This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2716936, IEEE Transactions on Mobile Computing

Enabling a Nationwide Radio Frequency Inventory Using the Spectrum Observatory

M. Zheleva[1], R. Chandra[2], A. Chowdhery[2,3], P. Garnett[2], A. Gupta[2], A. Kapoor[2], and M. Valerio[2]

[1]Department of Computer Science, University at Albany SUNY, *mzheleva@albany.edu*
[2]Microsoft Research, {*ranveer, paulgar, akapoor, mvaler*}*@microsoft.com, anoop.gupta@outlook.com*
[3]Department of Electrical Engineering, Princeton University, *aakanksha@princeton.edu*

Abstract—Knowledge about active radio transmitters is critical for multiple applications: spectrum regulators can use this information to assign spectrum, licensees can identify spectrum usage patterns and provision their future needs, and dynamic spectrum access applications can efficiently pick operating frequency. To achieve these goals we need a system that continuously senses and characterizes the radio spectrum. Current measurement systems, however, do not scale over time, frequency and space and cannot perform transmitter detection. We address these challenges with the *Spectrum Observatory*, an end-to-end system for spectrum measurement and characterization. This paper details the design and integration of the Spectrum Observatory, and describes and evaluates the first unsupervised method for detailed characterization of arbitrary transmitters called *TxMiner*. We evaluate TxMiner on real-world spectrum measurements collected by the Spectrum Observatory between 30MHz and 6GHz and show that it identifies transmitters robustly. Furthermore, we demonstrate the Spectrum Observatory's capabilities to map the number of active transmitters and their frequency and temporal characteristics, to detect rogue transmitters and identify opportunities for dynamic spectrum access.

Index Terms—Spectrum measurement, spectrum characterization, machine learning, Dynamic Spectrum Access, spectrum policy.

1 Introduction

We are faced with an increased need for additional RF spectrum to support the ever-growing demand for mobile data communications. However, nearly all the RF spectrum has been allocated for different purposes, e.g. TV, radio, cellular, radars, satellites, etc. Therefore, spectrum regulators worldwide are investigating the use of Dynamic Spectrum Access (DSA) techniques, such as in the TV white spaces or tiered access in 3.5 GHz of spectrum, to meet the additional demand. Using these techniques, mobile devices can send and receive packets over a frequency as long as they do not interfere with the licensed user of that frequency.

To identify new spectrum for DSA, the U.S. government, industry and spectrum regulators worldwide have endeavoured to create a large-scale spectrum inventory in order to determine the longitudinal spectrum usage at different locations [1]. Based on these measurements, spectrum regulators can open new portions of the spectrum for DSA [4], and new DSA technologies can be designed taking into account the characteristics of these bands. Such national spectrum inventory should answer various questions [1] including (i) how much spectrum is occupied/idle, (ii) how many transmitters occupy a given frequency band, and (iii) are they authorized to operate in this band. While the first question can be approached by simple estimation of power level in a given band, the other two questions require more elaborate analysis of spectrum occupancy. Such analysis needs to answer questions such as *are there more than one transmitters in a given band*, and *what are their received powers, operating frequencies, bandwidth and temporal characteristics*. Learning these characteristics from raw spectrum measurements is critical for improved policy and technological advances in the DSA domain.

Despite the need for deep understanding of spectrum occupancy, *there does not exist a platform* to create such nationwide spectrum usage footprint. This is primarily due to lack of scalable infrastructure for collection and processing of RF spectrum measurements. Traditionally, spectrum occupancy is analyzed via spectrum analyzers that capture large amounts of data. The latter poses challenges in scalable data storage. Furthermore, *the current approaches to mining and summarizing spectrum measurements are very limited*, making it hard to evaluate the collected spectrum data.

We address both these challenges in this paper. First, we present a novel RF measurement infrastructure dubbed the Spectrum Observatory[1] that harnesses the collective power of the spectrum research community to collect spectrum measurements. Spectrum analyzers hosted by various participants perform wideband measurements, from 10s of MHz to a few GHz, at different resolution bandwidths. A co-located PC processes the data and creates summaries before uploading them to the cloud. Unlike existing cloud-based spectrum measurement infrastructures [11, 16], our design is unique with its open nature, allowing wide community participation and creating a plethora of opportunities for research in spectrum measurement and management.

The second challenge, of identifying transmitters in spectrum data, is non-trivial. To illustrate this, let us consider the following trace collected by the Spectrum Observatory

1. https://observatory.microsoftspectrum.com/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2716936, IEEE Transactions on Mobile Computing

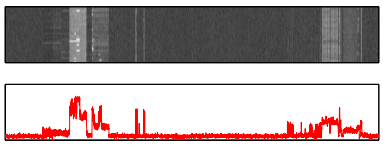


Fig. 1. Example of overlapping transmitters.

(Fig. 1). The top part of the figure plots power spectral density (PSD) measured over the course of 90 seconds between 700 and 900MHz. The bottom plot of Fig. 1 shows a maxhold of PSD in the entire frequency range; that is the maximum measured PSD value in each frequency bin. In some parts of the spectrum there exist more than one transmitters that occupy the same band in a time-division fashion. Direct analysis of the data in time-frequency domain is prone to errors due to the noisy nature of raw spectrum signals. Analysis of the maxhold, on another hand, can provide intuition of occupied fractions of this spectrum but hides the time-frequency characteristics of the individual transmitters. Thus, we need advanced spectrum characterization that goes beyond max-hold or direct time-frequency analysis. Current methods require prior knowledge of transmitter signatures and fine-grained spectrum measurements [7, 14], both of which are difficult to obtain in wide-band sweep-based spectrum sensing. Others [15] provide unsupervised separation of spectrum utilization patterns but do not cater to detailed transmitter characterization.

For detailed spectrum characterization, we design *TxMiner*, that identifies transmitters in raw spectrum measurements, even when the transmitter characteristics are not known and the spectrum sensing resolution is low. TxMiner leverages the phenomenon that fading of non-line-of-sight wireless signals follows a Rayleigh distribution, while noise follows a Gaussian distribution [5]. Thus, the raw spectrum samples can be modeled as a mixture of Rayleigh and Gaussian distributions. Based on this observation we design a machine learning algorithm that extracts Rayleigh and Gaussian sub-populations from a given RF signal population. Two challenges arise with such approach to transmitter characterization. First, the performance of our Rayleigh-Gaussian mixture model is dependent on the initialization of the model. To address this challenge, we design a multi-scale initialization scheme. Second, in order to extract frequency and temporal transmitter characteristics we design a post-processing technique. Thus, TxMiner is comprised of three critical components: (i) multi-scale initialization (§3.4), (ii) Rayleigh-Gaussian representation of raw spectrum measurements (§3.3) and (iii) post-processing for actual transmitter identification (§3.5). We evaluate TxMiner on spectrum measurements collected by the Spectrum Observatory, and on several controlled transmissions, and we find that it can accurately identify transmitters of different types including WiMax, TV & FM broadcasts, and proprietary DSA. We employ TxMiner to map the number of active transmitters and their bandwidths over a wide band from 30MHz to 6GHz, recognize rogue transmitters and identify opportunities for dynamic spectrum access.

This paper makes several key contributions: (i) we present the design and integration of the Spectrum Observatory, (ii) we design the first of its kind mechanism, called TxMiner, that can identify transmitters and their characteristics in raw spectrum measurements, (iii) we harness TxMiner to create a spectrum inventory through longitudinal, wideband analysis of traces collected by the Spectrum Observatory in the course of a year between 30MHz and 6GHz, and (iv) we demonstrate TxMiner's ability to detect rogue transmitters in raw spectrum scans and to quantify the opportunity for secondary access in licensed spectrum.

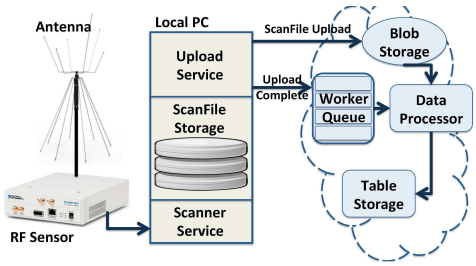


Fig. 2. The components of the Spectrum Observatory

This paper is organized as follows. §2 presents the design and integration of the Spectrum Observatory. In §3 we present TxMiner. We continue with evaluation in §4. In §5 we demonstrate the Spectrum Observatory's capability to create a nationwide spectrum inventory. We present related work in §6 and conclude in §7.

2 The Spectrum Observatory

The Spectrum Observatory provides a distributed spectrum monitoring platform and has been widely used by the community. It consists of two components as shown in Fig. 2: (i) local spectrum measurement equipment at various locations, and (ii) the storage and analysis component in the cloud. We note that previous work [11, 16] has designed cloud-based spectrum measurement architectures. The Spectrum Observatory is unique with its open nature, allowing anyone to plug a measurement station and/or pull data and spectrum usage results from the system. The inherent heterogeneity of measurement stations creates a plethora of opportunities for research in spectrum measurement and management. In what follows, we describe the Spectrum Observatory's components.

2.1 Local spectrum measurements

Each measurement location consists of one or more spectrum analyzers, antenna(s) and a PC. The spectrum analyzers are connected to a single or multiple antennas for each frequency segment, and the PC is connected to each of the spectrum analyzers, as shown in Fig. 2.

