Spectrum Patrolling with Crowdsourced Spectrum Sensors

Ayon Chakraborty, Arani Bhattacharya, Snigdha Kamal, Samir R. Das, Himanshu Gupta, Petar M. Djuric†
Stony Brook University
Email: {aychakrabort, arbhattachar, snkamal, samir, hgupta}@cs.stonybrook.edu, †petar.djuric@stonybrook.edu

***Abstract*—We use a crowdsourcing approach for RF spectrum patrolling, where heterogeneous, low-cost spectrum sensors are deployed widely and are tasked with detecting unauthorized transmissions in a collaborative fashion while consuming only a limited amount of resources. We pose this as a collaborative signal detection problem where the individual sensor's detection performance may vary widely based on their respective hardware or software configurations, but are hard to model using traditional approaches. Still an optimal subset of sensors and their configurations must be chosen to maximize the overall detection performance subject to given resource (cost) limitations. We present the challenges of this problem in crowdsourced settings and present a set of methods to address them. The proposed methods use data-driven approaches to model individual sensors and develops mechanisms for sensor selection and fusion while accounting for their correlated nature. We present performance results using examples of commodity-based spectrum sensors and show significant improvements relative to baseline approaches.**

I. Introduction

With growing realization of mobile communication's impact on the nation's economic prosperity, RF spectrum has emerged as an important natural resource that is in limited supply [1]. While various spectrum sharing models are being developed to improve spectrum usage, 'spectrum patrolling' to detect unauthorized spectrum use is emerging as a critical technology [2]. Such unauthorized uses can take many forms, such as lower-tier devices accessing spectrum reserved for higher tier devices in a tired spectrum sharing model [3], unauthorized devices accessing licensed spectra using software radios, or various forms of denial of service attacks. Techniques must be developed to detect such unauthorized accesses and large-scale spectrum monitoring is one effective way to do this.

However, large-scale spectrum monitoring using lab-grade spectrum analyzers is not scalable, given that such devices cost anywhere from several thousands to tens of thousands of US$ depending on the exact capability and require availability of AC power. Several recent papers have proposed to address this scalability issue by deploying low-cost, small form-factor, low-power spectrum sensors in large numbers perhaps using a crowdsourcing paradigm [4, 5, 6].[1] The overall monitoring performance achieved by a large number of such low-cost sensors can exceed that of a handful of lab-grade spectrum analyzers while costing several orders of magnitude less [4]. Due to this reason there is a growing body of literature in studying the performance characteristics of commodity-based inexpensive sensors [8, 9, 6].

Although using inexpensive, commodity-grade sensors in large numbers may provide a very encouraging cost-performance tradeoff, use of a crowdsourcing paradigm brings in certain management problems. Spectrum patrolling must involve signal detection. It is unlikely that all deployed sensors will be used in specific detection tasks [4]. Only a subset will be typically be employed ensuring that the required level of detection performance is achieved. This conserves the backhaul bandwidth and also energy when the sensors are battery operated (e.g., when mobile phones serve as spectrum sensors [8]). In case of multiple sensing needs in the same geographical space (e.g., detecting specific signals in multiple spectrum bands), sensors may need to be configured to engage in one specific task as their processing powers may not be sufficient for multiple concurrent signal detection tasks. The broad goal of this work is to *develop mechanisms to select the right set of sensors that optimizes the performance of detection task for a given cost.* There are two sub-problems that arise: 1) modeling individual sensor performance and cost for given configurations, 2) fusing data from multiple sensors and selecting the optimal subset to maximize detection performance subject to cost limitations (or, minimizing cost subject to a given detection performance). While these problems are not entirely new in a general sense, the specific nature of crowdsourced spectrum patrolling problem makes them challenging.

Challenge 1 – Modeling Individual Sensors: Fundamentally spectrum sensors must perform a signal detection task in form of a binary hypothesis testing (intruding transmitter present/absent). Detection performance is usually characterized by standard metrics like the *probability of detection* (P_D) or *false alarm rate* (P_{FA}). Assigning a specific sensor to a specific sensing task and choosing specific configurations, requires accurate estimation of its P_D and P_{FA} metrics and cost for such configurations. Modeling of the cost depends on the scenario and can include, e.g., energy cost, backhaul data cost or any form incentives to be paid to the owner of the sensor. However, given the heterogeneity and diversity of spectrum sensors in a crowdsensing paradigm estimating such metrics accurately is challenging. Existing literature extensively uses so-called first principles modeling approach that could miss various forms of imperfections (e.g., clock skew, I/Q imbalance, RF front end non-linearity) and noises

[1]There is at least one commercially successful crowdsourced application of spectrum sensing. FlightAware [7] deploys low-cost sensors via crowdsourcing to detect signals from aircrafts flying overhead.

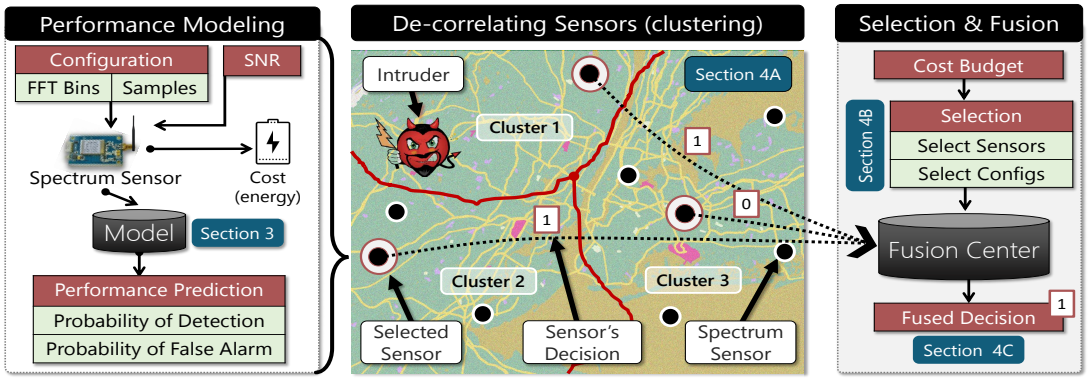


Figure 1: Overview of the proposed technique. Performance models of individual spectrum sensors are created first using a data-driven approach. Then sensor selection and fusion are done after a de-correlation step based on clustering. The figure indicates various sections different steps are described.

common in commodity platforms. Even when they are able to account for those, they require knowledge of internal details of the sensor or separate calibration efforts. These are either not practical or do not scale well. More specifics of these issues will be discussed in Section II.

