VIA: Establishing the link between spectrum sensor capabilities and data analytics performance

Karyn Doke, Blessing Okoro, Amin Zare†, Mariya Zheleva
Department of Computer Science, University at Albany, SUNY, {kdoke, aaokoro, mzheleva}@albany.edu, †Unaffiliated

IEEE INFOCOM 2024 - IEEE Conference on Computer Communications | 979-8-3503-8350-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/INFOCOM52122.2024.10621266

***Abstract*—Automated spectrum analytics inform critical decisions in dynamic spectrum access networks such as (i) how to allocate network resources to clients, (ii) when to enforce penalties due to malicious or disruptive activity, and (iii) how to chart policies for future regulations. The insights gleaned from a spectrum trace, however, are as objective as the trace itself, and artifacts introduced by sensor imperfections or improper configuration will inevitably affect analysis outcomes. Yet, spectrum analytics have been largely developed in isolation from the underlying data collection and are oblivious to sensor-induced artifacts.**

To address this challenge, we develop `VIA`, a framework that attributes sensor properties and configuration to spectrum data fidelity, and models the relationship between spectrum analytics performance and data quality. `VIA` does not require expert input or intervention and can be used to profile the fidelity of unknown sensors. `VIA` takes as an input a spectrum trace and the sensor configuration, and benchmarks data quality along three dimensions: (i) `Veracity`, or how truthfully a scan captures spectrum activity, (ii) `Intermittency`, characterizing the temporal persistence of spectrum scans and (iii) `Ambiguity` quantifying the likelihood of false detection. We employ `VIA` to measure the data fidelity of five common sensor platforms. We then predict the outcome of several spectrum analysis tasks including occupancy and transmitter detection, and modulation recognition using both controlled and real-world measurements. We demonstrate high prediction performance with an average mean squared error of 0.0013 across all tasks using both regression and neural network models.

I. Introduction

Spectrum analytics is a cornerstone for future wireless networking [1; 2]. The true benefit of data, however, is in the information that can be extracted from it, and its corresponding support of disparate spectrum applications. Emerging applications include spectrum enforcement, improved policy and the design of third-party measurement infrastructures. An enforcer might ask which among multiple transmitters is rogue, whether it is mobile or stationary and what is its location; a policy-maker might be interested in coexistence analysis in a certain band and identifying patterns of utilization; and a crowd-sourced sensing platform might be interested in the maximum attainable accuracy for a new sensor platform. Since these application pose different analytics questions, data quality from the spectrum measurement pipeline will affect them differently. Hence the importance of modeling the effects of spectrum data fidelity on the quality of downstream analytics.

Spectrum analytic algorithms have been traditionally developed in isolation from the underlying data collection principles. This poses a two-fold challenge. First, we lack methodologies to link sensor properties and configuration with the quality of collected spectrum data. Second, we lack understanding of the effects that data quality has on the accuracy of spectrum analysis. The importance of coupling spectrum data with scan metadata is gaining traction, resulting in several metadata standards: VITA49 [3], CHDR [4] and, most recently, SigMF [5], which underpins the IEEE 802.22.3–SCOS [6]. While important, these standards focus on the radio front-end's specification and not on the sensors' computational resources, environment, connectivity and mobility. However, the latter aspects, equally impact spectrum data quality. The relationship between sensor quality and analysis accuracy has been studied for specific sensor platforms [7; 8; 9; 10; 11]. However, there does not exist a principled approach that quantifies this relationship in a platform-agnostic manner. Furthermore, prior work focuses on a subset of sensor configurations, such as sensitivity and frequency resolution, however, additional properties need to be considered too.

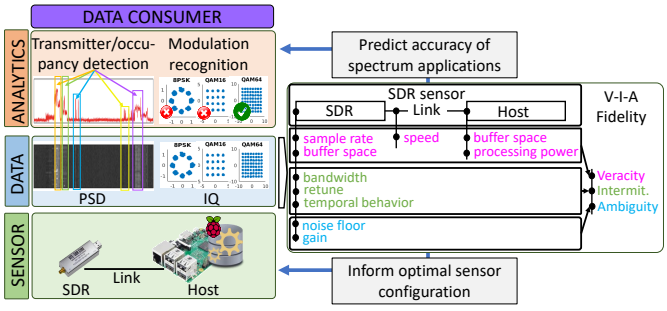

Fig. 1: **`VIA`** overview. A sensor's capabilities and configuration affect the quality of collected data. **`VIA`** profiles data quality across three dimensions: veracity, ambiguity and intermittency. The right pane of this figure color-codes low-level sensor properties that affect veracity, ambiguity and intermittency.

To address these challenges, we develop **`VIA`**, a framework that can automatically profile the fidelity of spectrum traces, and predict the accuracy of spectrum analytic tasks or inform optimal sensor configuration. Fig. 1 conceptualizes **`VIA`**. Considering an end-to-end data pipeline (left pane) comprised of spectrum sensors, data storage, analytics engine, and data consumers, **`VIA`** fits as a middle layer between the sensors and the data storage (right pane). **`VIA`** draws on the collected data (power spectrum density or `IQ` samples) and metadata about the sensor's configuration and environment, and adds three more metrics to the metadata: veracity, intermittency and ambiguity. `Veracity` is a measure of the truthfulness with which a sensor captures spectrum activity.

`Intermittency` measures the temporal persistence of a spectrum trace. `Ambiguity` captures the likelihood that transmitter values might be confused with noise and vice versa. Highlighted accordingly are low-level sensor capabilities that affect the **VIA** metrics. These three statistics can be used as predictors for the performance of downstream spectrum analytics applications, or to inform sensor configurations to maximize data fidelity or achieve a certain performance benchmark. We design **VIA** to lower the barrier for expert knowledge in sensor configuration and spectrum data quality, and to support a variety of use cases. For example, a policy-maker interested in the aggregate interference from a new ruling might query a set of historic data from relevant other bands whose data fidelity is above a certain **VIA** threshold. She can then use the data along with propagation models to inform new policies through quality data-driven analysis.

Using real-world traces across all configurations of five sensor platforms, we demonstrate that **VIA** quantifies data fidelity and is a strong predictor for analytics performance. We collect 1700 traces from a controlled transmitter, and 100 traces of real-world FM radio and Television White Space (TVWS) networks. We consider three spectrum analytic tasks: (i) occupancy and (ii) transmitter detection, and (iii) modulation recognition. To showcase **VIA**, we use two predictive models: (1) a simple regression and (2) a neural network, to predict the performance of the above three tasks. For both models, our input is comprised of the **VIA** vectors, whereas the output is the accuracy of the corresponding task. We demonstrate high prediction accuracy with an average mean squared error of 0.0013 even with a simple regression model when training is performed on all five platforms. This accuracy is retained across all tasks as we reduce the training pool from five to two platforms, while testing on the remaining unknown platforms. This demonstrates that **VIA** can predict the performance of unseen sensor platforms, making it an ideal tool for black-box characterization of the data fidelity of any arbitrary sensor.

This paper makes the following contributions.

- We conceptualize **VIA**, the first middle layer for spectrum measurements that can profile the data fidelity of unknown sensors while accounting for sensor properties, configuration and the radio environment.
- We evaluate **VIA** in a comprehensive experimental campaign that included five SDR-based sensors across all their possible configurations in indoor and outdoor settings, and with controlled and real-world transmissions.
- Using three spectrum analytic tasks, we demonstrate that **VIA** is platform-agnostic and can support data fidelity benchmarks for unknown sensors.
- **VIA** is extensible to various tasks (i.e. anomaly detection, localization, spectrum usage statistics) and across arbitrary unknown platforms. New data fidelity vectors can also be incorporated into **VIA**.
- **VIA** has the potential to lower the entry barrier into spectrum research by allowing non-experts to use it as a black-box tool-set to benchmark sensor capabilities.