Heterogeneity: Since each site is operated and maintained by third-parties, e.g. government organizations or universities, a key challenge is to support different measurement equipment: from the expensive CRFS RFEyesto the commonplace USRPs, to low-end RF Explorers. Each type of equipment supports different communication APIs, sampling and sweep rates, noise floors, and frequency ranges.

We support this heterogeneity by providing (i) software plug-ins that run on the PC to communicate with different spectrum sensors, (ii) a common XML configuration file that is read and translated for each spectrum sensor and (iii) a common file format for the output. We currently support plug-ins for the RFEye, and USRP, and are working

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2716936, IEEE Transactions on Mobile Computing

with other spectrum sensor vendors. The configuration file allows the measurement station administrator to set up the frequency range to sweep for each spectrum analyzer, the bandwidth window to use, the sample rate, sweep algorithm, and the number of samples per window. A process runs on the PC to query each of the attached spectrum analyzers with settings that are configured in the XML file and gets the raw IQ data from each of them. After the raw IQ data is loaded on the PC, the next step extracts feature vectors and stores them in the common file format. This format captures the hardware configuration as well as specific feature vectors extracted from the raw IQ data. Currently the only types of feature vectors that the file contains are the maximum, minimum, and average power at every frequency measured over a configured period of time. However, the file format is extensible, and additional feature vectors could be added in the future. For example, one can perform transmitter tracking and modulation recognition. We note that such advanced feature detection has privacy and security implications that need to be addressed but are beyond the scope of this work.

Upload bandwidth: Another challenge is the bandwidth required to upload the data in the cloud. If every IQ sample from each spectrum analyzer over the entire scanned spectrum is uploaded, we would need Gbps links at every site. The bandwidth requirement at the cloud servers would also be prohibitive. In contrast, if we only upload the power values (as opposed to IQ) at every measurement instant, we would need 100s of Mbps per site.

To illustrate this point, today the system is set up to support both USRP and CRFS RFEye sensors. Our USRP configuration uses a single PC attached to two USRP N200 RF sensors with a WBX and a SBX daughterboard. The WBX daughterboard scans the 50MHz-2200MHz range with a 25MHz window and 1024 samples per window, while the SBX daughterboard scans 2200MHz-4400MHz with a 25MHz window and 1024 samples per window. The feature vectors are aggregated over one minute with sixty of these in a single file. With this setup a single file could end up being approximately 61MB per hour without compression and about 50MB with compression. This file size can be drastically altered by changing any one or all of these settings. If for example, the measurement station is mobile and wants to write a file with the current GPS location every 5 seconds, only wants to scan 200MHz-1200MHz, but also wants higher resolution of 2048 samples per window, then the amount of data per hour generated would be 338MB uncompressed. To allow for different use cases and bandwidth limitations, station administrators are able to configure the amount of time over which the feature vectors are extracted as well as the number of these entries that are written to a single file to be uploaded.

2.2 Cloud storage and analysis

After the data has been written to the common file format on the local PC, a second process takes over. This process is responsible for long term storage of data in the cloud and managing data on the local machine. We discuss some challenges with cloud storage in this section.

Size of data and cost: The amount of cloud storage needed per measurement site is significant. For example, one of our deployments generates approximately 60MB/hour. If we upload this data and store it for one year, it will generate more than half TB. If we scale this up to 100, 1000, or 10000 measurement stations, the amount of data that needs to be stored on a yearly basis becomes very large. To solve this problem, the data is uploaded to blob storage instead of SQL tables. The latter reduces the metadata overhead and allows better flexibility in file size management. In our implementation we use Azure, but most of the design decisions are valid for AWS, which has mostly similar pricing and transactional limitations. By leveraging a cloud service, e.g. Azure, instead of deploying our own physical servers, we do not have to deal with other requirements such as uptime, stable networking and other basic infrastructure requirements.

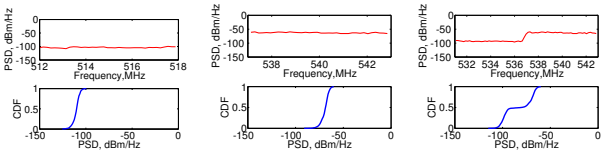
Data processing and analysis: Once the data is stored in the cloud, it needs to be acted upon. For every successful upload of a raw data file, a message is sent to a system running in the cloud to process the data to do further feature vector extraction and aggregation. This system needs to be scalable since we hope to eventually have thousands of measurement stations uploading raw data files every hour. Additionally, the processing of a single data file can take up to 15 minutes due to the amount of data being processed. With a single station today we upload over over 32,071,680 data points/hour. With 4350MHz of spectrum being scanned and 1024 data points per 25MHz window, we have 534,528 data points per minute. Each new measurement station, adds a file with a similar number of data points to be processed. To accomplish this, the process running in the cloud is run as a worker role in Azure. By using a worker role and a queueing system, we are able to take advantage of the cloud infrastructure and automatically scale the number of virtual machines that are instantiated. The latter is based on the length of the queue that maintains the list of the raw data files waiting to be processed.

Another problem is the number of transactions and the cost of those transactions. The raw data is already available in the cloud in a blob, but to enable quicker access to relevant data we store some of the processed data in table formats. Since we are limited to 20,000 transactions per second in each of our storage accounts, and we have over 32M data points/hour/measurement station, we need to compromise. We solve this issue by keeping the raw data that is uploaded to the cloud as compressed files stored as blobs. We only write aggregated data or new feature vectors out to the desired table structures. The data itself is aggregated hourly, daily, weekly, and monthly, and another table stores a pointer to the raw data file.

3 TxMiner design

Spectrum data collection and storage bring us half-way through creating a spectrum inventory. In order to answer the various questions on spectrum occupancy [1], we need to complement spectrum measurements with robust spectrum analytics. To this end, we design *TxMiner*, the first method for unsupervised detection of arbitrary transmitters. Traditionally, spectrum occupancy is analyzed manually by the use of tools, such as spectrograms of power spectral density. While such tools are informative, they are not very actionable. Particularly, they do not allow automated,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2716936, IEEE Transactions on Mobile Computing



(a) Idle (b) Occupied (c) Idle + occupied

Fig. 3. Probability Distributions of Power Spectral Density for different occupancy scenarios. The figure demonstrates how differences in measured signal distributions can inform transmitter characterization.

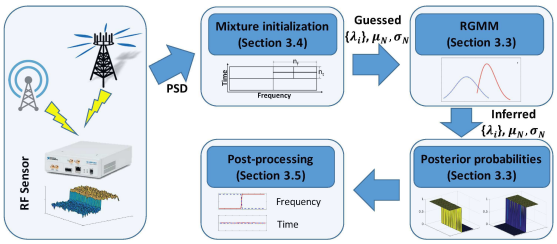


Fig. 4. TxMiner workflow.

fine-grained, long-term observation of spectrum occupancy patterns that are needed to inform DSA system design and policy. TxMiner tackles these problems by identifying transmitters in raw Spectrum Observatory data without prior knowledge of transmitter characteristics. Thus, it enables several new applications including transmitter-level mapping of spectrum occupancy, identifying rogue transmitters, DSA beyond TV white spaces and spectrum management.

Applications of TxMiner. The problem of spectrum mapping and management is relevant worldwide. In the US, there are joint initiatives involving the government, academia and industry [1], to create a platform for spectrum measurement and characterization. In developing countries, spectrum regulators often do not know how spectrum is being used[2]. TxMiner can be applied in both scenarios for advanced *mapping of spectrum occupancy*, which in turn enables effective spectrum use and regulation. Furthermore, it can inform *spectrum management* by answering questions such as (i) how many types of transmitters are using the channel? (ii) how many transmitters of each type are present? and (iii) what is the noise floor of the channel when these transmissions are not present? TxMiner can also be useful in *identifying rogue transmitters* by detecting discrepancies between expected and detected transmitters in a given band. This capability enables spectrum licensees and regulators to identify and remove spectrum squatters.

Beyond analysis of spectrum use, TxMiner can be applied in *support of DSA technologies*. The concept of DSA is often applied in the TV bands, where incumbents have stationary transmission patterns. Frequency ranges beyond TV bands provide vast opportunity for DSA access, however, the dynamic nature of transmitters in non-TV bands poses challenges for the operation of secondary devices. TxMiner can help by providing historical information of spectrum occupancy, which can inform DSA users about the transmission opportunity in various spectrum bands.

3.1 Key insights

The key insight behind TxMiner is that the *probability distributions of measured Power Spectral Density (PSD) reveal a lot about channel occupancy*. As an illustration, we study the probability distributions of three scans of the TV bands. Note that these observations are valid in other bands as well. Fig. 3 presents the probability distributions for the studied spectrum occupancy scenarios. The top graphs present a max-hold of PSD over a time window of 100 seconds, while the bottom graphs present the CDF of all values measured in this window over frequency and time. We see that the distributions of one occupied and one idle TV channel (Fig. 3(a) and 3(b)) are very similar in shape, however, the mean of the occupied channel is higher than that of the idle channel. In a frequency band, which is in part occupied and in part idle (Fig. 3(c)), the probability distribution we observe is bimodal, reflecting on the two spectrum activities. The means of the two modes correspond to the mean received power levels during the spectrum measurements.

