresholding step, labelled as *Energy detection*, is included for reference as well. The other methods discussed in Section I are not included in this comparison because they require a significant amount of manual intervention such as training or manual parameter configuration, which makes the performance of such methods more subjective (it is worth noting that their authors reported a performance below that of the methods considered in this comparison). The results in Fig. 12 show that the method proposed in this work provides significant performance improvements,

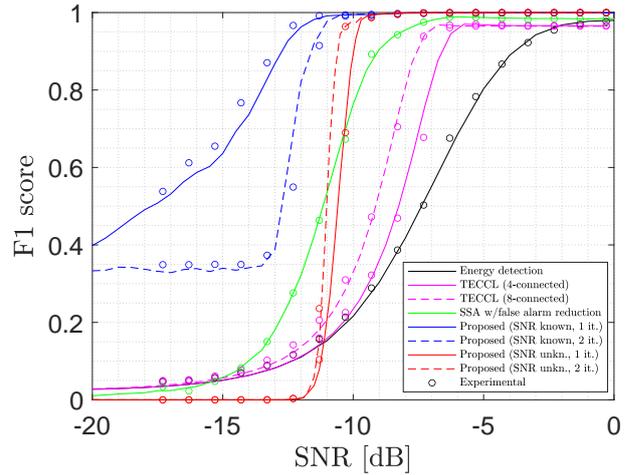


Fig. 12: Performance comparison between the method proposed in this work and other methods from the literature.

achieving virtually perfect accuracy (F1 score $\approx 100\%$) at any SNR above -11 dB if the SNR is known or -9 dB if the SNR is unknown. The proposed method not only provides the best sensitivity but is also the only one that achieves a virtually perfect accuracy. TECCL and SSA can provide accurate SAE for true signal components but are unable to fully remove false alarms, which are also reported as signal components with tiny SAs. This explains why the F1 score attained with these methods never reaches perfect accuracy, even at high SNR. On the other hand, the proposed method achieves virtually perfect accuracy at any SNR above its sensitivity threshold owing to its unique ability to effectively remove false alarms. This is a subtle but important difference when radio spectrograms are processed automatically in autonomous wireless systems since false alarms, when treated as signal components, may not only lead to incorrect decisions but also result in a waste of computational resources for any subsequent data processing steps that make use of such outcomes.

For the more likely scenario of unknown SNR, the obtained sensitivity (-9 dB) can be improved with a second iteration of the proposed method as illustrated in Fig. 13. The spectrograms shown in this figure are obtained from experimental measurements (not simulations) at an SNR of -10 dB and assuming that the SNR is unknown to the receiver. This SNR is below the sensitivity level mentioned above and the output of the proposed method (shown on the bottom left of Fig. 13) is incorrect. Compared to the ground truth (top left), the obtained spectrogram contains a larger number of smaller SAs as a result of the clustering stage failing to correctly group the data points observed at such low SNR (top right). However, the produced output suggests that these smaller SAs could be grouped into the correct number of clusters with a second iteration of the proposed method, which would then allow the correct estimation of their SAs. For this second iteration, the proposed method needs to be configured as follows:

- The neighbourhood radius (ε) should be made as large as possible to maximise the probability to cluster correctly the SA fragments produced by the first iteration (bottom

left). As discussed in Section III-B, the maximum radius should be either the minimum off/idle time of the transmitters or the channel guard band, whichever is lower.

- The clustering threshold (δ) should be made as low as possible to maximise the sensitivity of the clustering stage in the second iteration and ensure that no SA fragment is missed, even if the gap between SA fragments is so large that only a single point falls within the neighbourhood radius. Given that false alarms should have been removed in the first iteration, the value $\delta = 1$ can be selected.
- Since the output of the first iteration is expected to not contain any false alarms but just fragments of true SAs, the SAE stage in the second iteration should be based on the MABR (instead of the method proposed in Section IV) to ensure that no SA components are discarded.