II. Related Work

Platforms for spectrum measurement. There exist several spectrum measurement systems including the Chicago Spectrum Observatory [12], Microsoft's Spectrum Observatory [13], NTIA's repository [14], CityScape [15] and Electrosense [16]. These systems employ low-/mid-cost spectrum sensors such as the RTL-SDR [16; 17] or USRP [13], and perform sweep-based monitoring of the sub-6 GHz bands. Some of these platforms provide basic spectrum occupancy analytics, such as spectrograms or idle/occupied fraction of time. Their primary focus is on sensor integration, data storage and presentation, however, they do not facilitate deep spectrum analytics or application-driven measurement. Data is collected in the form of `IQ` samples or power spectral density (PSD) and is usually not coupled with meta-data about the underlying sensor properties or configuration. NTIA's repository [14] is the only one that attempts to collect meta-data with spectrum traces using the SigMF [18] data schema. However, this metadata is not being used to aid spectrum analytics. **VIA** is orthogonal to these systems as it seeks to establish a middle layer that ingests side information about the sensor configuration and the radio environment to quantify the fidelity of spectrum data. **VIA** also seeks to further inform spectrum meta-data formats that will help attribute data fidelity to the capabilities and configuration of the underlying hardware.

Spectrum data analytics. There exist a plethora of applications that ingest spectrum data to glean usage insights. These insights can then feed into resource allocation technologies, spectrum enforcement engines or policy-making decisions. Spectrum data analytics tasks include general spectrum awareness [19; 20; 21; 22; 23; 24; 25; 26; 27; 13; 28; 29], transmitter localization [8; 9; 30] and modulation recognition [31; 32; 33; 34; 35; 36; 37; 38; 39; 40; 41; 42; 43; 44]. All of these efforts assume that data is readily available and focus on the design of (often machine learning inspired) spectrum analysis methodologies. However, analysis efforts are usually oblivious to the underlying sensor platforms and any data quality issues they might have created. **VIA** is orthogonal to these works as it seeks to establish the relationship between sensor capabilities and the performance of these analysis algorithms.

Bridging sensor capabilities with spectrum analytics. Most closely related to our work is research on sensor benchmarks that establishes direct relationships between sensors' configuration and the performance of an analysis algorithm [7; 8; 9; 10; 11]. [8] focuses on crowd-sourced RTL-SDR measurements for transmitter identification and localization. [7] studies noise level, sensitivity and energy consumption of three realizations of an RTL-SDR sensor. [9] develops a framework that attributes RTL-based sensor properties (gain and FFT size) to the detection of idle/occupied bands, driving sensor selection and configuration. [10] informs spectrum sensing for narrow-band fleeting signals, while [11] creates benchmarks for sensor selection in transmitter localization. While these works break important grounds in attributing sensor properties to spectrum insights and sensor selection, they require full

supervision. Each target platform has to be exercised across all its configurations while collecting controlled spectrum traces. These traces have to be characterized before one can model the relationship between algorithm performance and sensor properties. This quickly becomes prohibitive for new platforms as each un-profiled sensor has to undergo a rigorous benchmark with experts in the loop. In addition, existing approaches that focus on one sensor type will face practical challenges in systems with heterogeneous spectrum sensors. Our work establishes an automated framework with limited supervision and no expert input that can automatically benchmark the fidelity of spectrum data for an arbitrary unknown sensor.

III. Sensors and data quality

Software defined radio (SDR) spectrum measurement platforms are increasingly considered for data collection [13; 16]. A SDR sensor is comprised of (i) an SDR-based radio frontend, (ii) a host and (iii) the link between the two. The SDR collects discrete samples from a pre-configured radio frequency and with a given sampling rate and feeds those samples through the link to the host for further processing. Depending on the task, the resulting data can be in the form of complex `IQ` samples [42], PSD [13] or otherwise compressed/pre-processed versions of those data types [45].

1) **The host** can be either a general purpose computer (PC/laptop) or an embedded computer (Raspberry Pi/mobile). The data processing speed, the probability for dropped samples, and the timeliness of SDR control are all issues introduced by the host capabilities. These may affect the rate at which a frequency band is revisited by a spectrum sweeping sensor, and the likelihood of that sensor to miss target activity.
2) **The link** connecting the SDR with the host can be realized via different technologies such as USB 2.0/3.0 or Ethernet. This determines the maximum transfer speed of samples between the SDR and the host. Samples might be dropped if the sampling rate of the SDR is higher than the transmission rate supported by the link. The volume of dropped samples, in turn, affects a sensor's ability to truthfully capture the spectrum activity, even when a frequency band is observed continuously.
3) **The SDR frontend** effects on data quality stem from the SDR's instantaneous bandwidth (which is determined by its supported sampling rates), tuning range, analog-to-digital resolution and sensitivity. First, as a target observable range might be orders of magnitude larger than the instantaneous bandwidth of a sensor, measurement infrastructures [16] utilize sequential sweeping of consecutive bands. As a result, any given band is scanned intermittently, which may lead to omission of spectrum activity. To increase the instantaneous bandwidth and reduce scan intermittency, sweeping sensors are often configured to scan with high sampling rates. This, however, may lead to buffer overflows, especially in sensors where lower-rate host-SDR links present a bottleneck [17]. Dropped samples caused by buffer overflows affect a sensor's capability to accurately capture spectrum activity, especially with short-lived or narrow-band transmissions. The analog-to-digital converter resolution and the sensitivity of the SDR affect the degree to which a sensor can pick out faint signals.

The combination of effects introduced by the SDR, the host and the link affect the quality of data collected by a sensor, the truthfulness with which this data represents spectrum activity and the accuracy with which spectrum insights can be drawn. Thus, understanding and quantifying the effects of sensor properties on data quality is essential for trustworthy spectrum measurement. To this end, we design **VIA**, detailed next.

IV. Methodology

We now present **VIA**, which quantifies the fidelity of `IQ` and PSD data collected by a spectrum sensor across three dimensions: `Veracity`, `Intermittency` and `Ambiguity`. `Veracity`, is a measure of how truthfully the data represents spectrum activity. In terms of lower-level sensor configuration, veracity is expressed as the amount of spectrum samples a sensor is actually able to retain using a particular configuration. Intuitively a large volume of dropped samples leads to lower accuracy in spectrum characterization. Sensor properties that affect veracity (Fig. 1 right pane) include the set sampling rate, buffer spaces and the speed of sample transmission from the sensor's front-end to the host. `Intermittency` captures the non-contiguous nature of sweep-based spectrum traces. Intuitively, if a sensor is set on a particular frequency at a given location for continuous amount of time, this sensor will capture all occurring spectrum activity. However, emerging systems [12; 13; 14; 16; 15] target wide bands and perform sweep-based sensing with stationary or mobile sensors. Thus, spectrum activity in a given band may only be captured intermittently. Underlying sensor properties that affect intermittency include the set instantaneous bandwidth, dwell time, target frequency range and the amount of time it takes for a sensor to tune to a particular frequency. Finally, the `Ambiguity`, captures the likelihood that data samples arising from transmitters are confused with noise and vice versa. Ambiguity is most affected by the sensor's noise floor and the set receiver gain. **VIA** can be used as a black box toolset to benchmark sensor capabilities by non-experts. Next, we describe our testbed and data, and detail the **VIA** framework.