Instead of relying on first principles models, we use a data-driven (blackbox) approach where models are created based on data from prolonged observation of the sensor. This type of approach is getting traction in other communities such as industrial process control where first-principles approaches are not practical for largely similar reasons (see, e.g., [10]). We abstract out the observable and easily quantifiable parameters of a sensor, its operating environment or runtime configuration. We use machine learning methods that treats the internal sensor hardware information (otherwise inaccessible) as hidden variables. This gives our methodology a direct and practical advantage over involved analytical models. Second, such models get richer with time and can easily accommodate new sensors without the need of explicitly calibrating them, an otherwise impossible task.

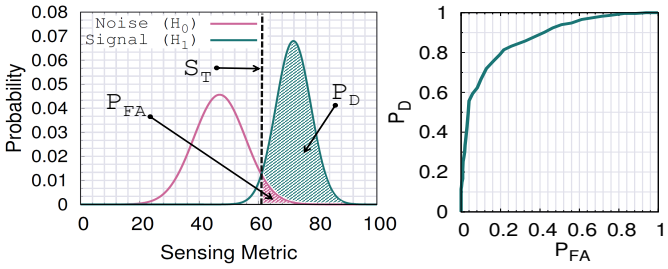
Challenge 2 – Sensor Selection and Fusion: Once individual sensors are modeled, we must select the subset of sensors (and their configurations if they are configurable) to achieve the best cost-performance tradeoff, i.e., the best detection performance for a given total cost (or minimum cost for a given desired performance). Here, the local sensor decisions (target present/absent) are to be combined into a global 'fused' decision. Thus, a fusion rule is needed. While there is a very rich literature on sensor fusion and developing optimal fusion rules much of techniques in literature assume that *sensor decisions are conditionally independent.* This is not true for spectrum sensors, where their decisions could be correlated depending on the sensor locations. The reason is that sensors located in the same neighborhood are likely to face the same fading environment, resulting in correlations in their observations/decisions. The case for correlated observations have been indeed studied (see, e.g., [11, 12, 13]). But these methods are either too complex computationally to implement in practical systems and/or requires prior knowledge of the correlation structure (e.g., in terms of higher-order moments of the sensor observations under each hypothesis [11] or spatial correlation coefficient [14], etc). Also, these techniques do not help addressing the sensor selection problem.

Instead, we propose a method that follows a two step process, 1) first decorrelating the sensors via a clustering technique and 2) then performing a sensor selection using these clusters for guidance. The method is computationally efficient and uses the data-driven approach developed as a part of challenge 1 to model individual sensor's cost vs. performance. Overall, this makes the proposed method perfect fit for crowdsourced spectrum sensing.

Contributions: Figure 1 pictorially describes the overall approach with pointers to various sections of the paper. Overall, we make two sets of contributions. *First,* we develop a systematic approach for data-driven models of spectrum sensors engaged in signal detection (Section III). The model takes the sensor's configuration and SNR as input and estimates detection performance and cost (we use energy to model cost in this work). We precede this modeling approach by highlighting limitations of traditional first-principles based analytical modeling approaches (Section II) and demonstrate improved model performance using the proposed data-driven approach using actual spectrum sensor hardware. *Second,* we develop a technique for the sensor selection and fusion problem taking into account the fact spectrum sensors are not conditionally independent (Section IV). The proposed technique though based on heuristics is suitable for crowdsourcing as it does not require information that is hard to obtain or estimate. We show that the overall detection performance improves significantly relative to baseline techniques.

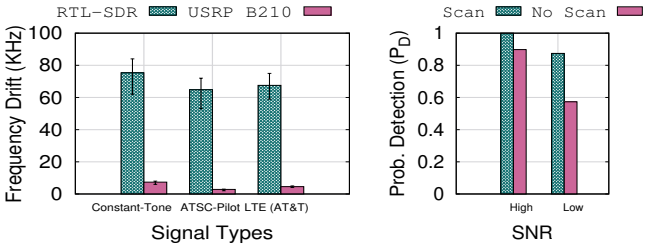
II. Modeling Detection Performance

The spectrum sensor detects the absence or presence of an intruding transmitter's signal. The corresponding hypotheses are denoted as H_0 (absence) and H_1 (presence) respectively. Raw sensed samples from the sensor are fed to the corresponding detection algorithm on board of the sensor that computes a *sensing metric*. The sensing metric is compared against a threshold (S_T) to output a binary decision. This is the local decision of the sensor.



(a) Detector Basics (b) ROC Curve

Figure 2: Working principle of a detector. S_T denotes the *threshold* of the sensing metric. Increasing S_T increases P_D but also increases P_{FA} as per the ROC curve.

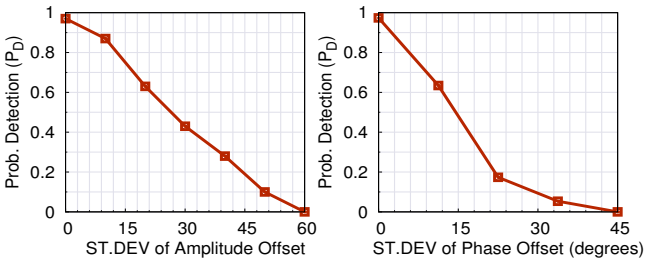


(a) Higher clock-skew in cheaper devices (b) Clock skew deteriorates P_D

Figure 3: Unpredictable clock skew makes frequency offset calculation harder resulting in poorer signal detection performance.

Performance Metrics: Given H_1, the rate at which the sensor detects the transmitter is known as the *probability of detection* (P_D). Second, given H_0, the rate at which the sensor incorrectly flags the presence of a transmitter is known as the *probability of false alarm* (P_{FA}). Figure 2 demonstrates the basic working principle. The sensing metric has two different distributions under hypotheses H_0 and H_1. Under H_0, the distribution reflects noise. P_D and P_{FA} depends on the selection of S_T. Varying S_T varies both P_D and P_{FA} between 0 and 1. This produces the *receiver operating characteristics* (ROC) curve. Specifying P_{FA} (common case) also determines P_D as per the ROC curve. However, the ROC curve itself would look different if the distributions of the sensing metric shown in Figure 2(a) change. This is possible when the signal power from the transmitter changes (due to a different location, e.g.). More on this below.