This second iteration will fail if the SNR is known. In this case, the clustering stage of the first iteration has been configured to be sensitive to the true spatial density of SA points that can be expected within the neighbourhood radius, which will decrease with the SNR. When the accuracy achieved by the first iteration starts to degrade, this is because the SNR is so low that the clustering stage starts to group false alarms, thus creating clusters where no SA is actually present. In this case, the proposed second iteration will simply estimate the SA of random clusters of false alarms, which will obviously produce less accurate results than those obtained with a single iteration as it can be seen from Fig. 12. However, when the SNR is unknown, the situation is different. In this case the first iteration of the clustering stage cannot be configured to be as sensitive as when the SNR is known and as a result the accuracy will start to degrade at a higher SNR, where the spatial density of true SA points is still larger than the density of false alarms. Thus, even though the produced clusters are incorrect, they are still detected in locations where true SAs are present (i.e., the clustering stage will not start clustering false alarms) and the second iteration, when configured as detailed above, can exploit this to improve the final accuracy. As shown on the bottom right of Fig. 13, this second iteration when the SNR is unknown can improve the accuracy from 69.2% (first iteration) to 99.4% (second iteration), thus enabling accurate detections at SNR values below the sensitivity achieved with only one iteration. According to Fig. 12, this second iteration when the SNR is unknown can provide a sensitivity improvement of about 1 dB (from -9 dB to -10 dB), thus making it more similar to the sensitivity that can be achieved when the SNR is known (-11 dB). As a result, the proposed method can extract signal components from a noisy radio spectrogram with a virtually perfect accuracy at SNR as low as -10 dB, even when the SNR cannot be estimated by the receiver.

VII. CONCLUSIONS

The capability to automatically extract information about the signal components present in a radio spectrogram opens the possibility to make intelligent decisions in context-aware autonomous wireless spectrum monitoring systems. In this context, this work has proposed a method to determine the occupied bandwidth and start/end times of each radio transmission in a spectrogram. The proposed method is composed

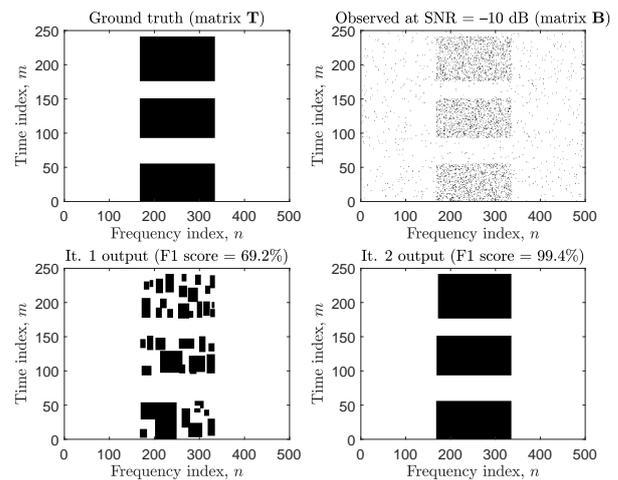


Fig. 13: Improvement of second algorithm iteration (SNR unknown).

of two stages: a spectrogram clustering stage to isolate the points belonging to each signal component and a signal area estimation stage to determine the time-frequency area occupied by each component. The performance has been assessed by means of simulations and hardware experiments, showing that the proposed method is robust to the degrading effects of noise and radio propagation and can provide a virtually perfect accuracy at signal-to-noise ratio values as low as -10 dB.