A. Testbed and Data Collection

Our testbed consists of a transmitter and five sensors. The transmitter is set with a USRP B210 [46] SDR and a PC (Intel i74770 CPU and 16GB RAM). For the sensors, we use two SDRs–an RTL-SDR [47] and a USRP B210, and three host platforms, a PC, a Laptop, and a Raspberry Pi (Table I). All hosts run Ubuntu 16 with GNURadio3.7 [48]. All sensors are equipped with a wide-band multi-polarized antenna [49]. Our testbed stores data as `IQ` samples, which are then converted to PSD for some analytics tasks (§VI). We make a best effort to control for ambient interference by selecting unoccupied spectrum for our controlled transmissions. We note that controlled measurements are only necessary for our experimentation and will not be required for using **VIA** in the wild.

TABLE I: Evaluated sensor platforms an dtheir corresponding configurations.

Platform	SDR	Host	SDR-HOST Link	Sample Rates (Msps)	RX Gain (dB)
RTL-Pi	RTL2832U	Raspberry Pi (quad-core, RAM 1GB)	USB2.0	1,2,3	24,30,40,50,56
RTL-Laptop	RTL2832U	Lenovo X201 Laptop (i7-5600, RAM 8GB)	USB3.0	1,2,3	24,30,40,50,56
RTL-PC	RTL2832U	Dell Desktop (i7-4770, RAM 16GB)	USB3.0	1,2,3	24,30,40,50,56
USRP-Laptop	USRP-B210	Lenovo X201 Laptop (i7-5600, RAM 8GB)	USB3.0	1,2,4,8,12,16,20,24,28,32	38,40,50,60,70,76
USRP-PC	USRP-B210	Dell Desktop (i7-4770, RAM 16GB)	USB3.0	1,2,4,8,12,16,20,24,28,32	38,40,50,60,70,76

We set up two controlled transmitter configurations for data collection. First, we collect samples to demonstrate **VIA** with occupancy and transmitter detection. Thus, we setup a controlled transmitter within an unoccupied TV channel at 572-578MHz [50], emitting periodic and broadcast patterns with a signal modulated with BPSK. For broadcast, we transmit 1200 bytes at a bandwidth of 1MHz and vary the gain from 30-80dB in increments of 10dB. For periodic patterns, we transmit 1200 bytes every second at a 1MHz bandwidth and 60dB gain. Both the transmitter and sensors are in line of sight at a distance varied between 5 and 25 ft. For the second setup, we configure our testbed to collect IQ samples across four modulations to demonstrate **VIA** with modulation recognition. We set the transmitter gain at 60dB to generate signals modulated with BPSK, QPSK, 8PSK, and QAM16. The transmitter emits a broadcast pattern by continuously transmitting 1200 bytes. The instantaneous bandwidth of the transmitter is set to 1-3MHz and the center frequency of the transmitter is set to 1.2GHz.

Data Collection. We collect 1700 traces with a controlled transmitter and 100 real-world traces of commercial technologies. ***For the controlled traces***, we first collect 1400 traces using all possible scan configurations across the five platforms (Table I), in an indoor and outdoor setting with a dwell time of 20 seconds. We use these traces for our transmitter and occupancy detection applications. We next collect 300 traces for the modulation recognition task using all the possible scan configurations across the three RTL platforms (Table I) indoors with a dwell time of 5 seconds. Section VI justifies the use of RTL sensors alone to collect modulation recognition traces. ***For the real-world traces***, we collect samples at 99.3MHz (FM radio) and in a TVWS testbed at 563MHz. All FM traces are collected indoors with all five sensors with a sampling rate of 1MHz, gain of 30dB (RTL), 40dB (USRP), and a dwell time of 20 seconds. The TVWS traces are collected outdoors using the battery-powered RTL-Laptop and USRP-Laptop sensors with a sampling rate of 2MHz (RTL) and 8MHz (USRP), a gain of 30dB (RTL), 40dB (USRP) and a dwell time of 20 seconds. We use iPerf to generate UDP traffic in the TVWS network by injecting 4 Mbits at 1 second intervals. For each trace, we calculate PSDs from the stored IQ data using a Python script.

B. Veracity

1) Definition: Veracity quantifies the retention of IQ samples. We define veracity (V) as the ratio between the number of expected samples N, and the actual collected samples N':

$$V = \frac{N'}{N} \tag{1}$$

The number of expected samples N depends on the sensor configuration as follows: $N = \Delta_t * f_s$, where Δ_t is the dwell time and f_s is the sampling rate of the sensor measured in samples per second (sps). A sensor dwelling for 1 second at a sampling rate of 1 Msps will collect 1M samples.

In reality, however, spectrum sensors may fail to obtain the number of expected samples N due to hardware limitations (§III), which results in the actual obtained sample count N' being smaller than the expected N. As the number of actual collected samples decreases, so does the quality of the collected spectrum scan, as its ability to capture all spectrum activity diminishes. The measure of veracity varies between 0 and 1, where 0 is an extreme (and arguably unrealistic) case where no samples are saved, whereas with 1, all samples are saved. While veracity is computed on IQ samples it quantifies the fidelity of both IQ and PSD data, as demonstrated in §VI.

2) Effects of sensor properties and configuration on veracity: Sensor properties such as the ADC/DAC resolution, buffer size, and the SDR-Host link speed impact veracity. The used sampling rate, which determines the volume of data transferred between the radio and the host, also affects veracity. We study the effects of these properties on IQ samples stored as binary files. Each sample is stored as an 8-byte number [48]. To calculate the number of collected IQ samples in a file, we measure the size of the file in bytes and divide by 8. We calculate the veracity of the trace as per Eqn. (1). Fig. 2 presents results across sensors. As the sample rate increases, the veracity declines for all platforms. This is due to buffer overflows resulting from the increased rate of symbols transmitted from the SDR to the host. Platforms that use low-capability hosts (RTL-Pi and USRP-Laptop) suffer significant deterioration in veracity compared to their higher-speed counterparts. 27% of the sample rates fall below 0.8 veracity, of which 66% are due to RTL-Pi and USRP-Laptop.

C. Ambiguity

An arbitrary spectrum trace in a frequency band with active transmitters might capture some samples that represent noise and others that represent transmitter activity. An example trace is presented in Fig. 3 (left), where lighter values represent transmitter activity, while darker values represent noise. Transmitters whose signal is low and close to the sensor's noise floor, can be ambiguously interpreted as noise. This relationship between the noise floor of a sensor and the level of a transmitter's signal will affect transmitter detection.

1) Definition: We develop a metric, dubbed *ambiguity (A)*, which quantifies the likelihood that transmitter samples in a scan are confused with noise. Given a spectrum trace $p(f, t)$ with $N = T * F$ samples we model the data as a Gaussian Mixture Model (GMM) [51] with two components: one for noise $G_N(\mu_N, \sigma_N)$ and one for non-noise values $G_T(\mu_T, \sigma_T)$.

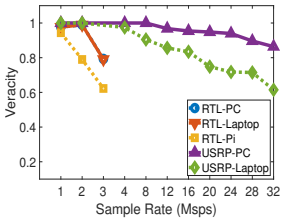

Fig. 2: Data veracity across the five platforms with increasing sampling rate.