In an urban or indoor environment, which are the prevalent settings where wireless communications take place, the transmitter's radio signal will attenuate with distance and encounter multiple objects in the environment that produce additional reflected, diffracted or scattered copies of the signal known as multipath signal components. Thus, the amplitude of the received signal can be characterized by a Rayleigh distribution while the phase can be characterized by a uniform distribution if we assume narrowband fading (i.e. different multipath components are not resolvable) [5]. In mathematical notation, the amplitude of the received signal $s(t)$ can be characterized by Rayleigh distribution as $R(s;\mu) = \frac{\pi s}{2\mu^2} \exp -\frac{\pi s^2}{4\mu^2}$, where μ is the mean of Rayleigh distribution and $4\frac{\mu^2}{\pi}$ is the average received power of the signal based on the attenuation resulting from path-loss and shadowing. Along with active transmitters, a spectrum scan might also capture noise, which can be modeled as white noise, and thus, follows a Gaussian distribution [5].

So far we observed that measured transmission signals follow a Rayleigh distribution, while measured noise follows a Gaussian distribution. Thus, power values from spectrum measurements can be modeled as a mixture of Rayleigh distributions, one for each measured transmitter, and a Gaussian representing the noise. Following this intuition, we develop a machine learning algorithm that models spectrum measurements as a mixture of Rayleighs and a Gaussian distribution. We dub this method *RGMM* (for Rayleigh-Gaussian Mixture Model). In what follows, we first outline the challenges of such transmitter characterization approach. We then describe how we address these challenges and present our RGMM algorithm in details.

3.2 Challenges

The challenges of unsupervised learning of transmitters include (i) mixture extraction, (ii) mixture initialization and (iii) post-processing to mine for transmitters. TxMiner addresses all these challenges as illustrated in Fig. 4. Its workflow takes as an input a matrix of power spectral density (PSD) over frequency and time. The workflow begins by determining the mixture initialization. It then fits a Rayleigh-Gaussian Mixture Model over the raw data and finally runs post-processing to characterize transmitters.

Mixture extraction. The goal of our analysis is, given a spectrum scan over time and frequency, to identify the

2. The authors have been approached by representatives of the Kenyan, Moroccan and Philippines government asking for help with analysis of spectrum occupancy.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2716936, IEEE Transactions on Mobile Computing

number and characteristics of transmitters that occupy the measured spectrum. We assume no prior knowledge for our spectrum data, thus this problem requires an *unsupervised machine learning technique*. As already established in §3.1, a population of radio signals can be represented as a mixture of Rayleigh and Gaussian distributions, however, there does not exist an off-the-shelf machine learning technique to fit such a mixture over unlabeled data. Thus, we develop a custom machine learning algorithm dubbed Rayleigh-Gaussian Mixture Model (RGMM) that fits a mixture of multiple Rayleigh and one Gaussian distributions over unlabeled data. We present RGMM in detail in §3.3.

Mixture initialization. While RGMM successfully models the power distribution of raw spectrum scans, obtaining a robust fit in a large time-frequency scan is a challenge. RGMM uses unsupervised machine learning and therefore requires a good initialization approach to extract a representative mixture model. To this end, we need a rough estimation of the signal distributions in a raw spectrum scan before running RGMM. There is a plethora of off-the-shelf data clustering algorithms that can be helpful in this step. TxMiner makes use of Gaussian Mixture Models for mixture initialization. We develop two mixture initialization techniques that are described and compared in §3.4.

Post-processing. Obtaining a robust mixture model that represents our raw data can help answer questions such as how many transmitters do we observe and what are their approximate power levels. This mixture model, however, hides time-frequency properties of the signal that answer more challenging questions such as what is the transmitter bandwidth and what are its temporal characteristics. In order to answer these questions we need a post-processing procedure that brings together the extracted mixture model and the time-frequency characteristics of the measured spectrum scan. We design a post-processing technique that (i) calculates the association probability of each measured power value with each of the distributions in the mixture model and (ii) smooths these associations to facilitate time-frequency analysis of the raw spectrum traces. We detail our post-processing algorithm in §3.5.

3.3 Rayleigh-Gaussian Mixture Models

The key feature of TxMiner that enables transmitter analysis is its ability to represent raw spectrum measurements as a mixture of Rayleigh and Gaussian distributions. This is enabled by our custom machine learning technique called Rayleigh-Gaussian Mixture Model (RGMM) that represents raw spectrum measurements as a mixture of several Rayleigh distributions – one for each sensed transmitter, and a Gaussian for the noise. We use this approach to identify sub-populations in the raw data that correspond to individual transmissions. A mixture model is a representation of a probability distribution as a weighted sum of individual probability distributions (densities). In our case, these individual densities correspond to k Rayleigh and one Gaussian densities. Each Rayleigh component in the mixture model is characterized via its mean, and is associated with a weight that captures its contribution to the mixture. The Gaussian density has three parameters: mean, variance and weight. Formally, the RGMM $p_{MM}(s)$ can be represented as

$$p_{MM}(s) = \sum_{i=1}^{k} w_i \cdot R(s;\mu_i) + w_{\mathrm{n}} \cdot N(s;\mu_{\mathrm{n}},\sigma_{\mathrm{n}}^2) \quad (1)$$

Here, $R(s;\mu)$ denotes the Rayleigh density with mean μ. Similarly, $N(s;\mu_{\mathrm{n}},\sigma_{\mathrm{n}}^2)$ is the Gaussian distribution with mean μ_{n} and variance σ_{n}^2. The weights $(w_1,..,w_{\mathrm{n}})$, means $(\mu_1,..,\mu_{\mathrm{n}})$ and the variance σ_{n}^2 comprise the parameters of the mixture model, which are discovered via the Expectation-Maximization (EM) algorithm. EM aims to discover the parameters that maximize the likelihood of the statistical model (i.e. the mixture) to represent the raw data. Formally, EM is an iterative procedure that starts with a random initial assignment of the parameters and keeps refining them by alternating between the E and the M step. The E and the M step for our application are defined as:

$$\text{E-Step:} p(s \in j) = \frac{R(s;\mu_j)}{\sum_{i=1}^{k} R(s;\mu_i) + N(s;\mu_{\text{noise}},\sigma^2_{\text{noise}})}$$

$$p(s \in \text{noise}) = \frac{N(s;\mu_{\text{noise}},\sigma^2_{\text{noise}})}{\sum_{i=1}^{k} R(s;\mu_i) + N(s;\mu_{\text{noise}},\sigma^2_{\text{noise}})}$$

$$\text{M-Step:} \mu_j = \frac{\sum_s s \cdot p(s \in j)}{\sum_s p(s \in j)}$$

$$\sigma_j^2 = \frac{\sum_s (s-\mu_j)^2 \cdot p(s \in j)}{\sum_s p(s \in j)}$$

$$w_j = \frac{\sum_s p(s \in j)}{N},$$

where $s \in j$ refers to s belongs to signal component j. The EM steps are repeated until convergence (change in parameters is less than a threshold). Each EM step increases the log likelihood of the data. Further the log-likelihood of the data is upper-bounded, as both the Gaussian and the Rayleigh distribution are bounded. These two conditions guarantee [18] that the EM procedure will always converge, at least to a local minimum. Analysis of the convergence rate of the EM algorithm is an active area of research [12] and out of the scope of our work.

Once we have learned the model that best represents the raw spectrum data we can calculate the likelihood of each original data sample to be generated by each of the components in our learned mixture model. We call these likelihoods *association probabilities* and note that they are essential in our post-processing analysis of transmitter characteristics (§3.5). We now explain our approach to calculating these association probabilities. Let S be the matrix of raw spectrum measurements over time and frequency. Each element of the matrix is s_{tf}, where t is the the row of the matrix (representing a time sample) and f is the column (representing a frequency sample). The association probability with each Rayleigh component R_i can be calculated using the probability density function (PDF) of a Rayleigh distribution as $R_i(s_{tf},\mu_i) = \frac{\pi s_{tf}}{2\mu_i^2} exp - \frac{\pi s_{tf}^2}{4\mu_i^2}$, where μ_i is the mean of the i-th Rayleigh distribution. Similarly, the association probability with the Gaussian component N can be calculated using the PDF of a Gaussian distribution $N(s_{tf},\mu_N,\sigma_G^2) = \frac{1}{\sigma\sqrt{2\pi}} exp \frac{(s_{tf}-\mu_N)^2}{2\sigma^2}$. We use the so-calculated association probabilities in our post-processing.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2716936, IEEE Transactions on Mobile Computing

Algorithm 1: MultiScale

```
1  Input:S data, (n_f, n_t) # partitions, l level, l_max - max level
2  Output:({λ_i}, μ_n, σ_n) Rayleigh transmitters and noise params
3  if l = l_max then
4      ({λ_i}, μ_n, σ_n) ←Rayleigh-Fit(D)
5      return ({λ_i}, μ_n, σ_n)
6  if l < l_max then
7      Λ ← ∅
8      Partition S into n_f × n_t regions S_{f,t}
9      for ∀S_{f,t} do
10         ({λ_i}, μ_n, σ_n) ← MultiScale(S_{f,t}, (n_f, n_t), l + 1, l_max)
11         Λ ← Λ ⋃ {λ_i}
12     {λ_i} ← Cluster(Λ, π)
13     if l < l_max then
14         return ({λ_i}, 0, 0)
15     else if l = 0 then
16         ({λ_i}, μ_n, σ_n) ←Rayleigh-Fit(S, {λ_i})
17         return ({λ_i}, μ_n, σ_n)
```

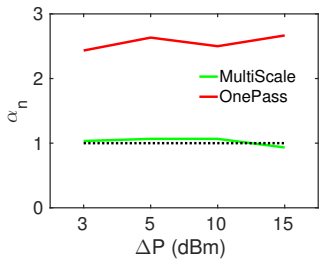


Fig. 5. Accuracy of OnePass vs. Multiscale. MultiScale is accurate even with low transmitter separation, while OnePass persistently overfits, even when the transmitters are 10 or 15dBm apart.