REFERENCES

- [1] Y. Jie, Y. Chenzhou, and Z. Yue, "Modulation classification based on spectrogram," *Journal of Systems Engineering and Electronics*, vol. 16, no. 3, pp. 475–488, Sep. 2005.
- [2] T. J. Lynn and A. Z. Sha'ameri, "Automatic analysis and classification of digital modulation signals using spectrogram time-frequency analysis," in *Proc. Int'l. Symp. Commun. and Information Technol. (ISCIT 2007)*, Oct. 2007, pp. 916–920.
- [3] A. Shklyayeva, P. Kovar, and D. Kubanek, "Classification of digital modulations mainly used in mobile radio networks by means of spectrogram analysis," in *Proc. 12th IFIP Int'l. Conf. Personal Wireless Commun. (PWC 2007)*, Sep. 2007, pp. 341–348.
- [4] S. Jeong, U. Lee, and S. C. Kim, "Spectrogram-based automatic modulation recognition using convolutional neural network," in *Proc. 10th Int'l. Conf. Ubiquitous Future Netw. (ICUFN 2018)*, Jul. 2018, pp. 1–3.
- [5] Y. Zeng, M. Zhang, F. Han, Y. Gong, and J. Zhang, "Spectrum analysis and convolutional neural network for automatic modulation recognition," *IEEE Wireless Comms. Letters*, vol. 8, no. 3, pp. 929–932, Jun. 2019.
- [6] W. Jiang, X. Wu, Y. Wang, B. Chen, W. Feng, and Y. Jin, "Time-frequency-analysis-based blind modulation classification for multiple-antenna systems," *Sensors*, vol. 21, no. 231, pp. 1–19, Jan. 2021.
- [7] F. A. Bhatti, M. J. Khan, A. Selim, and F. Paisana, "Shared spectrum monitoring using deep learning," *IEEE Tran. Cognitive Comm. and Networking*, vol. 7, no. 4, pp. 1171–1185, Dec. 2021.
- [8] J. M. Tarongi and A. Camps, "Radio frequency interference detection and mitigation algorithms based on spectrogram analysis," *Algorithms*, vol. 4, no. 4, pp. 239–261, Oct. 2011.
- [9] T. J. O'Shea, T. Roy, and T. Erpek, "Spectral detection and localization of radio events with learned convolutional neural features," in *Proc. 25th European Signal Process. Conf. (EUSIPCO 2017)*, Aug. 2017, pp. 331–335.
- [10] E. Almazrouei, G. Gianini, N. Almoosa, and E. Damiani, "A deep learning approach to radio signal denoising," in *Proc. IEEE Wireless Comms. and Networking Conf. (WCNC 2019)*, Apr. 2019, pp. 1–8.
- [11] X. Mankun, P. Xijian, L. Tianyun, and X. Mantian, "A new time-frequency spectrogram analysis of FH signals by image enhancement and mathematical morphology," in *Proc. 4th Int'l. Conf. Image and Graphics (ICIG 2007)*, Aug. 2007, pp. 610–615.