Challenges: Estimating an optimal value of S_T is straightforward when the distributions of the sensing metric for H_0 or H_1 (Figure 2(a)) are known or can be accurately estimated. Unfortunately, this is not the case in practice. The distributions depend on a variety of factors including the detection algorithm, specifics of the sensor hardware, SNR or SINR at the sensor location, number of sensed samples, FFT resolution and so on. Common detection algorithms are energy-based, waveform or feature-based, autocorrelation or cyclostationary-based. Existing analytical techniques [15, 16, 17] can help model such algorithms to estimate an optimal S_T. However, such models typically result in significant estimation errors [15, 16]. The reasons are as follows. First, many of these



(a) Effect of Amplitude Offset (b) Effect of Phase Offset

Figure 4: We demonstrate the effect of I/Q imbalance in deteriorating the performance of simple waveform based detector algorithm used in detecting an ATSC pilot tone.

models make idealistic assumptions about the distribution of the signal or noise or the noise associated with sensor hardware. For example, [18] shows that the performance of a sensor actually depends on both the signal parameters and the amount of RF front-end non-linearities of the sensors. Second, complex models do exist that take into account such factors [18, 19], but it is seldom possible to parameterize them correctly. This is due to the uncertainty in the hardware itself or inaccessible components that makes reliable measurements impossible. Third, even when such measurements are possible manual calibration of individual sensors does not scale well, especially in the context of crowdsourcing.

We provide two sets of benchmarking experiments to highlight the challenges.

***Clock-skew*:** As an example, we study the clock skew associated with the local oscillator (LO) in the sensor. The frequency set in LO tunes the sensor to the desired frequency. However, the LO-frequency drifts giving rise to clock skew. To understand the nature of such drifts in commodity sensor hardware, we use two different spectrum sensors based on RTL-SDR and USRPB210. These sensors are chosen due to their low-power, small form factor nature [8]. They are both USB-powered and could be driven by an embedded CPU board or even a smartphone. Three test signals are used for detection. The first two are constant frequency tones in the 915 MHz band and the pilot tone of an ATSC signal (DTV band). In both cases we observe a non-trivial frequency drift that varies widely across individual sensor instances. For the third, we use an LTE downlink signal from a real network (AT&T) using these sensors and recorded the frequency correction needed in order to decode the synchronization signals. The results are summarized in Figure 3(a). In most cases RTL-SDR suffers from a appreciable clock skew which is less prevalent in more expensive hardware like USRP. In Figure 3(b) we show the impact of such clock-skew in detecting an ATSC signal. The ATSC signal has a pilot tone located at an offset of 310 KHz that is expected by our waveform based detector algorithm. We create two variations of the algorithm that expects the pilot tone (i) exactly at the 310 KHz offset and (ii) ≈100 KHz surrounding the expected location that it scans. In a low SNR scenario, scanning provides almost a 50% improvement in P_D

compared to the detector that expects the pilot at a fixed offset demonstrating the impact of the clock skew problem.

***I/Q imbalance*:** Apart from clock skew, I/Q imbalance and RF front-end non linearities are other prominent issues. I/Q imbalance is introduced as a result of mismatch between the in-phase (I) and quadrature (Q) signal paths of the RF receive chain. For example, phase difference between the I and Q components is not always exactly 90° which results in an amplitude and phase offset in an I/Q sample. Since we do not have direct control over the radio circuitry we simulate I/Q imbalance by adding amplitude and phase offsets to real I/Q traces obtained for an ATSC signal using a RTL-SDR device. For both cases, we use an offset drawn from a zero-mean Gaussian with a standard deviation as shown in Figure 4. We report the detection rate of the ATSC signal using a waveform based detector that identifies the ATSC pilot signal. As the I/Q imbalance becomes more prominent it becomes impossible to detect the signal. Although I/Q imbalance can be addressed directly in the hardware [19] we expect crowdsourced spectrum sensors may use inexpensive hardware unable to do such corrections.

As mentioned earlier, while such problems can be accounted for by applying models that 'corrects' for such errors, these models are based on the 'first principles' approach. These models can only be applied only after knowing specific sensor-specific parameters (e.g., characteristics of frequency drift, whether the algorithm scans, or nature of I/Q imbalance, etc). This information may not be available in a crowdsourcing scenario given significant possible heterogeneity.

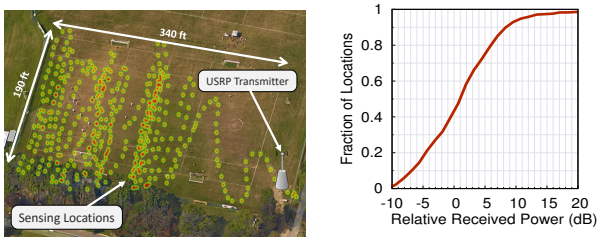
III. Data-Driven Performance Modeling

To address this problem of scalable modeling of heterogeneous sensors, we borrow from the concept of data-driven soft sensors utilized in industrial processes [10, 20]. Industrial processes find it impossible to use first principles models for their physical and chemical processes. These models are often idealized (e.g., assumes steady state behavior) or requires parameters that are hard to obtain. Instead, data-driven soft sensors models are gaining ground that takes an alternative blackbox approach where massive amount of collected data is used to model and predict the industrial process behavior in realistic conditions using statistical or machine learning techniques (see, e.g., [10, 20]).

In the following we present our approach for the data-driven analysis using an example dataset. We first present our dataset, quantify the errors associated with first-principles based analytical models and then present our data-driven performance model of spectrum sensors.

A. Dataset

We collect spectrum sensor measurements in an outdoor setting within the university campus. As shown in figure 5(a), we setup a USRP B210 based transmitter that transmits a constant tone in the 915 MHz band and collect sensing data (I/Q samples) using three RTL-SDR and two USRP B210 devices. We collect 1M samples at every location and our