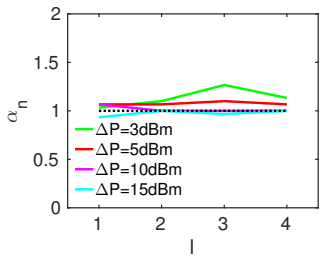


Fig. 6. MultiScale with increasing levels and increasing transmitter separation. MultiScale is highly-accurate as we increase l, even when the transmitter separation is as low as 3dBm.

3.4 Mixture initialization

Unsupervised machine learning methods such as RGMM enable us to analyze transmitter characteristics without prior knowledge of signatures. Obtaining a robust mixture model to represent raw spectrum measurements, however, is not trivial. *The robustness of the Rayleigh-Gaussian Mixture Model depends on the initialization of our RGMM algorithm.* To initialize RGMM we need a rough estimation of the mean values in measured signal distributions. Since our initialization goal is to simply estimate the means as opposed to fitting a specific distribution, any robust clustering technique for 1-D data is appropriate. To this end we use a generic Gaussian Mixture Models (GMM) fit to estimate the means in the raw data. The output of GMM clustering is a set of normal distributions characterized with a mean, standard deviation and mixing weights. We use the means of these distributions to initialize RGMM.

We propose two initialization techniques, both of which are based on GMM. The first initialization technique takes all the raw data of interest as an input, runs GMM and uses the means of the fitted distributions to initialize our RGMM algorithm. We dub this initialization method *OnePass*. The key benefit of this initialization approach is fast calculation of the seed values for RGMM. The drawback, however, is that if we consider a spectrum scan that features multiple transmitters, some of these transmitters might either be omitted or more components than the existing transmitters might be discovered. The reason for such deviations is that it is harder to model data with a large number of generating processes (i.e. transmitters).

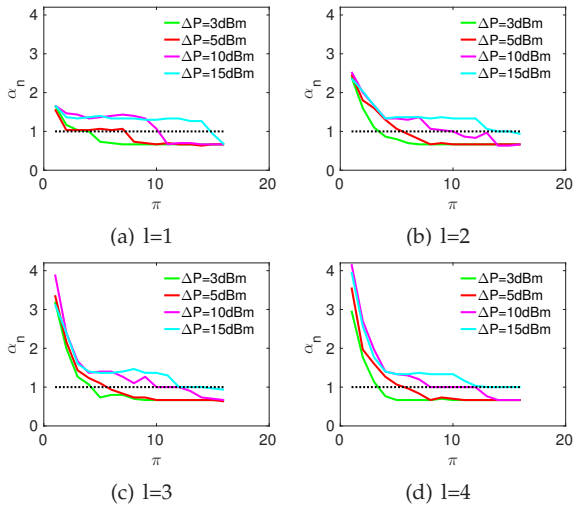
To reduce the number of generating processes and achieve robust initialization we design a second initialization approach, *MultiScale*, that calculates the initialization in a divide-and-conquer, bottom-up fashion. MultiScale divides the raw data in sub-spaces with increasing resolution. At the highest resolution MultiScale runs GMM in each subspace to find the representative distributions. It then groups the discovered distributions in decreasing resolution until it produces a single set of initialization values.

Our multi-resolution scheme MultiScale is presented in Alg. 1. The input to our function consists of the power measurement data S, the number of partitions in which the domain is to be recursively split in time n_t and frequency n_f, the current resolution level of l originally set to 0 and the maximum level l_{max}. The maximum level parameter l_{max} controls the maximum resolution at which we will obtain the initial fit. The output of MultiScale is the set of Rayleigh parameters $\{\lambda_i\}$ and the mean μ_n and standard deviation σ_n used to initialize RGMM.

The base case of the recursion ($l = l_{max}$) corresponds to the highest resolution of the frequency-time space (Lines 3-5). We perform a Rayleigh mixture fit with our default initialization based on GMM (Line 4) and return the obtained model parameters. The internal resolution levels are described in Lines 6-17. We initialize a set of Rayleigh parameters from the higher levels Λ with an empty set (Line 7) and partition the current time-frequency space S into $n_f \times n_t$ regions $S_{f,t}$ uniformly in each of the two dimensions (Line 8). Next, recursively invoke MultiScale for each of the subspace regions while incrementing the current level in the invocations and add the parameters of obtained Rayleigh components to Λ (Lines 9-11). We cluster the set of all Rayleigh parameters from the higher resolution using a threshold-based approach that groups all components that are less than π dBm apart (Line 12). Finally, if we are at an internal level (i.e. non-zero level), we return the clustered set of Rayleigh parameters (Lines 13-14), while at level 0 we perform one fit over the whole data initiating with the aggregated parameters from higher resolutions and return the final fit including the noise component (Lines 15-17). The final fit with initialization over all available data (Line 16) ensures that transmitters that have been separated due to the uniform partitioning of the space are fit based on all their data. Informally, this last step"readjusts" learned parameters within the whole data.

The time complexity $T(t, f)$ of MultiScale for data of size $t \times f$ and maximum level $l_{max} = L$, where at each level the time-frequency block is divided in four sub-blocks, can be expressed as $T(t, f) = 4^L * T(t/2^L, f/2^L) + L$. Here, the first term captures the time necessary to perform GMM at the highest resolution, while the second term captures the constant time necessary to group the means at each level (Alg. 1, Line 12). Since GMM uses the EM algorithm, the time complexity of MultiScale can further be expressed as $T(t, f) = 4^L * T_{EM}(t * f/2^{2L}) + L$. As we will shortly discuss, MultiScale performs robustly in various settings even for small l_{max} (Fig. 6). Thus, our method utilizes small values of l_{max} (up to 4), in which case, the complexity of MultiScale is similar to that of GMM [12].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2716936, IEEE Transactions on Mobile Computing



(a) l=1 (b) l=2 (c) l=3 (d) l=4

Fig. 7. Accuracy of MultiScale with level l and clustering threshold π. MultiScale is accurate for a wide window of π values, indicating that exact selection of π is not required.

We demonstrate the benefits of MultiScale on synthetic data, generated by a mix of Rayleigh and a Gaussian distributions and resembles real spectrum scans. We ensure our generative models are realistic by informing their parameters (i.e. mean and deviation) from real scans collected by USRP-based sensors. We note that synthetic data evaluation is necessary to allow tight control over experiment parameters including number of transmitters and transmitter separation. Our evaluation metric is *initialization accuracy* α_n, defined as the ratio of detected vs. expected transmitters.

We begin with a comparative evaluation of MultiScale and OnePass in Fig. 5. The figure plots average over 10 runs of α_n with increasing transmitter separation ΔP. For each run, MultiScale's parameters l and π are set to 1 and ΔP, respectively. We see that MultiScale is highly-accurate even with low transmitter separation, while OnePass persistently finds more transmitters than expected (i.e. overfits), even when the transmitters are 10 or 15dBm apart.

MultiScale requires the specification of two parameters: the maximum resolution l and the clustering threshold π. Naturally, one might ask what is an appropriate maximum resolution and what is a good approach to set the clustering threshold π, given that we will not have prior knowledge of transmitter separation in real spectrum characterization. Our following evaluation answers these questions. We begin by evaluating the impact of level selection l on initialization accuracy α_n in Fig. 6. The figure plots the average α_n over 10 runs as we increase l from 1 to 4 and the transmitter separation ΔP from 3 to 15. This result allows two important observations: (i) MultiScale is accurate even when $l = 1$, illuminating the unique benefits of component clustering (Alg. 1, Line 12) and (ii) MultiScale is persistently-accurate as l increases, indicating that the accuracy of our initialization does not hinge on exact selection of l. Further, we study the effects of cluster threshold on initialization accuracy in Fig. 7. For this experiment, we vary l from 1 to 4 (presented in Fig. 7(a)-7(d)) and π from 1 to 16. We evaluate α_n for four different transmitter separation values ΔP from 3 to 15dBm. We observe that as π increases to 3, the initialization rapidly becomes accurate for all values of l. The highest accuracy is achieved when π approaches the transmitter separation value. We also note that MultiScale is accurate for a fairly-large window of π values, indicating that exact selection of π is not required.

Our evaluation of MultiScale shows that it is able to robustly compute an initialization for RGMM. We also note that initialization does not need to be obtained every time TxMiner is ran, which will save computational resources. Rather, we can use the same initialization until RGMM obtains models with which the raw data values are poorly associated. Such poor association will be an indicator that a new initialization should be computed.