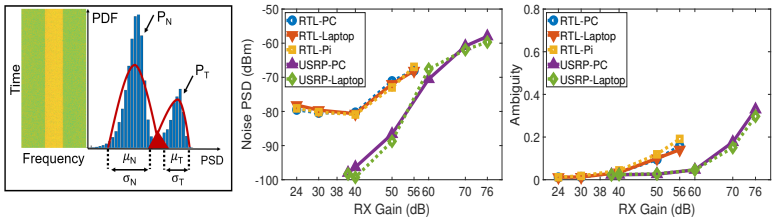

Fig. 3: Example trace (left) left pane with corresponding GMM model right pane. Noise floor across platforms as a function of the receiver gain of the sensor (middle). Ambiguity of the collected traces as a function of the receiver gain (right).

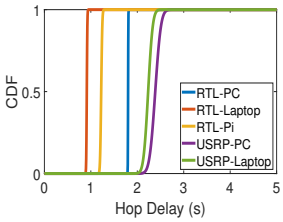

Fig. 4: Distribution (CDF) of hop delay time across all the five platforms.

Given these two distributions, for each measured sample p_f^t we can calculate the posterior probability that p_f^t is associated with each of the two components, as follows: $p_N(p_f^t) = \frac{1}{\sigma_N\sqrt{(2\pi)}} e^{(\frac{(p_f^t-\mu_N)^2}{2\sigma_N^2})}$ and $p_T(p_f^t) = \frac{1}{\sigma_T\sqrt{(2\pi)}} e^{(\frac{p_f^t-\mu_T)^2}{2\sigma_T^2})}$. Intuitively, these two probabilities quantify the likelihood that a sample arises from noise/transmitter activity. We label a sample as *ambiguous* if both its association probabilities are non-zero, i.e. if $p_N(p_f^t) > 0$ and $p_T(p_f^t) > 0$. Thus, the *ambiguity* of a trace $p(f,t)$ is defined as the fraction of ambiguous samples $\hat{N}$ from all collected samples N:

$$A = \frac{\hat{N}}{N} \tag{2}$$

A varies between 0 and 1. A trace with ambiguity of 0 will have a clear separation between noise and non-noise values such as in high SNR settings, and should, be easy to characterize. A trace with ambiguity close to 1, will have a small margin between noise and non-noise samples (i.e. low SNR), and will be hard to characterize. Ambuguity can be calculated either directly with PDS data or with the amplutude of `IQ` samples, and is thus applicable to both data streams. Fig. 3 (left) illustrates the overlap between sample distributions that model noise and non-noise signals for an example PSD trace. The blue bars represent the distribution of the empirical PSDs, the red line presents the fitted GMM $G_N(\mu_N, \sigma_N)$ and $G_T(\mu_T, \sigma_T)$. Highlighted red are the ambiguous PSD values.

GMM is a general unsupervised clustering approach, which has been used to mine the number of transmitters in a spectrum trace [52; 53]. In our work, we apply it to discern between transmitter and non-transmitter measurements in real-world traces with no ground truth. This allows us to identify data points that are ambiguous and determine the likelihood of a false detection in a trace. Additionally, we emphasise that our ambiguity measure is different than the classical SNR. SNR is defined as a comparison of the level of a desired signal to the level of background noise. In contrast, ambiguity determines the likelihood that a measured signal will be misclassified.

2) How we use GMM: GMM takes as an input (i) the data $p(f,t)$ to be modeled, (ii) a number of distributions K to be fitted over the data and (iii) a guess of the mean, standard deviation and weight (μ, σ, w) of each of the distributions. GMM then uses an Expectation Maximization algorithm (EM) to refine this guess, by maximizing the likelihood that the outputted model $(\hat{\mu}, \hat{\sigma}, \hat{w})$ represents the data correctly. While the above approach is fully unsupervised, and thus, does not require prior knowledge of the noise and transmitter distributions, the goodness of the GMM fit often depends on the setting of the initial guess. We study the effects of the initial guess on ambiguity in Fig. 5. Fig. 5 (top) explores the effects of the noise guess. The x-axis represents the difference Δ_{μ_N} between the means of the empirical and the guessed noise component. We fix the non-noise guess to be 20dB higher than the empirical noise mean. Δ_{μ_N} is negative when the guess is lower than the empirical mean, and positive when the guess is higher than the empirical mean. Red and blue represent USRP- and RTL-based sensors, respectively. The ambiguity measure is nearly zero and stable for a wide range of offsets from the empirical noise mean for both platforms: -8dB to 20dB for RTL and -5dB to 20dB for USRP. Similar trends can be observed on the counterpart graph that studies ambiguity as a function of the transmitter guess (Fig. 5 (bottom)). We fix the noise guess to be equal to the empirical noise mean and vary the difference between the empirical transmitter mean and the guessed mean. The ambiguity measure is stable for a wide range of guesses for both RTL and USRP: -7 to 20dB for RTL and -1 to 20dB for the USRP. These results emphasize the robustness of ambiguity to GMM's initialization.

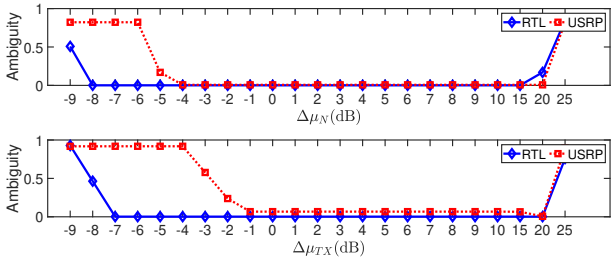

Fig. 5: Ambiguity across different noise (top) and transmitter (bottom) offsets is stable within a wide range for both SDRs.

To refine our GMM noise guess we profile a sensor's noise floor in the wild. Traditionally, noise floor is profiled in an anechoic chamber to ensure that measurements are not tainted by interference. Since our goal is to automatically profile sensors without expert input or facilities, we propose an alternative approach that utilizes spectrum white spaces for coarse noise floor profiling. Various spectrum bands can be used for this purpose, including TVWS, which are documented

in a public database [50], or protected spectrum used by radio astronomy and passive remote sensing [54]. For a noise benchmark trace $p_N(f,t)$ collected in one of these bands, μ_N and σ_N are calculated as the mean and standard deviation of $p_N(f,t)$. The so obtained μ_N and σ_N can then be supplied as domain-informed input to GMM.

Having profiled a sensor's noise floor, we use GMM as follows. We model a collected trace $p(f,t)$ as two distributions (i.e. $K = 2$): one arising from noise and one arising from transmitter values. Our initial guess for the noise distribution (μ_N, σ_N, w_N) is set as per the noise floor profiling method discussed in the previous paragraph. The non-noise guess (μ_T, σ_T, w_N) is set as $\mu_T = \mu_N + 20dB$ and $\sigma_T = \sigma_N$. The weights w_N and w_T are each set to 0.5.

3) Effects of sensor properties and configuration on ambiguity: Ambiguity is affected by a sensor's noise floor, the noise floor stability across the sensor's tunable range, the receiver gain and the transmitter signal strength. We explore these factors in turn. We start with the variance of the noise floor of RTL and USRP radios across their entire tunable range in indoor and outdoor settings and in different parts of the day and days of the week (figure omitted for space). The noise fluctuations do not exceed 2dB across both sensors. At the same time ambiguity is robust within a much larger range of deviation of the input guess Fig. 5. We also study the effects of the sensor's receiver gain on the noise floor (Fig. 3 (middle)) and ambiguity (Fig. 3 (right)). As the receiver gain increases, so do the noise floor and ambiguity. This highlights the importance of operating within a safe gain regime.