(a) ≈1000 sensing locations (b) Received power CDF

Figure 5: Spectrum sensor data collection

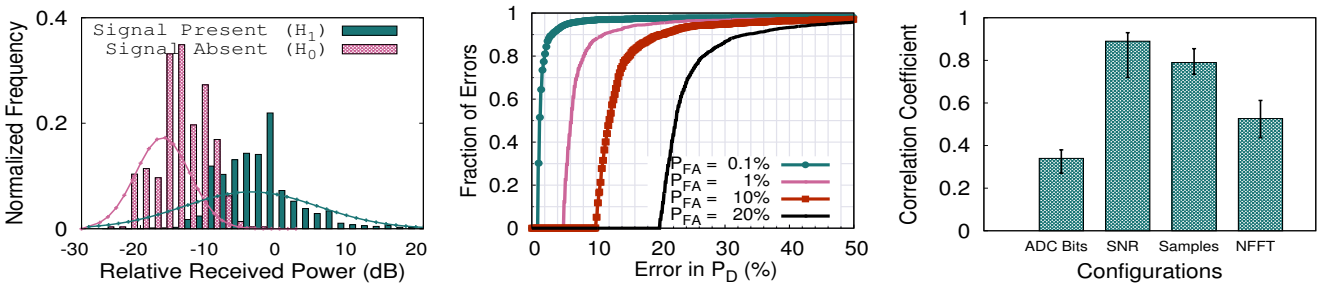
sensing area covers approximately 1000 locations within a 190×340 ft^2 region (Figure 5(a)). The distribution (H_1) of the received power is also shown in Figure 5(b). We bias our data collection towards relatively lower SNR zones so as to have more variations in detection performance. This also presents a more challenging test case – detection is much easier when SNR is high. Using the same set of sensors we also collect a *noise dataset* by turning off the transmitter. This data corresponds to the distribution for H_0. Note that H_0 is agnostic to the sensor's location.

For every location we employ three different detection algorithms (energy, feature and autocorrelation based) [8] both on the signal and the noise dataset. We vary two key parameters of the algorithm that directly influences $P_D - P_{FA}$ as well as energy cost in the sensor [8]: (i) N, number of sensed samples and (ii) $NFFT$, resolution of the FFT. N and $NFFT$ are varied from 32 (2^5) to 4096 (2^{12}) by repeated doubling with the constraint of $N \geq NFFT$ (36 configurations). We introduce heterogeneity in the resolution of sensed samples by changing the number of bits per sample. We produce additional data sets of 14, 12, 10 and 6 bit samples by ignoring least significant bits from the collected 16 bit samples. Note that this depends on the resolution of the ADC in the sensor and heavily influences the dollar cost.

Across all locations, detection algorithms running with different configurations (≈650K in all) we obtain the sensing metrics for H_0 and H_1 respectively. For each location and for every possible configuration at that location, we repeat the detection experiment 1000 times by selecting a contiguous chunk of N samples from the respective 1 M samples starting at a random offset. This gives us 1000 instances of the sensing metrics under the same configuration and we compute P_D and P_{FA} for a given value of the sensing threshold, S_T. By varying S_T, we obtain the *ground truth* ROC curves for all such configurations across all locations.

B. Limitations of Analytical Models

Before directly delving into the internals of the data driven model, we first demonstrate the limitations of first-principles based analytical models using our dataset. Due to space restriction we are not able to explain individual variations of analytical models we use but will explain the general conclusions and trends. Figure 6(a) shows two histograms of the sensing metric corresponding to H_0 and H_1 obtained by using the energy-based detector algorithm (N = 2048, NFFT = 1024). We use the analytical model for energy-based detector



(a) Empirical vs. Analytical Model for a fixed configuration

(b) Error in estimating P_D

(c) Factor analysis for the estimation error

Figure 6: Analysis of estimation errors associated with analytical models and its dependency on the sensor's operating environment or configurations. Median estimation error in P_D can be as high as 25%. Higher errors are highly associated to low SNR operating environments.

to estimate the distributions for H_0 and H_1 for the same location. Figure 6(a) visually shows the difference between *ground truth* and *estimated* distributions. In Figure 6(b) we present the estimation errors for different values of P_{FA}. Note that the median error can be as high as 25% that in many cases. We observe that the errors are particularly higher in low SNR scenarios. We also show (Figure 6(c)) the correlation of such errors to the sensing configurations. Unlike other factors, the number of ADC-bits does not show a very high degree of correlation. This may be because we attempt to detect a simple tone at a constant power in this study.

C. Data-Driven Performance Model

Given the relatively poor performance of parametric models, we make use of 'training data' collected from spectrum sensors to take a non-parametric data-driven approach. Essentially, the task of the model is to determine an optimal sensing threshold, S_T^{opt} that maximizes P_D for a given P_{FA} . For training the model we use feature vectors of the form `V`: <`Algorithm`, `N`, `NFFT`, `B`, `SNR`, $\texttt{P}_{FA}^{target}$>. P_{FA}^{target} is the allowable false alarm rate. `Algorithm` refers to the signal detection algorithm the sensor runs that uses `N`, `B`-bit samples and involves an `NFFT`-bin FFT. We use energy, waveform and autocorrelation based detection algorithms. `SNR` refers to the signal-to-noise ratio of the intended signal at the sensor's location. Every $\texttt{V}_i$ is mapped to a corresponding $S_T^{opt_i}$ in the training examples. Note that we do not explicitly take into account internal hardware details unlike the involved analytical models [21, 22]. We explore off-the-shelf machine learning techniques to learn the estimator for S_T^{opt}. Out of several popular techniques we tried out, the Support Vector Regressors (SVR) works best in our case. We have also explored deep-learning methodologies [10] using convolutional neural networks (CNN), however the amount of training data required to get reasonable estimation performance is significant. This makes CNN impractical in our case and we adopt SVR for creating the performance model.

Evaluation: We demonstrate the performance of our data-driven model in Figure 7. Given configuration of the sensor and the SNR it operates in, our model predicts the optimal threshold S_T^{opt} that maximizes P_D for a fixed P_{FA}. We use

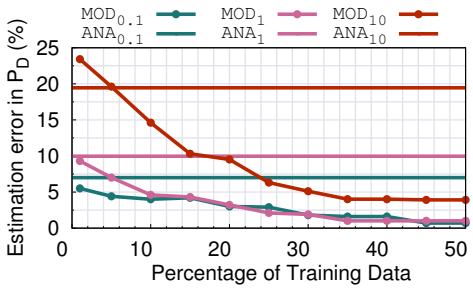


Figure 7: Performance of the data-driven models $MOD_{P_{FA}}$ compared to the analytical models $ANA_{P_{FA}}$ for different values of P_{FA} .

the sensor traces and the model predicted $\widehat{S_T^{opt}}$ to compute $\widehat{P_D}$ for a given P_{FA}. The relative error of $\widehat{P_D}$ with respect to P_D is reported. We restrict our evaluation to sensor traces that has moderate to low SNR values as under such scenarios the models are error prone. We show estimation error in P_D for P_{FA} equal to 0.1%, 1% and 10% respectively. The data-driven models are indicated by $MOD_{P_{FA}}$ in Figure 7. We also present the estimation errors of the analytical models ($ANA_{P_{FA}}$) for the same set of data points (low/moderate SNRs). In all cases after our model is moderately trained we reduce our estimation error by a significant margin with respect to the analytical models. For instance, MOD_{10} outperforms ANA_{10} by $\approx$12% for a training set of size 20%. With more training samples the estimation error of our model becomes negligible and we see a clear improvement over analytical performance models.