3.5 Post-processing

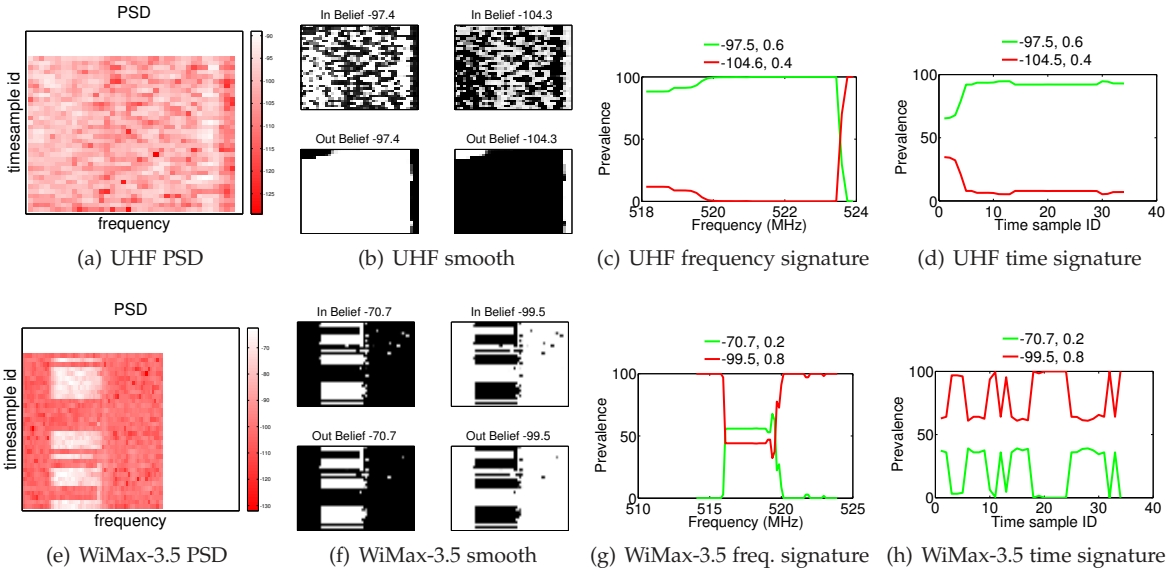
While RGMM allows mining of the number of transmitters and their sensed power levels, it does not allow for time-frequency analysis of the collected data. Such time-frequency analysis enables characterization of other important transmitter properties such as bandwidth and temporal behavior. In order to mine time-frequency properties we implement a post-processing procedure that uses the calculated association probabilities (§3.3).

The association probabilities provide intuition about the time-frequency properties of sensed transmitters, however, the inherently noisy nature of spectrum scans makes it hard to mine transmitter characteristics directly from the association probability matrices. Towards this end we make the following observation. Since transmitters occupy adjacent time and frequency samples, transmitter scans are coherent in the time-frequency domain. That is, adjacent values that are of similar magnitude are likely to be due to the same transmission. This observation allows us to apply spatial regularization to smooth the association probabilities and reduce the noisiness of the post-processed signal. For the purpose of spatial regularization we use a machine learning technique called Belief Propagation.

In the remainder of this section we detail our spatial regularization approach and describe how we use the regularized data to extract transmitter characteristics. Along with our methodology, in Fig. 8 we present an illustrative example of mining transmitter characteristics in two transmitter scenarios: a TV broadcast and a WiMax TDMA. Our RGMM method has fitted two components in each transmission: one representing the power of the transmitter and one capturing the noise. We detail each of these figures as we describe our post-processing technique.

Data regularization using Belief Propagation. The inherently noisy nature of RF signals causes our association probability matrices to suffer from salt-pepper noise. We can see the effect of salt-pepper noise in our example on Fig. 8. The first column from left to right represents the original PSD data over frequency and time. The more white the color is, the higher the measured power. The second column presents results before and after the regularization. The "In Belief" plots are the association probabilities before smoothing, while the "Out Belief" plots are the resulting smoothed association probabilities. Darker colors represent lower values. We see that the "In Belief" suffers from salt-pepper noise, whereby neighboring cells differ in their values. The latter makes it hard to determine if adjacent values belong to the same transmitter, which in turn makes it hard to detect transmitter characteristics.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2716936, IEEE Transactions on Mobile Computing



(a) UHF PSD (b) UHF smooth (c) UHF frequency signature (d) UHF time signature

(e) WiMax-3.5 PSD (f) WiMax-3.5 smooth (g) WiMax-3.5 freq. signature (h) WiMax-3.5 time signature

Fig. 8. Illustrative example of TxMiner post-processing, emphasizing the importance of belief propagation in noisy ("salt-and-pepper") signals.

We propose to alleviate this problem via spatial regularization using a machine learning technique popular in the image segmentation literature [2]. In particular, we formulate an energy minimization problem where we consider adjacent cells in the PSD matrix (both in frequency and time) as neighbors. The goal is to determine a solution that aligns with the mixture model available from the previous step and is spatially smooth. Formally, let us use $x_i \in \{1, ..k, \text{noise}\}$ to denote the index of the mixture distribution, with which the data s_i is associated. Then we consider the following form of the energy:

$$E(\mathbf{X}) = \sum_i -\log p_{MM}(s_i \in x_i) + \sum_{ij} V(x_i, x_j, s_i, s_j). \quad (2)$$

Here, $p_{MM}(s_i \in x_i)$ is a unary term that depends upon the output association probabilities from the mixture model. Intuitively, this term favors assignments that are obtained from the inference when fitting the model. The second term considers all pairs of neighbors (i and j), and smooths the data by using a function $V(\cdot)$ that depends upon the corresponding observations s_i and s_j in the PSD matrix $\mathbf{S}$:

$$V(x_i, x_j, s_i, s_j) = \begin{cases} -\log e^{-\beta|s_i - s_j|} & \text{if } x_i = x_j \\ -\log[1 - e^{-\beta|s_i - s_j|}] & \text{Otherwise} \end{cases}$$

Note that the pairwise term favors similar assignments to s_i and s_j only when the values x_i and x_j are similar. Intuitively, the pairwise term will favor dissimilar assignments to adjacent cells only when there is a large difference in observations in the PSD matrix.

An assignment that minimizes the above energy provides a solution that is coherent in time and frequency and aligned with the solution provided from the mixture model procedure. However, determining the minimum energy assignment for such energies has been determined to be NP-complete. Reasonable approximation can be computed via message passing schemes such as loopy Belief Propagation [20]. In this paper we specifically, use the sum-product version of loopy belief propagation, where given the mixture model inferences, we formulate the energy and obtain a solution via loopy message passing until convergence.

The "Out Belief" plots in Fig. 8(b) and 8(f) show the result after running the loopy Belief Propagation. The resulting signal is more regularized in the time-frequency domain and does not suffer from salt-pepper noise.

TABLE 1
Rules for determining transmitter type.

Type	Rule
Broadcast	$\sigma_T < THR_T$ and $\sigma_F < THR_F$
TDMA	$\sigma_T > THR_T$ and $\sigma_F < THR_F$
FDMA	$\sigma_T < THR_T$ and $\sigma_F > THR_F$
Hopping	$\sigma_T > THR_T$ and $\sigma_F > THR_F$

Mining transmitter characteristics. The smoothed association probabilities obtained in the previous step enable efficient extraction of transmitter signatures in order to mine transmitter characteristics. In this analysis we determine key transmitter properties including: bandwidth, active time and type (including TDMA, FDMA, broadcast and frequency hopping). Towards this end we compact the association probabilities from the time-frequency domain in one-dimensional space in either frequency or time. We call these compacted probabilities *temporal* and *frequency* transmitter *signatures* and denote them as P^t and P^f. A temporal P_i^t and frequency P_i^f signature is calculated for each Rayleigh distribution i fitted onto the raw spectrum measurements. We calculate P_i^t and P_i^f as follows:

$$P_i^t = \frac{\sum_f^F R_i(s_{ft}, \mu_i)}{F} \text{ and } P_i^f = \frac{\sum_t^T R_i(s_{ft}, \mu_i)}{T} \quad (3)$$

Our illustrative example in Fig. 8 presents the time and frequency signatures of a TV broadcast and WiMax TDMA transmitter. Since a broadcast channel occupies all the time-frequency samples, we see that the signatures of such transmitters have low variance over time and frequency (Fig. 8(c) and 8(d)). In contrast, a TDMA transmitter such as WiMax occupies a fixed bandwidth, however, its active time is non-contiguous. This is reflected in its signatures (Fig. 8(g) and 8(h)): the frequency signature has low variability, whereas the time signature varies, capturing the intermittent presence of this transmitter over time.

We use these observations to design our detection of transmitter bandwidth, active time and type. Specifically, we calculate the transmitter bandwidth by determining the span of non-zero frequency signature. Similarly, we determine the transmitter active time by calculating the span

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2716936, IEEE Transactions on Mobile Computing

of non-zero time signature. Lastly, we use the variance of transmitter signature to determine the transmitter type. Let us denote the variance of the time signature by σ_T and the variance of the frequency signature by σ_F. We can then determine the transmitter type by following the rules in Table 1. THR_T and THR_F denote thresholds of time and frequency signature variance against which we decide the type of transmitter. Since the magnitude of the variance depends on the magnitude of the transmitter signature, we pick a percentage of the maximum signature value as our threshold. Of note is that the threshold can be adjusted if we had prior knowledge about the expected transmission. Since we assume no such knowledge, we use 20% on the maximum signature as a threshold in our evaluation.