D. Intermittency

Sweep-based spectrum sensing (§III) introduces intermittency in spectrum scans, which in turn, may affect data quality. Thus, we propose a metric to quantify the intermittency of a spectrum scan as a function of the sensor's sweeping configuration. We define intermittency as the *revisit time (ΔT)* elapsed between consecutive revisits of a given band:

$$\Delta T = \sum_{i=1}^{N-1} (\tau + \Delta t_i) \tag{3}$$

Here N is the number of sweep steps $\Delta f_i, [Hz]$ that a sensor must complete in order to scan the entire target frequency range $\Delta F, [Hz]$. Δt_i is the amount of time a sensor dwells on a given sweep step Δf_i (also known as *dwell time*). Finally, τ is the hop delay defined as the time from the start of the radio reconfiguration until it records the first sample. τ factors in the time it takes to configure the hardware, the local oscillator to settle, the operating system scheduling, and the time taken at the application level to start producing data.

1) Effects of sensor properties and configuration on intermittency: While N and Δt_i are apriori known configuration parameters, the hop delay τ is not deterministic, because the time to configure, schedule and produce samples will vary depending on the SDR-Host platform. Fig. 4 presents the hop delay over 1000 runs for the five platforms. It is different across platforms, owning to the different hardware configuration, processors, and USB link speeds. In addition, the hop delay varies from run to run for the same platform, as indicated by the deviation of measured values for each platform. All RTL sensors had lower deviation of their hop delays compared to the USRP-based sensors. Finally, all RTL-based platforms were faster to re-initialize compared to their USRP-based counterparts, due to their simpler hardware.

We note that our traces were collected without sweeping, as our sensors dwelled for 20 seconds per run. Thus no hop delays were experienced. In order to study the effects of hop delay, we excised portions of each trace at a random location, according to the measured platform hop delays in Fig. 4.

V. **VIA** AS APPLICATION ACCURACY PREDICTOR

VIA quantifies the quality of spectrum data and thus, can be used as a performance predictor for a variety of spectrum analysis tasks. We explore **VIA**'s predictive capabilities with three such applications (i) transmitter detection, (ii) occupancy detection, and (iii) modulation recognition as detailed below.

A. Target Applications

1) Transmitter Detection (TD). We employ a recent unsupervised transmitter detection algorithm from the literature called AirVIEW [28] that is robust in low SNR regimes. It takes as an input an array of PSD measurements over time for a set of frequencies (sweep). It finds the sweep's wavelet decomposition represented as a binary tree and then calculates the multiscale product of lossy reconstructions of the underlying PSD. It then thresholds the multiscale product to determine idle and occupied frequency ranges. Finally, occupied ranges are reconciled into a single transmitter over time based on the alignment across consecutive sweeps. Given groundtruth for transmitter activity, the accuracy of transmitter detection is calculated as the bidirectional Jaccard similarity between actual and detected active time-frequency blocks [28].
2) Occupancy Detection (OD). We employ edge detection in order to identify idle and occupied bands [55]. The method takes as input an array of PSD measurements over time. A threshold is applied to detect the rising and falling edges of the occupied bins for each PSD sweep. To calculate the threshold, we obtain the max frequency hold over time. We find the minimum and maximum value, take the difference and divide in half. We calculate accuracy as the overlap between expected and detected active frequency bins. We average the so-calculated overlaps across all sweeps with detected activity.
3) Modulation Recognition (MR). For this application, we employ the methodology from [41] assuming 100% overlap between the transmitter's and the sensor's bandwidths. The methodology employs dictionary learning with local sequential patterns in `IQ` timeseries to determine the modulation of a signal. We calculate the accuracy of this application as the fraction of total samples with a correctly identified modulation.

B. **VIA** as a Performance Predictor

To demonstrate **VIA**'s potential as a performance predictor, we model the relationship between **VIA** and the accuracy of

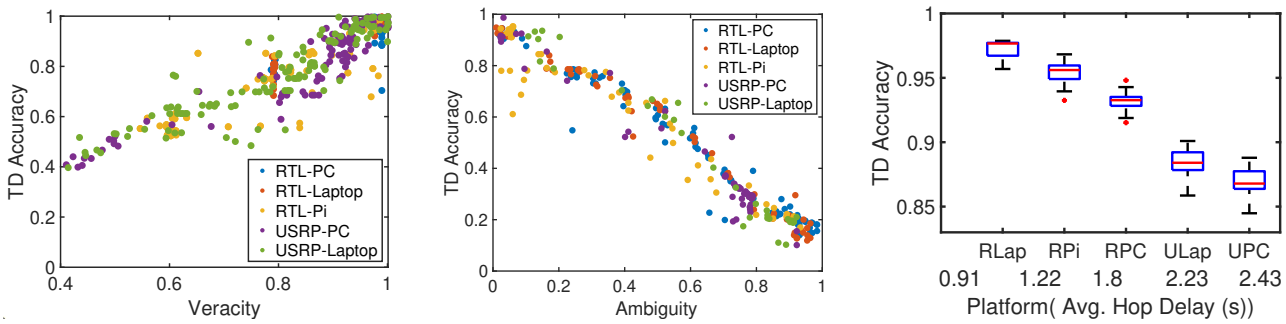

Fig. 6: Demonstration of **VIA** as a performance predictor. (left) TD accuracy as a function of observed data veracity (middle) TD accuracy as a function of observed data ambiguity. (right) TD accuracy as a function of hop delay time.

the above applications. We use two prediction models: ridge regression [56] and a neural network [57]. Ridge regression is an extension of linear regression and is highly effective in eliminating multicollinearity. The neural network is a feedforward network which consists of an input layer with a fully connected hidden layer (five neurons), and the output layer.

For each trace, we calculate veracity, intermittency and ambiguity (**VIA** vector). Veracity is calculated using Eqn 1. Intermittency is represented by the hop delay time described in §IV-D. Ambiguity is calculated using the methodology described in §IV-C and using Eqn 2. The dataset captures a full range of **VIA** values. Finally, we employ TD, OD and MR applications to all traces and calculate the respective accuracy as described in §V-A. The input to our models is the **VIA** vector (i.e. the model inputs) and the output of the model is accuracy (i.e. the model output) of the targeted application. Our prototype framework is implemented in MATLAB (**VIA** vector calculation) and Python (prediction models).

C. Case Study

To showcase **VIA**'s relationship with application accuracy, we study the relationship between the **VIA** vectors and the performance of transmitter detection (TD). We seek to demonstrate two core **VIA** characteristics: (i) its predictive power for spectrum tasks and (ii) the fact that it is sensor-agnostic. Fig. 6 plots TD accuracy vs. each of the three indicators.

Fig. 6 (left) shows accuracy as a function of veracity. For this result, we consider all traces with ambiguity of 0 and no hop delay in order to isolate the effects of veracity. As a reminder, lower veracity means more dropped samples, which adversely affects the accuracy. We see a clear trend of decreasing accuracy as the veracity decreases, regardless of the sensor platform. Thus, veracity is a strong predictor for accuracy regardless of the platform.

Fig. 6 (middle) shows the relationship between ambiguity and TD accuracy. For this result, we consider all traces with veracity of 1 and no hop delay in order to isolate the effects of ambiguity. As a reminder, higher ambiguity makes transmitters harder to detect, thus leading to reduced accuracy. Accordingly, we see that as the ambiguity grows, the accuracy decreases across all sensor platforms.

Finally, we showcase the effects of intermittency on TD accuracy (Fig. 6 (right)). For this result, we consider all the traces with ambiguity of 0 and veracity of 1 in order to isolate the effects of intermittency. The x-axis presents the five platforms in increasing order of average intermittency. Boxplots present the accuracy distribution for each platform. Lower intermittency results in higher overall accuracy, thus the intermittency, can serve as a predictor for accuracy.