IV. Sensor Selection and Fusion

The approach described in the previous section gives us the power to estimate the detection performance of an individual sensor deployed in the wild without explicitly calibrating it. In this section we use such models to optimize the (network-wide or global) detection rate. This is done by selecting an optimal set of sensors (and their configurations such as number of ADC bits, number of samples or FFT bins etc.) and fusing their local decisions into a *network-wide (global)* decision. This needs a simultaneous solution of sensor selection and sensor fusion problems. As discussed in Section I, a wide body of literature exists that propose mathematical techniques to fuse sensor

decisions to optimize certain detection performance metrics (typically Bayes risk). In a widely used method proposed by Chair and Varshney [23] that we will also use, an optimal fusion rule is developed to minimize the sum of false alarm and missed detection rates, but specifically for case when the sensors are conditionally independent.

As explained in Section I, the conditional independence assumption does not hold for spectrum sensors and existing techniques to account for correlated sensor observations are hard to apply for case of crowdsourced spectrum sensors either due to complexity or unavailable parameters. We develop an alternative heuristics-based approach below that we will demonstrate to perform well in practice.

First as a de-correlation step, we partition the set of sensors into spatial clusters (Section IV-A). As a result of the clustering we can assume that the sensors belonging to two different clusters are independent. This allows us to fuse decisions from sensors belonging to different clusters using the Chair-Varshney fusion rule [23]. Second, we develop algorithms that select sensors from each cluster to maximize network-wide probability of detection subject to a given cost budget (Section IV-B). Several variations of the algorithm are proposed: i) homogeneous sensors, ii) heterogeneous sensors, where a 'better' sensor incurs a higher cost, iii) heterogeneous sensors where sensor configurations can be chosen and we select sensors along with their respective configurations. The sensor's energy usage is used as proxy for cost, though our work can be easily adapted for other cost models.

A. De-correlated Sensors

Assume a spatial distribution of sensors in the region of interest. Sensors closer to each other have a higher likelihood to face similar fading environment and record similar observations. This is the basis of our clustering scheme. We use both spatial proximity of the sensors as well as similarity of the RSS values in computing the *distance metric* in between two sensors. The spatial proximity alone is not sufficient as similar distances in between sensors may not result in similar differences in the RSS values. This is due to the non-uniform nature of propagation losses of a wireless signal due to location specific shadowing effects.

We use the *k-means* clustering algorithm to partition the set of sensors into k 'de-correlated' clusters. The distance metric D_{ij} (Mahalonobis distance) between sensors S_i and S_j is computed as,

$$D_{ij} = \sqrt{w_1\, D^2_{Euclidean} + w_2\, D^2_{RSS}} \tag{1}$$

$D_{Euclidean}$ is the Euclidean distance between the sensors S_i and S_j. D_{RSS} is the absolute difference of their respective RSS values, $|RSS_i - RSS_j|$. The weights w_1 and w_2 are equal to the inverse-variance of the Euclidean distance among the sensors and RSS values across the sensors, respectively. We choose the value of k using the *Bayesian Information Criterion* (BIC) method as proposed by Banfield and Raftery [24] for model based clustering techniques.

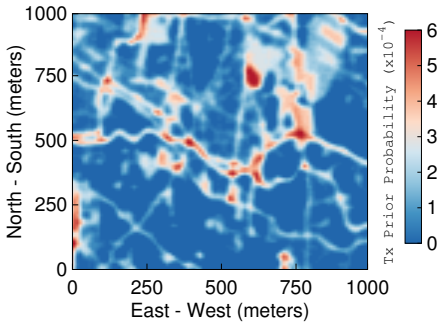


Figure 8: Prior probabilities of the transmitter across a 1000 m×1000 m grid.

B. Sensor Selection

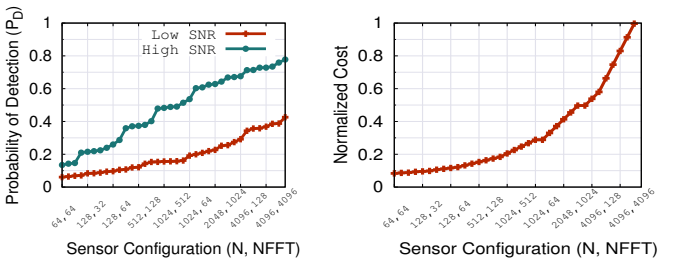
Transmitter Prior Model: Consider a geographical region consisting of K discrete locations $1, \ldots, K$. Let L be the discrete random variable that represents such locations, $L \in \{1, \ldots, K\}$. We assume the existence of a probability mass function (pmf) π_l representing the *prior probabilities* (also called *prior map*) that the intruder or transmitter in question is at location l. Note that to each location l, we associate a coordinate (x_l, y_l) in the physical 2-d plane. See Figure 8 for an example that is used later in our simulations.

Sensor Cost Model: As mentioned before, we use energy as a proxy for sensor cost. This is reasonable as we anticipate that in a crowdsourced scenarios many sensors could be battery driven and could be a part of a mobile device [8, 25]. We adopt some of the energy benchmark results presented in [8]. The energy benchmarks are available for different software configurations (samples, FFT resolution) of signal detection algorithms running on a Raspberry-Pi device interfaced with RTL-SDR. We create a cost model for a sensor using such measurements. We then normalize the cost values to the range (0,1] as shown in Figure 9.

To optimize performance under the constraint of a cost budget, we need to select a set of sensors S that collectively offers the best network-wide detection performance. Let $P_D(S)$ denote the probability that the set of sensors S detects an intruder. We denote the selection of a sensor by setting the decision variable $z_i = 1$, otherwise $z_i = 0$. Let C_i denote the cost of utilizing sensor i. Our objective is to maximize the probability of detection while keeping the cost within a fixed budget B:

$$\text{Maximize } P_D(S) \text{ subject to: } \sum_{i=1}^{N} z_i C_i \leq B.$$

Sensor Ranking: To solve the above optimization problem, we want to rank the sensors based on their contribution to $P_D(S)$. Such ranking depends on the SNR at the sensor's location and hence the location of the intruder itself. Intruder locations are obtained by sampling the prior probabilities. Every location of the intruder is associated with a spatial distribution of received signal strength (RSS) over the sensor locations. This is obtained using a log-normal propagation model. Next, we de-correlate the sensors to generate clusters of sensors as discussed in the previous section.