4 TxMiner Evaluation

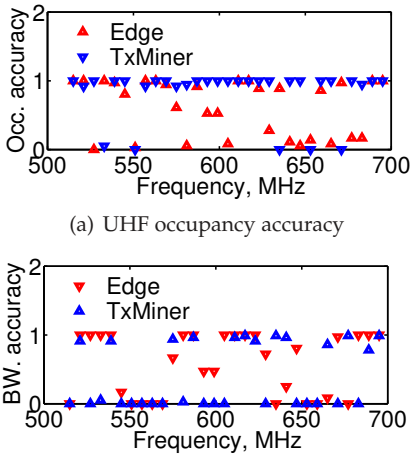
In this section we evaluate TxMiner on real-world scans collected by the Spectrum Observatory. First, we focus on accuracy of detecting active transmitters. We then evaluate TxMiner's ability to extract individual transmitter characteristics such as transmitter count, bandwidth and type. We compare TxMiner with a state-of-the-art algorithm for occupancy detection called edge detection [19]. Our evaluation shows that TxMiner outperforms edge detection in both controlled settings as well as in real world measurements. We show that TxMiner has high accuracy in detecting occupancy of individual transmitters and their bandwidths. Furthermore, TxMiner is capable of detecting transmitter count and bandwidth in multi-transmitter scenarios.

4.1 Implementation, measurement setup and data

We implement TxMiner in MATLAB, using our custom implementation of RGMM, MathWorks' implementation of GMM (`gmdistribution`) and an implementation of loopy Belief Propagation from [9]. We use scans collected by Spectrum Observatory sites equipped with the CRFS RfEye sensors. The scans were captured from 30MHz to 6GHz every 3 seconds with variable frequency resolution, depending on the band. The sensors are equipped with a multi-polarized receiver antenna that supports the entire band from 25MHz to 6GHz. We use the following datasets.

Ground truth. To establish our ground truth, we use TV-UHF spectrum scans (512-698MHz) collected by a stationary RfEye sensor scanning the spectrum every 3 seconds with a step of 160kHz. We establish the ground truth through a two-step process that combines spectrum measurements with information from several TVWS databases (FCC CDBS, AntennaWeb, TVFool, Spectrum Bridge and iConnectiv). In the first step we verify with the databases which channels are allocated to broadcasters at a given sensor location. In the second step we measure the spectrum to confirm whether the allocated channels are actively used or not. We found that five of all the allocated channels were in fact idle. All of the non-allocated channels were measured as idle. Following this two-step process we designate the active and idle bands and use the so-established ground truth to evaluate the accuracy of TxMiner detection.

Controlled. We utilize a few controlled transmissions to evaluate TxMiner's ability to detect custom transmitters. We record traces from three modes of wide-range outdoors WiMax transmission: one using 1.75MHz bandwidth, a second using 3.5MHz and a third transmitting at 7MHz. We also performed spectrum scans during on-campus wide-range outdoor trials of FCC-certified white spaces radios running proprietary DSA protocols. Both the WiMax and the trial traces were collected with a stationary RfEye scanning every 3 seconds with a frequency step of 160kHz.



(a) UHF occupancy accuracy

(b) UHF bandwidth accuracy

Fig. 9. (a) Occupancy and (b) bandwidth detection. TxMiner outperforms edge detection in both occupancy and bandwidth accuracy. Edge detection fails in nearly 50% of the cases to accurately detect an occupied channel. It often detects bandwidth where there is no active transmitter or does not detect anything where there is an active transmitter.

Artificially mixed. We generate artificially-mixed signals drawn from our TV-UHF ground truth. We intertwine over the same frequency band different transmissions or alternate transmission with idle period. By doing so we can emulate single- or multiple-transmitter TDMA schemes, which allows us to establish a ground truth set and quantitatively evaluate TxMiner's ability to detect multiple transmitters.

4.2 TxMiner performance

Occupancy detection. We begin our evaluation by analyzing occupancy detection. For this experiment we run TxMiner on our ground truth data in 6MHz steps and calculate the accuracy of occupancy detection. In each 6MHz bin there are F samples, depending on the scan configuration. For each of these samples we find if it is occupied or idle. Our accuracy metric then captures the fraction of correctly-detected samples divided by the total number of samples F. Intuitively, an accuracy of 1 corresponds to correct detection of an occupied TV channel, whereas and accuracy of 0 corresponds to a correct detection of an idle TV channel. An accuracy between 0 and 1 indicates a failed detection.

Fig. 9(a) presents our accuracy results for occupancy detection, where the blue markers correspond to TxMiner and the red ones represent Edge Detection. TxMiner has a detection accuracy of 0 or 1 and outperforms Edge Detection in nearly 50% of the cases. For example, channel 23 is idle but is surrounded by two low-power channels, thus edge detection fails to recognize it, while TxMiner detects it successfully. The reason for the poor performance of edge detection is that it often fails to recognize a rising or falling edge, which forces longer frequency spans to be incorrectly recognized as idle or occupied.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2716936, IEEE Transactions on Mobile Computing

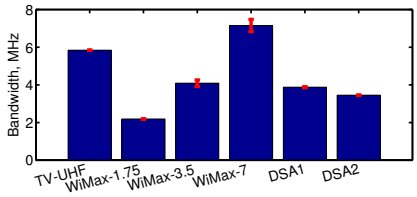


Fig. 10. Bandwidth detection of different transmitters. TxMiner is persistently able to accurately detect the bandwidth of various transmitters.

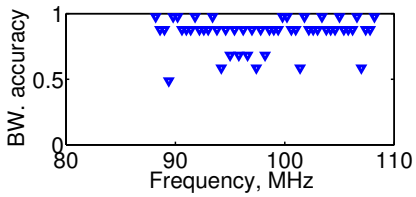


Fig. 11. Bandwidth detection in the radio FM band. TxMiner performs is highly-accurate in detecting narrow-band transmissions.

Bandwidth detection. We evaluate TxMiner's ability to detect transmitters' bandwidths. First, we run TxMiner on our TV-UHF ground truth in 6MHz steps. At each step we calculate the bandwidth of the detected transmitter. Fig. 9(b) presents a comparison between TxMiner and Edge Detection. The y-axis on the graph presents bandwidth accuracy, defined as the ratio between detected and expected bandwidth, where expected bandwidth is equal to 6MHz (TV channel width in the US). As we can see, TxMiner successfully detects the bandwidth of active transmissions and detects a bandwidth of 0MHz where we have measured no transmission or where there is no expected transmission. At the same time Edge Detection often fails to detect the bandwidth of active transmitters, or detects a 6MHz transmitter in idle channels. The reason for the poor performance of Edge Detection is that it often times fails to account for a rising or falling edge, which results in larger areas being detected as idle or occupied than there actually exist.

Next, we evaluate TxMiner's capability to persistently detect transmitter bandwidth. Particularly we look at the TV-UHF band, three TDMA WiMax transmissions with known bandwidths of 1.75MHz, 3.5MHz and 7MHz and two proprietary TDMA DSA transmissions with bandwidths of 4MHz and 3.5MHz. For TV-UHF we present average and standard deviation of detected bandwidth across all the channels we identify as occupied. For all the WiMax and DSA transmissions we present average and standard deviation across five distinct periods from the captured traces. All but the DSA2 scan periods are of 100s duration. For DSA2 we use a 300s scan duration because the TDMA nature of this transmission makes it so we cannot capture enough transmission samples within 100s. Fig. 10 presents our results. TxMiner is persistently able to detect the bandwidth of each transmitter type, as indicated by the small standard deviation bars. Furthermore, the detected bandwidths are very close to the expected bandwidths.

We also evaluate TxMiner's performance on narrow-band transmissions such as those in the FM-radio band. Fig. 11 presents our results for accuracy of bandwidth detection. In this experiment we ran TxMiner over the entire FM band from 88MHz to 108MHz in steps of 400kHz. The graph presents for each 400kHz chunk the bandwidth

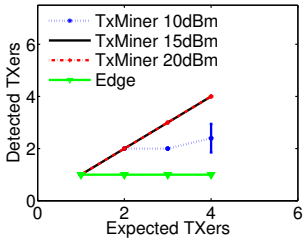


Fig. 12. Characterization with increasing number of transmitters. TxMiner detects the increasing number of transmitters and outperforms edge detection, which cannot identify more than one transmitter.

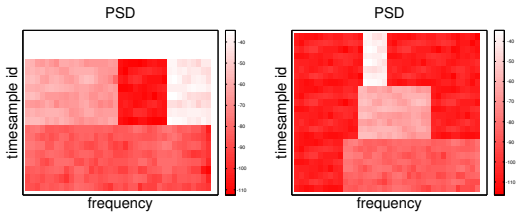
accuracy expressed as the ratio between detected bandwidth and step size. As we can see, majority of the detected channels have bandwidth accuracy of either 0.98 or 0.88, which corresponds to a bandwidth of 392kHz and 352kHz, respectively. The 392kHz bandwidths likely correspond to HD radio transmissions, which occupy wider bands. The 352kHz transmissions correspond to stations that were sensed with very strong signal, in which case we would see the squelch tones as a separate peak. Finally, we see chunks where bandwidth accuracy is lower. Those are likely to be radio transmissions that were sensed with low power, thus their bandwidth does not span the entire 400kHz band.