VI. Evaluation

To evaluate **VIA**'s ability to predict application performance, we utilize the regression and neural network models described in §V-B. Furthermore, we focus on the three applications–transmitter detection (TD), occupancy detection (OD) and modulation recognition (MR) detailed in §V-A. We first explore **VIA**'s predicting performance for known platforms then study the transferability of **VIA** models across unknown platforms and propagation environments. Finally, we explore the predictive power of individual **VIA** vectors and analyze the effects of the training data size.

Evaluation metrics and implementation. For TD and OD, which draw on PSD data, the models are trained with 150 samples of controlled traces and tested with 25 samples of controlled or real-world traces, all from the five platforms with 10-fold validation. For MR, which uses `IQ` data from the RTL-based platforms, we train on 150 samples of controlled traces and tested with 25 of controlled traces. In all experiments, we report the mean squared error (MSE), which is the average squared error between the model's prediction and the target value. We select MSE as it is a common standard metric for both prediction models. As a reminder, our predicted and target value is TD/OD/MR accuracy, which varies between 0 and 1. Thus, MSE close to 0 corresponds to a good predictive model.

A. VIA Performance for Known Platforms

We begin by exploring **VIA**'s ability to predict TD and OD accuracy across known platforms. A platform is "known" if it was included in the model training process. For this experiment we train on a mix of all platforms. Our results are presented in Fig. 7 (left) across six testing scenarios, comparing both applications and models. "All" includes equal representation of testing samples from the five platforms. For

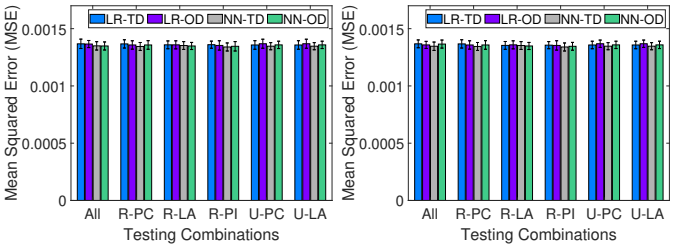

Fig. 7: MSE for known platforms. Testing on controlled traces (left) and testing on real-world traces (right).

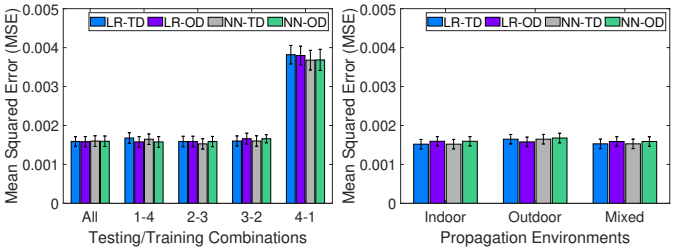

Fig. 8: MSE for unknown platforms (left) and for known/unknown propagation environments (right).

the remaining five cases, all test samples are drawn from a single platform. All six cases achieve high and stable performance regardless of the target sensor, application and prediction model, as indicated by the consistently low MSE. With a mix of testing samples our average MSE is 0.0013 across all testing scenarios.

We next evaluate, **VIA**'s prediction performance on real-world traces. For this experiment, we train on the controlled traces with all five platforms and test using the FM radio and TVWS real-world collected traces. Unlike the controlled traces collected with our testbed, for which we have groundtruth for both bandwidth and temporal duration, for the real-world FM/TVWS traces we only know their bandwidth (based on transmitter specification) but not duration. Thus, for real-world data we report accuracy in bandwidth detection. Our results are presented in Fig. 7 (right) with testing on a mix of all platforms ("All") and individual platforms. All six cases achieve high and stable performance as indicated by the low MSE scores. With the real-world testing samples our average MSE score is less than 0.0015 across all testing scenarios. These results show that **VIA** models transfer across transmitter settings and do not require training on different technologies, as we were able to train on indoor controlled traces and test on wide-area outdoor traces.

B. Transferability of VIA Models

Next, we explore **VIA**'s ability to predict TD and OD performance for unknown platforms and unseen multipath environments. A platform is unknown if it was not included in training. We generate all possible combinations $\binom{n}{k}$, where n is the number of training platforms, k is the number of testing platforms and $n+k=5$. Fig. 8 (left) represents MSE across all combinations. In "All" both the training and the testing pool are comprised of samples from all five platforms. For the remaining four cases, we exclude an increasing number of platforms from the training set, i.e. "1-4" trains on 4 platforms and uses the fifth for testing, "2-3" trains on three and tests on the remaining two and so on. The presented results are average over all possible combinations for a given train-test ratio. Overall, the first 4 combinations have a low and consistent MSE (an average of 0.0014 or better with a small variance). The rightmost combination "4-1" scores just below 0.004 with increased variance, which indicates that as we limit the amount of training platforms and increase the unknown testing platforms, **VIA**'s performance declines. Upon closer examination of the individual "4-1" combinations, we found that training on a single RTL sensor, while testing on a mix of RTL and USRP, gains bad performance. This is due to the limited set of sensor configurations that can be explored with an RTL compared to a USRP. As a result, we see much fewer realizations of the **VIA** vectors than we would with a sensor with many more possible configurations such as the USRP. Thus, in order to be a good training candidate, a platform should provide a full range of **VIA** realizations.

We also evaluate **VIA**'s transferability across multipath propagation environments. Fig. 8 (right) presents results for three cases: *Indoor*, for which training and testing is on the indoor data, *Outdoor*, with training and testing on the outdoor data and *Mix*, with training on the indoor and testing on the outdoor data. All three scenarios maintain a low MSE score (less than 0.0015) with low variance. These results hold for both known (*Indoor*/*Outdoor*) and unknown (*Mixed*) propagation environments, across spectrum tasks and prediction models. Particularly noteworthy is the *Mixed* scenario, showing that **VIA** performs well even when sensors are only profiled in indoor settings. This can lower the barrier in data collection by allowing sensor profiling in indoor environments.

Our evaluation so far demonstrates that ***VIA*** *is platform-agnostic*, can successfully be transferred from one platform to another and from indoor to outdoor scenarios. Furthermore, **VIA** can predict application accuracy even when the training dataset contains fewer platforms than the testing data.

C. VIA with IQ Data for Modulation Recognition

We next explore **VIA**'s predictive capabilities for MR applications using `IQ` data. We limit this experiment to data collected on the RTL-based sensors for two reasons. First, existing MR algorithms require that a transmitter's bandwidth be scanned alone and at 100% or less overlap between the bandwidth of the sensor and the transmitter. Thus, modulation recognition cannot be performed on spectrum scans that encompass sideband noise. Additionally, our USRP-based transmitter can reliably run with a bandwidth up to 4Mhz, after which it starts dropping samples. Since we want to avoid transmitter-induced imperfections in our scans, we only run our transmitter with a bandwidth up to 3Mhz. This inherently limited the bandwidths we could use on the sensor side to 3MHz as well. USRP platforms do not suffer performance deterioration at 3MHz bandwidth, while RTLs begin to drop samples at that bandwidth (§IV). Thus, to ensure 100% overlap

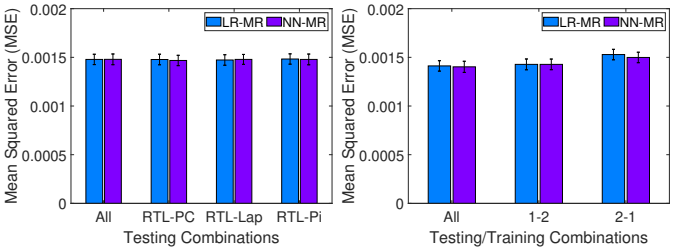

Fig. 9: MSE for known platforms (left) and unknown platforms (right) for MR task.