(a) Configuration vs. P_D for P_{FA} = 0.1%

(b) Configuration vs. Normalized Cost

Figure 9: (a) P_D for different configurations of the sensor under low and high SNR. (b) Sensor cost model. N is number of samples, NFFT is no. of FFT bins.

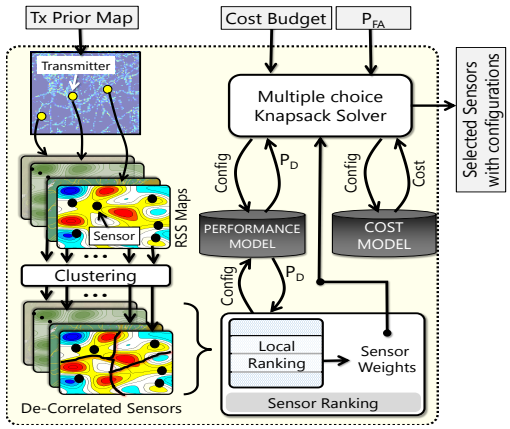


Figure 10: Schematic of the sensor selection technique

The sensor clusters are ordered based on the Euclidean distance of the cluster centroid from the sampled intruder location. We rank the sensors by iterating over every cluster in order and selecting the sensor with the highest P_D value. Once a sensor is chosen from every cluster we repeat the process and choose sensors with the second highest P_D and so on.

Let the vector $R^i_{local} =< w^i_1, w^i_2, \cdots w^i_N >$ denote the local ranking of the N sensors present for the i^{th} possible location of the intruder. In R^i_{local}, w^i_j denotes the local rank of sensor S_j. The weight of sensor S_j is computed as:

$$W_j = \sum_{i=1}^{K} \frac{\pi_i}{w^i_j}.$$

The sensors are ranked in the decreasing order of their weights.

Sensor selection schemes: For each sensor S_i, we now have a fixed weight W_i and a particular cost C_i. This is an instance of a *0-1 knapsack problem*, which is in general NP-hard. Figure 10 shows a schematic diagram of our process of calculating weights and costs for use by our knapsack selection. We first look at solving it in the simple case where sensors are homogeneous in terms of configurations, and followed by heterogeneous configurations.

Homogeneous Sensors (HOMS) We assume all sensors are identical and have the same configuration. Hence their costs are equal and we assume unit cost for every sensor, i.e., C_i = 1. In this case we always need to choose the sensors with the highest weights (W_i). This can be simply achieved by selecting sensors in decreasing order of their W_i's, until their sum exhausts the cost budget.

Heterogeneous Sensors (HETS): In this case the sensors have heterogeneous configurations that are preconfigured for every sensor and cannot be changed. Accordingly, the sensor's cost C_i is a function of its configuration as demonstrated in Figure 9(b). Depending on the sensor's configuration, C_i can vary anywhere from the minimum cost value to 1. Thus, we select sensors based on their weights, while also ensuring that selecting them does not incur too much cost. This problem can be solved approximately by keeping track of the cost for each sensor added to the existing set of selected sensors using dynamic programming (Algorithm 1). This algorithm solves the problem in $O(N^3/\theta)$ time, where θ is a parameter representing the tradeoff between optimality and time complexity. Here, increasing the value of θ reduces time complexity while increasing the level of approximation and vice-versa.

Reconfigurable Sensors (RES): In this case there are multiple possible configurations, making it difficult to rank the individual contributions of each sensor. We therefore divide the budget B itself into different components B_{loc} for each cluster. We formulate this problem as follows. Let there be M possible configurations for each sensor. Each sensor S_i needs to choose some configuration j from $\{1, \ldots, M\}$. Then each sensor configuration has a weight W_{ij} and a cost $C_{ij}, \forall j \in \{1, \ldots, M\}$. We also augment the decision variable to z_{ij}, where z_{ij} denotes whether sensor S_i uses j^{th} configuration. We rewrite the optimization problem as:

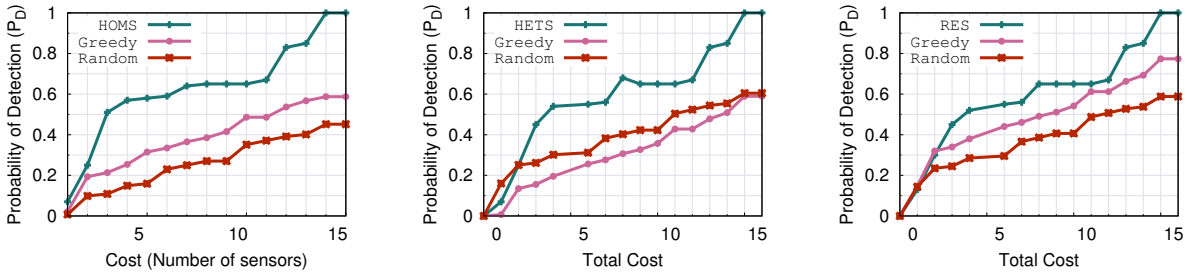
$$\text{Maximize } P_D(S) \text{ subject to:}$$
$$\sum_{j=1}^{M} z_{ij} \leq 1 \text{ and } \sum_{i=1}^{N}\sum_{j=1}^{M} z_{ij}C_{ij} \leq B_{loc}$$

This is an instance of *multiple-choice knapsack problem*. We solve this by adding another dimension to Algorithm 1 and then using a similar dynamic programming approach. Figure 10 shows a schematic of the entire sensor selection process.

C. Sensor Fusion

We now have a selection of sensors and their configurations. We use the Chair-Varshney optimal sensor fusion rule [23] that fuses the local decisions of the individual sensors into a global (fused) decision to minimize the error rate. We apply this fusion rule repeatedly for each possible location of the intruder. Assume that $U_{i,L=j}$ is the local decision (1 or 0) of the sensor S_i, if the intruder signal is detected or not detected (respectively) by this sensor given the intruder is at location j. Using [23], we compute the fused decision $U_{L=j}$ of the sensors given this location of the intruder as:

$$U_{L=j} = \sum_i [U_{i,L=j} \log \frac{P_{Di,L=j}}{P_{FAi}} + (1 - U_{i,L=j}) \log \frac{1 - P_{Di,L=j}}{1 - P_{FAi}}] \quad (2)$$



(a) Homogeneous Sensors (HOMS) (b) Heterogeneous Sensors (HETS) (c) Reconfigurable Sensors (RES)

Figure 11: Comparison of P_D values for the three proposed schemes with *greedy* and *random* baseline heuristics.