Transmitter type detection. As detailed in §3.5 we make use of the variance of the time and frequency signatures of a transmitter to determine its type. We now demonstrate TxMiner's ability to determine the transmitter type of our ground truth transmissions. We focus on a TV broadcast operating on channel 22 (518-524MHz). We use 20% of the maximum signature to determine the variance thresholds. For the TV broadcast $THR_F = 20$ and $THR_T = 9.66$. The calculated variance of this transmitter's signatures are 3.73 and 18.31 for time and frequency, respectively. Both the variances are lower than the respective thresholds and thus the transmitter is correctly identified as a broadcast.

Multiple transmitters. We evaluate TxMiner's performance with multiple active transmitters. To emulate such scenarios we artificially mix and amplify measured signals. Our first evaluation focuses on TxMiner's ability to detect an increasing number of transmitters of the same bandwidth. For this experiment we mix over time measured signals from the TV-UHF band and artificially amplify them by adding 10, 15 or 20dBm. We then run TxMiner and count the number of detected transmitters. Fig. 12 plots the number of detected transmitters as a function of the number of expected transmitters. We present three results for TxMiner averaged over five runs and compare TxMiner's performance with Edge Detection. As we can see, TxMiner outperforms edge detection. The reason for the poor performance of Edge Detection is that it only considers an average of the measured signal and unlike TxMiner, does not take into account the time-frequency properties of the signal. In contrast, TxMiner is capable of detecting the number of transmitters with high accuracy. The accuracy of TxMiner is lower with 10dBm margin, where the algorithm sometimes fails to differentiate between transmitters.

Next we evaluate TxMiner's ability to extract multiple transmitters with variable bandwidths. For this experiment too we use artificially mixed and amplified signals. We study two cases of spectrum occupancy presented in Fig. 13.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2716936, IEEE Transactions on Mobile Computing



(a) Case 1 (b) Case 2

Fig. 13. Multi-transmitter evaluation scenarios.

TABLE 2
Detection of multiple transmitters.

	TX 1		TX 2		TX 3	
	E.BW (MHz)	D.BW (MHz)	E.BW (MHz)	D.BW (MHz)	E.BW (MHz)	D.BW (MHz)
Case1	6	5.84	3	2.84	1.4	1.26
Case2	4.375	4.26	2.34	2.68	0.78	0.63

Each of these cases includes a different configuration of three transmitters. In case 1 we have a 25 second transmission with 6MHz bandwidth, followed by two concurrent transmissions, one 3MHz wide and one 1.4 MHz wide and separated by an idle zone. The second case features three consecutive transmissions each of 25 seconds. Table 2 presents for each case and each transmitter the expected and the detected bandwidth (E.BW and D.BW, respectively). TxMiner successfully detects all the expected transmissions and is also accurate in detecting their bandwidths.

4.3 Impact of scan duration

In this section we evaluate the impact of scan duration on the accuracy of occupancy detection. The presented results indicate how quickly can TxMiner begin detecting transmitters after a spectrum scan is initiated. To this end, we run TxMiner on all the channels in the TV UHF band while changing the number of time samples we consider. We start with a scan duration of 3 seconds, which in our setup corresponds to two sweeps, and double the scan duration up to 192 seconds (65 sweeps). Fig. 14 presents average and standard deviation of accuracy (as calculated in §4.2) over all the TV channels for each scan duration. Even for small scan durations the average accuracy is high which indicates that TxMiner can detect transmitters successfully even after two frequency sweeps. Notably, the deviation across channels for small scan times is high as well, which would not be desirable for stable performance across various scenarios. This deviation depends on how noisy the channel is: intuitively the more noisy the channel the more samples TxMiner needs in order to perform accurate transmitter detection. As the scan duration increases up to 96 seconds (33 sweeps), we see that the standard deviation becomes minimal, which indicates that Txminer can persistently achieve high accuracy in about 33 sweeps across different transmission scenarios.

5 Creating a Spectrum Inventory

We now put together the Spectrum Observatory's measurement capabilities and TxMiner's spectrum analytics to create a RF inventory that gives information about transmitter characteristics over frequency, time and space. Specifically, we utilize data collected by the Spectrum Observatory over a day at a single location and seek transmitter patterns. We present results from wide-band and long-term analysis of

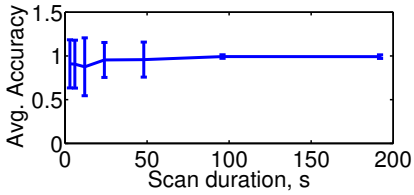
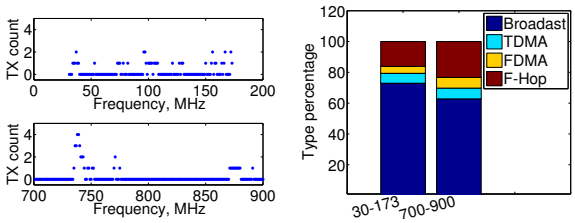


Fig. 14. Accuracy with increasing scan duration. TxMiner has high detection accuracy with scan durations as short as 3 seconds (2 sweeps). The stability of transmitter detection across different channels, regardless how noisy they are, is guaranteed at 96 seconds duration (33 sweeps).



(a) Number of transmitters (b) Transmitter type

Fig. 15. Number of transmitters (a) and transmitter type (b) detected over a wide frequency range. TxMiner detects multiple transmitters in a single 1MHz chunk in bands that are characterized with narrow-band transmissions. It also detects wide-band transmitters by extracting a single transmitter in each 1MHz chunk of a contiguous band. Lastly, TxMiner identifies transmitter type in bands occupied by a single transmitter.

spectrum occupancy using TxMiner. First, we map spectrum occupancy by analyzing the number of transmitters and their type over wide frequency band. We then propose a technique to detect rogue transmitters and utilize it to detect a rogue transmitter in the Spectrum Observatory traces. Finally, we make a case for TxMiner-based support of DSA systems through longitudinal analysis of the DSA opportunity in parts of the UHF band.

5.1 Mapping spectrum occupancy.

When mapping spectrum occupancy, it is important to look at occupancy states both over a wide frequency range as well as over long time. We now demonstrate TxMiner's capability to support such analysis.

Mapping number of transmitters. Our analysis of number of transmitters focuses on two frequency bands: 30-173MHz and 700-900MHz. We choose these bands to demonstrate TxMiner's ability to detect the number of transmitters in bands that are typically occupied by narrow-band transmitters such as 30-173MHz and parts of 700-900MHz and other characterized with wide-band transmitters such as portions of 700-900MHz band. Fig. 15(a) plots the number of transmitters detected in each 1MHz chunk. In ranges that are characterized with narrow-band transmissions TxMiner detects up to 4 transmitters in a single 1MHz chunk. In contrast, where wide-band transmitters are present, TxMiner detects contiguous 1MHz chunks as occupied by a single transmitter.

Transmitter type detection. Along with transmitter count we utilize TxMiner to detect transmitter type in a wide frequency band. Fig. 15(b) presents a bar-graph with detected transmitter types in 1MHz chunks occupied by a single transmitter. Each bar presents the percentage of transmitter types detected in the two frequency bands of interest. As we can see, majority of the transmitters in both bands are broadcast. We observe a higher percentage of TDMA, FDMA and frequency hopping transmitters in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2716936, IEEE Transactions on Mobile Computing

700-900MHz band in comparison with the 30-173MHz band. This can be explained with the nature of the incumbent transmitters in these bands: while 30-173MHz is characterized with narrow-band broadcasts such as FM radio, the 700-900MHz band hosts technologies such as public safety land mobile communications[3] that are non-broadcast.

5.2 Identifying rogue transmitters

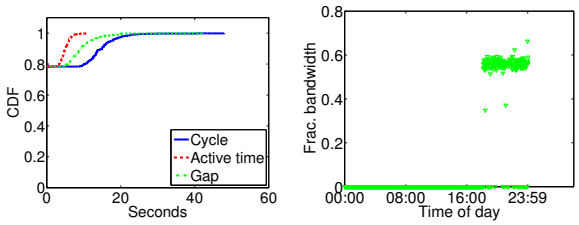
To illustrate TxMiner's capability to detect rogue transmitters we define *rogue coefficients* C_β and $C_\mathcal{T}$ that capture the likelihood that the transmitter sensed in a time-frequency chunk is rogue by analyzing the bandwidth β and active time $\mathcal{T}$ of the detected transmitter. Towards this end we require prior knowledge of the characteristics of the transmitter that is expected to operate in a given band. We note that such prior knowledge can be obtained by considering the previous transmitter characteristics discovered by TxMiner. Thus, our rogue coefficients captures the difference between the expected and the detected transmitter characteristics as $C_\beta = \beta_d/\beta_e$ and $C_\mathcal{T} = \mathcal{T}/\mathcal{T}_e$, where β_d and β_e are the detected and expected transmitter bandwidth, while $\mathcal{T}_d$ and $\mathcal{T}_e$ are the detected and expected active time. These coefficients vary between 0 and 1, where 1 indicates that the detected and expected transmitters are the same, 0 indicates that there is no transmitter, where a transmitter is expected and anything between 0 and 1 indicates that the detected and expected transmitters are different. Of note is that this method does not consider the time-frequency characteristics of the sensed transmitter; that is, if a rogue transmitter has the same active time pattern as an incumbent but transmits at different times it will not be recognized. We leave more robust detection of rogue transmitters as a future work.