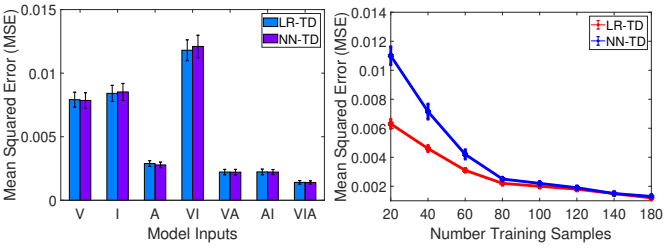

Fig. 10: Analysis of **VIA** input importance (left) and training data size (right).

between the sensor and the transmitter bandwidths, while allowing for some imperfections in the spectrum scan to occur, we limit this experiment to the RTL-based sensors.

Our results are presented in Fig. 9. On the left we have MSE across known platforms. Here, the training set includes all RTL-based sensor platforms. "All" includes equal representation of testing samples from the three platforms. For the remaining three cases, all test samples are drawn from a single platform. All cases achieve high and stable performance as indicated by the low MSE (under 0.0015). On the right, we have MSE across unknown platforms. Once again, "All" includes equal representation of testing samples from the three platforms. For the remaining 2 cases, we exclude an increasing number of platforms from the training set. Overall, all combinations have a low and consistent MSE (0.0015 or under). These results demonstrate that ***VIA*** *can reliably predict application accuracy for a variety of applications and data types*, including modulation recognition from `IQ` traces.

D. Analysis of Input Importance and Input Size Requirements

We now evaluate the importance of the three **VIA** vectors in predicting the performance of the PSD-based transmitter detection task. Our goal is to determine whether some **VIA** vectors hold higher predictive power than others, and justify the use of all three vectors in combination as opposed to individually. To that end, we modify the inputs to our models to be (i) the individual **VIA** vectors, (ii) pairs of **VIA** vectors and (iii) the combination of all three and evaluate the performance of TD with both models. Our results are presented in Fig. 10 (left). Veracity and intermittency alone achieve an MSE score just under 0.009. The combination of these exacerbates the performance (over 0.012). The two combinations that include ambiguity achieve an MSE score of 0.0022, which is better than ambiguity alone (0.0027). However, neither of these two combinations reach the **VIA** score of 0.0013. Thus, all of the **VIA** vectors are important predictors for application accuracy.

Finally, we study the effects of the training data size on **VIA**'s performance. This will inform the amount of required traces and characterizations. For this experiment, we increase the size of the training samples from 20 to 180 in increments of 20. The training and testing samples are taken from all five platforms. Fig. 10 (right) shows our results across all training sizes for both models and the transmitter detection task. We observe that with a training pool of 20-60 samples, the LR model performs better, likely due to the higher training data requirement of neural network based models. As the training pool increases, the MSE score improves for the NN model and converges with the LR model. These results inform our choice of 150 training samples for the previously presented results. They also have important practical implications, as they demonstrate that **VIA** can reliably predict application accuracy with as few as 100 controlled spectrum characterizations.

VII. Discussion and Conclusion

Understanding and modeling the effects of spectrum sensors on data quality is instrumental to measurements that can objectively capture spectrum activity. The state-of-the-art in spectrum measurement, however, lacks methods to catalogue sensor properties and attribute them to data quality and application performance.

To address this issue we develop **VIA**, a framework that can profile the fidelity of spectrum traces while accounting for the sensor configuration and radio environment, and link data quality with the performance of spectrum analytics. **VIA** benchmarks data across three vectors: `veracity`, `intermittency` and `ambiguity`. Using an extensive experimental campaign with five RTL- and USRP-based sensor platforms we demonstrate that **VIA** is a strong predictor for spectrum analytics' performance. We consider a variety of applications — transmitter and occupancy detection, and modulation recognition — that use both power spectrum density traces and raw `IQ` data. **VIA** can benchmark data fidelity for arbitrary and unknown platforms with limited supervision. **VIA** has the potential to lower the entry barrier into spectrum research and measurements for non-experts by providing an automated approach to sensor and data quality benchmarks.

While our current efforts focus on **VIA**'s ability for spectrum analytics prediction, an extension of this framework for sensor selection is worth exploring. Furthermore, expanding our dataset to include data from new sensor platforms, NLOS scenarios, and testing in real-world environments will also be a priority. Finally, the current **VIA** design accounts for the impact on data quality of the most influential sensor capabilities. Further exploration of sensor imperfections such as I/Q imbalances, automatic gain control, front end saturation, and RF nonlinearity may inform additional **VIA** fidelity vectors.

REFERENCES

[1] "The Importance of Artificial Intelligence and Data for the TelecommunicationsIndustryand the FCC." https://www.fcc.gov/sites/default/files/fcc_aiwg_2020_whitepaper_final.pdf.

[2] Illinos Institute of Technology, "NSF Workshop on Spectrum Measurement Infrastructures, Chicago, IL, USA." http://www.cs.albany.edu/~mariya/nsf_smsmw/, April 6-7 2016.

[3] T. Cooklev, R. Normoyle, and D. Clendenen, "The VITA 49 Analog RF-Digital Interface," *IEEE Circuits and Systems Magazine*, vol. 12, pp. 21–32, Fourthquarter 2012.

[4] "Ettus research. compressed header (chdr) radio transport protocol.." https://files.ettus.com/manual/page_rtp.html.

[5] "Signal Metadata Format (SigMF)." https://github.com/gnuradio/SigMF.

[6] "IEEE 802.22.3 - Standard for Spectrum Characterization and Occupancy Sensing." https://standards.ieee.org/develop/project/802.22.3.html.

[7] A. Nika, Z. Zhang, X. Zhou, B. Y. Zhao, and H. Zheng, "Towards commoditized real-time spectrum monitoring," HotWireless'14, (Maui, Hawaii, USA), 2014.

[8] A. Nika, Z. Li, Y. Zhu, Y. Zhu, B. Y. Zhao, X. Zhou, and H. Zheng, "Empirical validation of commodity spectrum monitoring," SenSys'16, (New York, NY, USA), 2016.

[9] A. Chakraborty, M. S. Rahman, H. Gupta, and S. R. Das, "Specsense: Crowdsensing for efficient querying of spectrum occupancy," INFOCOM'17, (Atlanta, GA, USA), 2017.

[10] M. Dasari, M. B. Atique, A. Bhattacharya, and S. R. Das, "Spectrum protection from micro-transmissions using distributed spectrum patrolling," PAM'19, 2019.

[11] A. Bhattacharya, C. Zhan, A. Maji, H. Gupta, S. R. Das, and P. M. Djurić, "Selection of sensors for efficient transmitter localization," INFOCOM'20, (Virtual Conference), 2020.

[12] M. A. McHenry, P. A. Tenhula, D. McCloskey, D. A. Roberson, and C. S. Hood, "Chicago Spectrum Occupancy Measurements and Analysis and a Long-term Studies Proposal," Proc. of TAPAS Conference, Aug 2006.

[13] "Spectrum Observatory." https://observatory.microsoftspectrum.com/.

[14] M. Souryal, M. Ranganathan, J. Mink, and N. El Ouni, "Real-time centralized spectrum monitoring: Feasibility, architecture, and latency," DySPAN'15, (Stockholm, Sweden), 2015.