- [12] Y. He, Y. Su, Y. Chen, Y. Yu, and X. Yang, "Double window spectrogram difference method: A blind estimation of frequency-hopping signal for battlefield communication environment," in *Proc. 24th Asia-Pacific Conference on Communications (APCC 2018)*, Nov. 2018, pp. 439–443.
- [13] J. Kim and J. G. Andrews, "Sensitive white space detection with spectral covariance sensing," *IEEE Trans. Wireless Commun.*, vol. 9, no. 9, pp. 2945–2955, Sep. 2010.
- [14] W. M. Lees, A. Wunderlich, P. J. Jeavons, P. D. Hale, and M. R. Souryal, "Deep learning classification of 3.5-GHz band spectrograms with applications to spectrum sensing," *IEEE Trans. Cognitive Commun. and Networking*, vol. 5, no. 2, pp. 224–236, Jun. 2019.
- [15] S. Kojima, K. Maruta, Y. Feng, C.-J. Ahn, and V. Tarokh, "CNN based joint SNR and Doppler shift classification using spectrogram images for adaptive modulation and coding," *IEEE Trans. Commun.*, vol. 69, no. 8, pp. 5152–5167, Aug. 2021.
- [16] C. Hory, N. Martin, and A. Chehikian, "Spectrogram segmentation by means of statistical features for non-stationary signal interpretation," *IEEE Trans. Signal Process.*, vol. 50, no. 12, pp. 2915–2925, Dec. 2002.
- [17] C. Smith, Q. R. Black, and M. Magee, "Computer vision for improved single-sensor spectrum sensing," in *Proc. Sensor Signal Processing for Defence (SSPD 2012)*, Sep. 2012.
- [18] S. Phonsri, S. S. Mukherjee, and M. Sellathurai, "Computer vision and bi-directional neural network for extraction of communications signal from noisy spectrogram," in *Proc. IEEE Conf. Antenna Measurements and Applications (CAMA 2015)*, Dec. 2015, pp. 1–4.
- [19] X. Zha, H. Peng, X. Qin, G. Li, and S. Yang, "A deep learning framework for signal detection and modulation classification," *Sensors*, vol. 19, no. 18, pp. 1–21, Sep. 2019.
- [20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: single shot multibox detector," in *Proc. 14th European Conf. Computer Vision (ECCV 2016)*, Oct. 2016, pp. 21–37.
- [21] W. Li, K. Wang, and L. You, "A deep convolutional network for multi-type signal detection and classification in spectrogram," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–16, Sep. 2020.
- [22] K. Umabayashi, H. Iwata, J. J. Lehtomäki, and M. López-Benítez, "Study on simple signal area estimation for efficient spectrum measurements," in *Proc. 26th European Conf. Networks and Commun. (EuCNC 2017)*, Jun. 2017, pp. 1–5.
- [23] K. Umabayashi, K. Moriwaki, R. Mizuchi, H. Iwata, S. Tiirio, J. J. Lehtomäki, M. López-Benítez, and Y. Suzuki, "Simple primary user signal area estimation for spectrum measurement," *IEICE Trans. Commun.*, vol. E99-B, no. 2, pp. 523–532, Feb. 2016.
- [24] R. Mizuchi, K. Umabayashi, J. Lehtomäki, and M. López-Benítez, "A study on FFT-ED based signal area estimation for spectrum awareness," in *Proc. Int'l. Workshop on Smart Wireless Commun. (SmartCom 2016)*, IEICE Tech. Rep., vol. 116, no. 29, SR2016-9, pp. 27–34, May 2016.
- [25] R. Mizuchi, K. Umabayashi, J. J. Lehtomäki, and M. López-Benítez, "A study on false alarm cancellation for spectrum usage measurements," in *Proc. IEEE Wireless Commun. and Networking Conf. (WCNC 2017)*, 3rd Int'l. Works. Smart Spectrum (IWSS 2017), Mar. 2017, pp. 1–6.
- [26] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int'l. Conf. on Knowledge Discovery and Data Mining (KDD-96)*, Jan. 1996, pp. 226–231.
- [27] J. Kokkonen and J. Lehtomäki, "Spectrum occupancy measurements and analysis methods on the 2.45 GHz ISM band," in *Proc. 7th Int'l. ICST Conf. Cognitive Radio Oriented Wireless Networks and Commun. (CROWNCOM 2012)*, Jun. 2012, pp. 285–290.
- [28] Y. Zhou, Y. Feng, V. Tarokh, V. Gintautas, J. McClelland, and D. Garagic, "Multi-level mean-shift clustering for single-channel radio frequency signal separation," in *Proc. IEEE 29th Int'l. Works. Machine Learning for Signal Processing (MLSP 2019)*, Oct. 2019, pp. 1–6.
- [29] L. D. Stefano and A. Bulgarelli, "A simple and efficient connected components labeling algorithm," in *Proc. 10th Int'l. Conf. Image Analysis and Processing (ICIAP 1999)*, Sep. 1999, pp. 1–6.
- [30] D. W. Capson, "Performance comparisons of contour extraction algorithms," *IEEE Trans. Instrum. and Measurement*, vol. IM-35, no. 4, pp. 409–417, Dec. 1986.