Algorithm 1 HETS: Heterogeneous Sensor Selection.

Input: Weight of sensors G, cost of sensors C, cost budget B, degree of approximation (θ)
Output: Optimal selection A

$G_i \leftarrow G_i/\theta,\ \forall i \in N$
$T = \sum_{i=1}^{N} G_i$
$A[0, j] \leftarrow \infty,\ \forall j = 0, \ldots, T$
/* A[i,k] stores lowest cost among first i sensors with at least k value.*/
for all $i = 1, \ldots, N$ **do**
  for all $k = 0, \ldots, T$ **do**
    if $G_i < k$ **then**
      $A[i-1, k] = A[i-1, k - G_i] + C_i$
    else
      $A[i-1, k] = \min(A[i-1, k - G_i] + C_i, A[i-1, k])$
    end if
  end for
end for
for all $k = 1, \ldots T$ **do**
  if $A[N, k] > B$ **then**
    return $A[N, k-1]$ /*Best config with cost at most B */
  end if
end for

The summation above is for all selected sensors. $P_{Di,L=j}$ is the probability of detection of sensor S_i for intruder location j. $U_{L=j} > 0$ indicates presence of the intruder (at location j), otherwise it is considered absent. To estimate the presence of an intruder anywhere, we first compute the values of $U_{L=j}$ for all possible locations j. We conclude that there is an intruder anywhere only if at least one of these $U_{L=j}$'s is positive. Otherwise, we conclude that no intruder is present.

D. Evaluation

We simulate a 1000 m×1000 m grid where we randomly deploy 100 spectrum sensors. The sensors can choose among 36 different configurations. Each configuration corresponds to the tuple (N, NFFT), N being the number of I/Q samples and NFFT, the resolution of the FFT in the sensor's detection algorithm. N, NFFT $\in \{2^5, 2^6, \cdots, 2^{12}\}$ such that N $\geq$ NFFT. For each sensor, we set P_{FA} = 10% (or 0.1) and obtain the P_D from our data-driven performance model (MOD_{10}). The sensors have a cost model as mentioned in Section IV-B. Next, we simulate an intruder in the grid. The intruder is represented by a wireless transmitter with a transmit power of 10 dB. We use the log-normal model to compute RSS at all the sensor locations. We make the intruder's *prior map* realistic to account for different factors such as terrain information or proximity to residential or navigable areas. We create the *prior map* directly from a snapshot of Google map's satellite imagery data. To remove intricate details (e.g., buildings, texture) in the image, we apply Gaussian blur, a well known image filtering technique. Next we resize the image to a dimension of 1000×1000 to emulate our grid. We make the prior probability of the transmitter to be present in a certain cell $< i, j >$ proportional to the pixel intensity at $< i, j >$. Figure 8 shows our prior map. For all simulations we sample the intruder's location 10 K times from the *prior map* that we use to obtain weights for our sensor selection algorithms. Every time the intruder appears the selected sensors attempt to determine its presence with their respective values of P_D. The fused decision is compared to the ground truth and the detection rate for the given instance of selected sensor is computed by simulating the intruder 1000 times.

We compare the performance of our sensor selection algorithms with two baseline algorithms. As baseline, we first run a *random* selection algorithm where we pick the sensors randomly with uniform probability across the region of interest. We also run a *greedy* algorithm where we pick the best sensors without accounting for their correlation. When the sensors are homogeneous, the *greedy* algorithm selects the sensors for which the prior probabilities are the highest. For other cases, the *greedy* algorithm selects sensors in decreasing order of their P_D's.

Observation: Figure 11 shows the P_D obtained by the sensors selected by our algorithms compared baseline heuristics across different cost budgets. For HOMS, we consider the number of sensors as the cost, i.e., C_i = 1. However for HETS and RES, the cost $C_i \in [min_{cost}, 1]$. We note that our algorithms perform significantly well compared to *greedy* and *random* schemes for higher budget levels. For all cases, till a budget of 2, our algorithms perform similar to the *greedy* scheme. This is because both of them select sensors only from the cluster with high prior probability. When we increase the budget above 2, the *greedy* method keeps selecting from the same cluster, since it does not consider the effect of correlation. For instance, at a budget of 15, HONS, HETS and RETS outperform *greedy* scheme by 40%, 35% and 28% respectively and outperforms

the *random* scheme by 50%, 35% and 37%. Our algorithms, because of its system of sensor weights, selects sensors from the different clusters which improves the detection rate much faster after a certain budget level.

V. Related Work

Shared spectrum architectures need to enforce suitable policies to control spectrum access among secondaries [26, 27]. Second, with the advent of cheaper radio hardware the licensed spectrum is prone to unauthorized use [28]. This makes the problem of spectrum patrolling important. [2] introduces the concept of crowdsourced enforcement of spectrum policies.

Performance of low cost spectrum sensors: The authors in [2] assume complete knowledge about the performance of crowdsourced sensors which is not practical. [2] also assumes the sensors to be homogeneous which is generally not true in a crowdsourced environment. Spectrum monitoring using cheap crowdsourced sensors is not new [4, 9, 6] but they do not provide any insights regarding performance or reliability of sensing. We also show that analytical techniques [29] that model the sensor's detection performance are often simplistic and error prone. [19, 21] builds upon the analytical techniques providing corrections for hardware related aspects like I/Q imbalance, RF front-end non-linearities etc. Inspired by [20, 10], we use a data-driven approach to create performance models of heterogeneous spectrum sensors.

Sensor Selection and Fusion: A good amount of literature exists that study the problem of selecting sensors and combining the decisions of multiple sensors. Chair and Varshney [23] provide an optimal sensor fusion rule when the individual sensor outputs are conditionally independent of one another. Different techniques of fusing multiple sensor decisions are presented in [30]. Some studies have also looked at the problem of distributed spectrum monitoring. [3] proposes using collaborative sensing across multiple sensors to better monitor spectrum. Our work builds upon these studies to focus on detecting the presence of spectrum intruder.

VI. Conclusion

In this work we address the problem of spectrum patrolling using crowsourced heterogeneous sensors. To the best of our knowledge this is the first work that models the performance of a spectrum sensor in a data-driven way. Our model provides significant improvement over state-of-the-art 'whitebox' models. Next we address the problem of sensor selection and fusion of heterogeneous sensors deployed over a region of interest to improve intrusion detection performance within a cost budget. We investigate different scenarios of homogeneous, heterogeneous and reconfigurable sensors. Our sensor selection algorithms perform significantly better than reasonable baseline heuristics. We highlight challenges of the patrolling problem in a cost-effective fashion using crowdsourced sensors and develop mechanisms to address them.