[15] S. Roy, K. Shin, A. Ashok, M. McHenry, G. Vigil, S. Kannam, and D. Aragon, "Cityscape: A metro-area spectrum observatory," ICCCN'17, (Vancouver, Canada), 2017.

[16] B. Van den Bergh, D. Giustiniano, H. Cordobés, M. Fuchs, R. Calvo-Palomino, S. Pollin, S. Rajendran, and V. Lenders, "Electrosense: Crowdsourcing spectrum monitoring," DySPAN'17, (Baltimore, MD, USA), 2017.

[17] D. Pfammatter, D. Giustiniano, and V. Lenders, "A software-defined sensor architecture for large-scale wideband spectrum monitoring," IPSN'15, (Seattle, WA, USA), 2015.

[18] "Spectrum Characterization and Occcupancy Sensing Transfer Specification." https://github.com/NTIA/sigmf-ns-ntia/.

[19] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys Tutorials*, vol. 11, pp. 116–130, Jan 2009.

[20] S. D. Jones, E. Jung, X. Liu, N. Merheb, and I.-J. Wang, "Characterization of Spectrum Activities in the U.S. Public Safety Band for Opportunistic Spectrum Access," DySPAN'07, (Washington, DC, USA), 2007.

[21] Z. Tian and G. Giannakis, "A wavelet approach to wideband spectrum sensing for cognitive radios," CROWNCOM'06, (Mykonos Island, Greece), June 2006.

[22] L. Yang, W. Hou, L. Cao, B. Y. Zhao, and H. Zheng, "Supporting Demanding Wireless Applications with Frequency-Agile Radios," NSDI'10, (San Jose, California), 2010.

[23] Z. Tian and G. B. Giannakis, "Compressed sensing for wideband cognitive radios," ICASSP'07, (Honolulu, HI), April 2007.

[24] E. Candes and M. Wakin, "An introduction to compressive sampling," *Signal Processing Magazine, IEEE*, vol. 25, pp. 21–30, March 2008.

[25] J. Laska, W. Bradley, T. W. Rondeau, K. E. Nolan, and B. Vigoda, "Compressive sensing for dynamic spectrum access networks: Techniques and tradeoffs," DySPAN'11, (Aachen, Germany), May 2011.

[26] A. Palaios, J. Riihijärvi, and P. Mähönen, "From paris to london: Comparative analysis of licensed spectrum use in two european metropolises," DySPAN'14, (McLean, VA, USA), 2014.

[27] D. Chen, S. Yin, Q. Zhang, M. Liu, and S. Li, "Mining spectrum usage data: A large-scale spectrum measurement study," *IEEE Transactions on Mobile Computing*, vol. 11, June 2012.

[28] M. Zheleva, P. Bogdanov, T. Larock, and P. Schmitt, "Airview: Unsupervised transmitter detection for next generation spectrum sensing," INFOCOM'18, (Honolulu, HI, USA), 2018.

[29] A. Scalingi, D. Giustiniano, R. Calvo-Palomino, N. Apostolakis, and G. Bovet, "A framework for wireless technology classification using crowdsensing platforms," INFOCOM'23, 2023.

[30] A. Chakraborty, A. Bhattacharya, S. Kamal, S. R. Das, H. Gupta, and P. M. Djuric, "Spectrum patrolling with crowdsourced spectrum sensors," INFOCOM'18, (Honolulu, HI, USA), 2018.

[31] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET communications*, vol. 1, no. 2, pp. 137–156, 2007.

[32] O. A. Dobre, Y. Bar-Ness, and W. Su, "Higher-order cyclic cumulants for high order modulation classification," MILCOM'03, (Boston, MA), 2003.

[33] L. Han, F. Gao, Z. Li, and O. A. Dobre, "Low complexity automatic modulation classification based on order-statistics," *IEEE Transactions on Wireless Communications*, vol. 16, no. 1, pp. 400–411, 2017.

[34] K. Karra, S. Kuzdeba, and J. Petersen, "Modulation recognition using hierarchical deep neural networks," DySPAN'17, (Baltimore, MD, USA), 2017.

[35] J. L. Xu, W. Su, and M. Zhou, "Likelihood ratio tests for modulation classification," MILCOM 2000, (Los Angeles, CA), 2000.

[36] A. Swami and B. M. Sadler, "Hierarchical digital modulation classification using cumulants," *IEEE Transactions on communications*, vol. 48, no. 3, pp. 416–429, 2000.

[37] H. Abuella and M. K. Ozdemir, "Automatic modulation classification based on kernel density estimation," *Canadian Journal of Electrical and Computer Engineering*, vol. 39, no. 3, pp. 203–209, 2016.

[38] K. Zhang, E. L. Xu, H. Zhang, Z. Feng, and S. Cui, "Data driven automatic modulation classification via dictionary learning," *IEEE Wireless Communications Letters*, 2018.

[39] G. Lu and et.al., "Modulation recognition for incomplete signals through dictionary learning," WCNC 2017, (San Francisco, CA, USA), 2017.

[40] Z. Zhu and A. K. Nandi, "Blind digital modulation classification using minimum distance centroid estimator and non-parametric likelihood function," *IEEE Transactions on Wireless Communications*, vol. 13, no. 8, pp. 4483–4494, 2014.

[41] W. Xiong, P. Bogdanov, and M. Zheleva, "Robust and efficient modulation recognition based on local sequential iq features," INFOCOM'19, (Paris, France), 2019.

[42] W. Xiong, L. Zhang, M. McNeil, P. Bogdanov, and M. Zheleva, "Symmetry: Exploiting mimo self-similarity for under-determined modulation recognition," *IEEE Transactions on Mobile Computing*, 2021.

[43] W. Xiong, P. Bogdanov, and M. Zheleva, "Modeless: Modulation recognition with limited supervision," SECON'21, (Virtual Conference), 2021.

[44] E. Perenda, S. Rajendran, G. Bovet, M. Zheleva, and S. Pollin, "Contrastive learning with self-reconstruction for channel-resilient modulation classification," INFOCOM'23, 2023.

[45] M. Zheleva, T. Larock, P. Schmitt, and P. Bogdanov, "Airpress: High-accuracy spectrum summarization using compressed scans," (DySPAN'18), (Seoul, Korea), 2018.

[46] "USRP." https://www.ettus.com/.

[47] "RTL-SDR." http://www.rtl-sdr.com/.

[48] "GNU Radio." https://www.gnuradio.org/.

[49] "MP Antenna." https://www.mpantenna.com/.

[50] "TVWS-DB." https://usa.wavedb.com/.

[51] N. Bouguila and W. Fan, *Mixture Models and Applications*. Springer, 2020.

[52] M. Zheleva, R. Chandra, A. Chowdhery, A. Kapoor, and P. Garnett, "TxMiner: Identifying Transmitters in Real-World Spectrum Measurements," IEEE DySPAN'15, (Stockholm, Sweden), 2015.

[53] M. Z. Zheleva, R. Chandra, A. Chowdhery, P. Garnett, A. Gupta, A. Kapoor, and M. Valerio, "Enabling a nationwide radio frequency inventory using the spectrum observatory," *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, pp. 1–1, 2017.

[54] "NRAO." https://science.nrao.edu/facilities/vla/docs/manuals/obsguide/modes/line/.

[55] D. Waghule and R. Ochawar, "Overview on edge detection methods," in *2014 International Conference on Electronic Systems*, (Nagpur, India), January 2014.

[56] "Ridge." https://pytorch.org/.

[57] "PyTorch." https://scikit-learn.org/stable/.