- [31] M. López-Benítez and F. Casadevall, "Methodological aspects of spectrum occupancy evaluation in the context of cognitive radio," *European Trans. Telecommun.*, vol. 21, no. 8, pp. 680–693, Dec. 2010.
- [32] E. Schubert, J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "DBSCAN revisited: Why and how you should (still) use DBSCAN," *ACM Transactions on Database Systems*, vol. 42, no. 3, pp. 19:1–19:21, Jul. 2017.
- [33] M. López-Benítez, A. Al-Tahmeesschi, and D. Patel, "Accurate estimation of the minimum primary channel activity time in cognitive radio based on periodic spectrum sensing observations," in *Proc. 24th European Wireless Conf. (EW 2018)*, May 2018, pp. 131–136.
- [34] M. Abramowitz and I. A. Stegun, *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, 10th ed. Dover, 1972.
- [35] H. Iwata, K. Umabayashi, A. Al-Tahmeesschi, M. López-Benítez, and J. J. Lehtomäki, "Time and frequency varying noise floor estimation for spectrum usage measurement," in *Proc. IEEE Wireless Commun. and Networking Conf. (WCNC 2019)*, 5th IEEE Int'l. Workshop on Smart Spectrum (IWSS 2019), Apr. 2019, pp. 1–6.
- [36] M. López-Benítez, J. Lehtomäki, K. Umabayashi, and D. Patel, "Accurate noise floor calibration based on modified expectation maximisation of Gaussian mixture," in *Proc. IEEE Wireless Commun. and Networking Conf. (WCNC 2019)*, Apr. 2019, pp. 1–6.
- [37] H. Urkowitz, "Energy detection of unknown deterministic signals," *Proceedings of the IEEE*, vol. 55, no. 4, pp. 523–531, Apr. 1967.
- [38] M. López-Benítez and F. Casadevall, "Improved energy detection spectrum sensing for cognitive radio," *IET Communications*, vol. 6, no. 8, pp. 785–796, May 2012.
- [39] M. M. Alammam and M. López-Benítez, "Evaluation of the impact of thresholding and frequency/time resolution on signal area estimation methods," in *Proc. 93rd IEEE Vehic. Tech. Conf. (VTC 2021-Spring)*, 7th Int'l. Works. Smart Spectrum (IWSS 2021), Apr. 2021, pp. 1–7.
- [40] M. Werman and D. Keren, "A Bayesian method for fitting parametric and nonparametric models to noisy data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 5, pp. 528–534, May 2001.
- [41] J. Yang and Z. Jiang, "Rectangle fitting via quadratic programming," in *Proc. 17th IEEE Int'l. Workshop on Multimedia Signal Processing (MMSP 2015)*, Oct. 2015, pp. 1–6.
- [42] H. Freeman and R. Shapira, "Determining the minimum-area enclosing rectangle for an arbitrary closed curve," *Communications of the ACM*, vol. 18, no. 7, pp. 409–413, Jul. 1975.
- [43] D. Chaudhuri and A. Samal, "A simple method for fitting of bounding rectangle to closed regions," *Pattern Recognition*, vol. 40, no. 7, pp. 1981–1989, Jul. 2007.
- [44] D. Chaudhuri, N. K. Kushwaha, I. Sharif, and A. Samal, "Finding best-fitted rectangle for regions using a bisection method," *Machine Vision and Applications*, vol. 23, no. 6, pp. 1263–1271, Nov. 2012.
- [45] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, & Cyber.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.
- [46] M. M. Alammam and M. López-Benítez, "A minesweeper algorithm for improved signal area estimation in spectrum aware systems," in *Proc. 28th Int'l. Conf. Telecommun. (ICT 2021)*, Jun. 2021, pp. 1–6.
- [47] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *J. Machine Learning Technol.*, vol. 2, no. 1, pp. 37–63, Dec. 2011.



Miguel López-Benítez (Senior Member, IEEE) received the BSc and MSc degrees (both with Distinction) from Miguel Hernández University, Elche, Spain in 2003 and 2006, respectively, and the PhD degree from the Technical University of Catalonia, Barcelona, Spain in 2011. From 2011 to 2013, he was a Research Fellow with the Centre for Communication Systems Research, University of Surrey, Guildford, UK. In 2013, he became a Lecturer (Assistant Professor) with the Department of Electrical Engineering and Electronics, University of Liverpool, UK, where he has been a Senior Lecturer (Associate Professor) since 2018. His research interests are in the field of wireless communications.



Mohammed M. Alammam received the M.Sc. degree in Electrical Engineering from the University of Dayton, USA in 2016. He is a Lecturer at King Khalid University, Abha, Saudi Arabia. He is currently pursuing the Ph.D. degree at the University of Liverpool, Liverpool, UK. His research interests include image processing, signal processing and embedded systems.