Acknowledgement

This work is partially supported by NSF grants AST-1443951, CNS-1642965 and by the MSIT, Korea, under the ICT Consilience Creative Program (IITP-2017-R0346-16-1007). The authors are grateful to the anonymous reviewers for their constructive feedback.

References

[1] S. Huang, X. Liu, and Z. Ding, "Opportunistic spectrum access in cognitive radio networks," in *Proc. IEEE INFOCOM*, 2008.

[2] A. Dutta and M. Chiang, "See Something, Say Something – crowdsourced enforcement of spectrum policies," *IEEE Trans. on Wireless Communications*, vol. 15, no. 1, pp. 67–80, 2016.

[3] A. Ghasemi and E. S. Sousa, "Collaborative spectrum sensing for opportunistic access in fading environments," in *Proc. IEEE DySPAN*, 2005.

[4] A. Chakraborty, M. S. Rahman, H. Gupta, and S. R. Das, "Specsense: Crowdsensing for efficient querying of spectrum occupancy," in *Proc. IEEE INFOCOM*, 2017.

[5] T. Zhang, N. Leng, and S. Banerjee, "A vehicle-based measurement framework for enhancing whitespace spectrum databases," in *Proc. ACM Mobicom*, 2014.

[6] R. Calvo-Palomino, D. Giustiniano, V. Lenders, and A. Fakhreddine, "Crowdsourcing spectrum data decoding," 2017.

[7] "FlightFeeder for Android, FlightAware." http://flightaware.com/adsb/ android/.

[8] A. Chakraborty, U. Gupta, and S. R. Das, "Benchmarking resource usage for spectrum sensing on commodity mobile devices," in *Proc. ACM HotWireless*, 2016.

[9] A. Nika, Z. Li, Y. Zhu, Y. Zhu, B. Y. Zhao, X. Zhou, and H. Zheng, "Empirical validation of commodity spectrum monitoring," in *Proc. ACM SenSys*, 2016.

[10] W. Yan, D. Tang, and Y. Lin, "A data-driven soft sensor modeling method based on deep learning and its application," *IEEE Trans. on Industrial Electronics*, 2017.

[11] M. Kam, Q. Zhu, and W. S. Gray, "Optimal data fusion of correlated local decisions in multiple sensor detection systems," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 28, no. 3, pp. 916–920, 1992.

[12] E. Drakopoulos and C.-C. Lee, "Optimum multisensor fusion of correlated local decisions," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 27, no. 4, pp. 593–606, 1991.

[13] J. Unnikrishnan and V. V. Veeravalli, "Cooperative sensing for primary detection in cognitive radio," *IEEE Journal of selected topics in signal processing*, vol. 2, no. 1, pp. 18–27, 2008.

[14] A. S. Cacciapuoti, I. F. Akyildiz, and L. Paura, "Correlation-aware user selection for cooperative spectrum sensing in cognitive radio ad hoc networks," *IEEE JSAC*, vol. 30, no. 2, pp. 297–306, 2012.

[15] H. Urkowitz, "Energy detection of unknown deterministic signals," *Proc. IEEE*, pp. 523–531, 1967.

[16] F. F. Digham, M.-S. Alouini, and M. K. Simon, "On the energy detection of unknown signals over fading channels," *IEEE Trans. on Communications*, pp. 21–24, 2007.

[17] J. Chen, A. Gibson, and J. Zafar, "Cyclostationary spectrum detection in cognitive radios," in *Cognitive Radio and Software Defined Radios: Technologies and Techniques, 2008 IET Seminar on*. IET, 2008.

[18] A.-A. A. Boulogeorgos, H. A. B. Salameh, and G. K. Karagiannidis, "Spectrum sensing in full–duplex cognitive radio networks under hardware imperfections," *IEEE Trans. on Vehicular Technology*, vol. 66, no. 3, pp. 2072–2084, 2017.

[19] A. Gokceoglu, S. Dikmese, M. Valkama, and M. Renfors, "Enhanced energy detection for multi-band spectrum sensing under rf imperfections," in *Proc. IEEE CROWNCOM*, 2013.

[20] C. Shang, F. Yang, D. Huang, and W. Lyu, "Data-driven soft sensor development based on deep learning technique," *Journal of Process Control*, vol. 24, no. 3, 2014.

[21] A.-A. Boulogeorgos, H. B. Salameh, and G. Karagiannidis, "Spectrum sensing in full–duplex cognitive radio networks under hardware imperfections," *IEEE Trans. on Vehicular Technology*, 2016.

[22] A. Gokceoglu, S. Dikmese, M. Valkama, and M. Renfors, "Energy detection under IQ imbalance with single-and multi-channel direct-conversion receiver: Analysis and mitigation," *IEEE JSAC*, 2014.

[23] Z. Chair and P. Varshney, "Optimal data fusion in multiple sensor detection systems," *IEEE Trans. on Aerospace and Electronic Systems*, 1986.

[24] J. D. Banfield and A. E. Raftery, "Model-based gaussian and non-gaussian clustering," *Biometrics*, pp. 803–821, 1993.

[25] A. Chakraborty, S. R. Das, and M. Buddhikot, "Radio environment mapping with mobile devices in the tv white space," in *Proc. ACM Mobicom'13*.

[26] J.-M. Park, J. H. Reed, A. Beex, T. C. Clancy, V. Kumar, and B. Bahrak, "Security and enforcement in spectrum sharing," *Proc. IEEE*, vol. 102, no. 3, pp. 270–281, 2014.

[27] X. Jin, J. Sun, R. Zhang, Y. Zhang, and C. Zhang, "Specguard: Spectrum misuse detection in dynamic spectrum access systems," in *Proc. IEEE INFOCOM*, 2015.

[28] "FCC Proposes Levying Huge Fine on New York Police Radio Jammer, ARRL news," http://bit.ly/2uPsB1P.

[29] R. Tandra and A. Sahai, "SNR walls for signal detection," *IEEE Journal of selected topics in Signal Processing*, pp. 4–17, 2008.

[30] B. Ao, Y. Wang, L. Yu, R. R. Brooks, and S. Iyengar, "On precision bound of distributed fault-tolerant sensor fusion algorithms," *ACM Computing Surveys (CSUR)*, p. 5, 2016.