We calculate the rogue coefficient for all TV bands and identify that one TV channel is occupied by a non-TV transmitter. The channel in question is channel 20 (506-512MHz), for which TxMiner calculates rogue coefficients $C_\beta = 0.61$ and $C_\mathcal{T} = 0.22$. While the expected transmitter here is a TV broadcast with 6MHz bandwidth and continuous active time, the detected transmitter exhibits different characteristics as captured by the rogue coefficients. A closer look at the occupant indicates that the transmitter has a bandwidth of 4MHz and transmits in a TDMA fashion.

5.3 Support for DSA systems.

Dynamic spectrum access is a concept most often applied in the context of TV White Spaces, where the primary transmitters have stationary behavior. Thus, these bands are well-suited for database-driven management. There are plethora of radio bands such as radar and satellite that are seldom used by their incumbents and provide a great opportunity for dynamic spectrum access. However, these bands pose challenges in operation of secondary users due to the highly-dynamic nature of incumbents. In order for secondary users to fully utilize the potential of these bands they need a mechanism to evaluate the transmission opportunity in both frequency and time by assessing not only if there is an incumbent but also how much bandwidth and time does it occupy and whether the temporal occupancy patterns are predictable or not. TxMiner can provide such information. To illustrate how, we analyze one proprietary DSA transmission that exhibits TDMA behavior.

3. http://www.ntia.doc.gov/files/ntia/publications/4b_5_11_0.pdf



(a) Active time (b) Bandwidth

Fig. 16. Characteristics of a single TDMA transmitter over 24 hours. TxMiner successfully identifies transmitter bandwidth and active time over a long period, and can thus inform DSA technologies about the transmission opportunity in a given frequency band.

Fig. 16 presents our analysis of a 6MHz band (506-512MHz) over 24 hours. TxMiner identifies a single transmitter in this band that is sensed at -94dBm (graph omitted due to space limitations). Furthermore, this transmitter is active for about 20% of the entire 24-hour period. We analyze the frequency and temporal characteristics of this transmitter in Fig. 16(a) and 16(b). In analyzing the temporal characteristics we consider three metrics: (i) the *active time* duration, (ii) the active time *cycle*, that is the time from the beginning of one active period to the beginning of the next active period, and (iii) the *gap* between consecutive active times, that is the time between the end of one active period and the beginning of the next. In Fig. 16(a) we plot a CDF of the average active time, cycle and gap in intervals of 100s over the 24-hour period. Since the transmitter is active only 20% of the time, 80% of the values are zero. Based on the values that correspond to transmitter activity we can see that the average duration has a median of 5 seconds and does not vary much over different 100 seconds snapshot. In contrast, the gap has a median of 9 seconds, which is larger than the active time, and varies significantly (from 5 to 42 seconds). Lastly, the cycle has a large variation (between 9 and 48 seconds). These temporal characteristics indicate that the observed transmission is a-periodic and the transmitter is inactive for a larger fraction of the time. Finally, we analyze occupied bandwidth. Fig. 16(b) plots the ratio of detected bandwidth vs. analyzed bandwidth (which is 6MHz in this analysis) in each 100s period. As we can see, the fraction of occupied bandwidth is persistently around 0.6, which indicates that 40% of the analyzed band is idle.

This analysis can inform a secondary DSA transmission as follows. Since the incumbent is only present 20% of the time, the secondary transmitter can use the entire band for transmission in 80% of the day. In periods where the incumbent is active, due to its a-periodic nature, it would be hard to predict opportunities for secondary transmission without real-time sensing. Depending on the sensing efficiency of the secondary transmitter, it can decide whether to opt for sensing and transmission based on the average gap duration supplied by TxMiner. Finally, 2MHz of the 6MHz analyzed band is persistently available, thus the secondary transmitter can decide to utilize this portion continuously without the need of complex sensing techniques if this would satisfy the application requirements.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2716936, IEEE Transactions on Mobile Computing

6 Related Work

Prior work on spectrum analysis can be classified into 3 categories: wide band spectrum occupancy analysis, envelope detection for identifying unknown signals, and detecting transmitters with known signatures.

Several studies have analysed large scale spectrum measurements to identify portions of spectrum that are not used [11, 17], or identify patterns of primary users such that unused spectrum can be opportunistically reused [3, 10]. This body of work assumes no knowledge about the transmitter. They typically apply a threshold for noise, and any signal above this threshold is assumed to be occupied, anything below is assumed to be free. [3] analyses spectrum from China, and models the arrival of users in the cellular bands. [10] analyses spectrum from 30 MHz to 6 GHz, and studies opportunities for dynamic spectrum access in these bands. [15] features unsupervised separation of spectrum utilization patterns to inform efficient and adaptive spectrum sensing. However, none of these works share the goal of TxMiner, and are unable to extract detailed transmitter characteristics from a wideband spectrum trace.

Another set of techniques, which is primarily used by practitioners, is to tease apart unknown transmissions from known transmitters. This is frequently used to identify interferers in the spectrum, for example, in the wireless carrier spectrum. The most common technique is that of envelope detection [6]. A circuit (or these days software) tries to fit a curve around the max-hold (or mean) of signals. Although this technique is useful in determining anomalies in the curves, it does not provide much insight into the distributions that make up the max-hold or mean.

Most closely related to our work are SpecNet [8], DoF [7], AirShark [14] and DECLOAK [13]. SpecNet is a system for large-scale spectrum measurements, which harnesses high-end spectrum analyzers contributed by SpecNet participants and provides basic functionality for SNR-driven occupancy detection. In contrast, our Spectrum Observatory makes use of lower-end spectrum sensors and incorporates TxMiner for advanced transmitter characterization. DoF builds cyclostationary signatures for different transmitters in 2.4 GHz, such as Wi-Fi, Bluetooth, etc., and mines spectrum data for these signatures to determine the users of the spectrum. AirShark tried to solve a similar problem, but using commodity Wi-Fi chipsets. DECLOAK focuses on OFDM transmissions only and uses a combination of cyclostationary features with Gaussian Mixture Models to extract transmitter characteristics. While all three techniques are useful, they only work when the transmitter patterns are known. TxMiner takes the next step, and identifies transmitters when their patterns are not known.

7 Discussion & Future Work

To summarize, this paper presents the Spectrum Observatory, a system that collects, analyzes and shares spectrum measurements from multiple locations. A key feature of the Spectrum Observatory is its ability to perform detailed, unsupervised transmitter characterization by the use of our novel machine learning method called TxMiner. We use TxMiner to create a spectrum map that features transmitter count and characteristics. We demonstrate detection of rogue transmitters and analysis of DSA opportunity in licensed bands. Although the knowledge gleaned by the Spectrum Observatory and TxMiner is very useful, it is still the first step. We believe that there is a need for more fundamental research contributions both in spectrum measurement infrastructures and in spectrum characterization. We list some of our research efforts in this direction below.

Spectrum measurement infrastructures: While stationary spectrum observation is intuitive first step towards spectrum characterization, there are several limitations of this approach, including (i) limited view of spectrum occupancy, (ii) inability to perform advanced detection (e.g. transmitter localization) and (iii) poor scalability in achieving ubiquitous spectrum sensing and characterization. A promising future direction we are currently exploring is that of a hybrid fixed and mobile (dedicated or crowdsourced) measurement infrastructure, whereby part of the spectrum scanners are fixed, while others are mobile. Such measurement infrastructure addresses the above limitations and introduces a plethora of research challenges related to storage, processing and characterization of spectrum data from heterogeneous sensors. Beyond sensing and characterization, a fully-fledged spectrum infrastructure should provide for a variety of services including policy, technology, research and enforcement. While the Spectrum Observatory caters to policy and research, it requires further advances to support real-time DSA technology and enforcement. This requires the design of APIs and flexible sensor configuration to cater to the different timeliness requirements of these applications.

Advanced spectrum occupancy characterization: While TxMiner is the first method to achieve unsupervised characterization of spectrum occupants, there remain multiple research problems to be further explored. One such is *transmitter collocation*. Since TxMiner looks at power profiles of transmitters, it is unable to distinguish two collocated transmitters with similar power profiles. In such cases the two transmitters together will be classified as a single transmitter. To this end we can incorporate prior knowledge of the occupants' characteristics to determine the number of active transmitters. Second, TxMiner does not *feature detection of mobile transmitters*. We note that the properties of signal distributions can be applied to this problem as well. Particularly, the signal distributions of mobile transmitters are different than those of static in that they change over time depending on the speed and direction of the transmitter with respect to the RF sensor. Using this observation, we are designing methods for identification of mobility and speed. Lastly, TxMiner *does not incorporate prior knowledge of transmitter characteristics.* Previous work [7, 14] has looked at identifying transmitters with known temporal signatures. TxMiner can leverage such approaches to eliminate known transmitters from spectrum scans and focus on unknown transmitters, thus improving detection time and accuracy.

Despite these limitations, the current implementation of the Spectrum Observatory and TxMiner revolutionizes spectrum mapping by allowing extraction of transmitter count and characteristics, detection of rogue transmitters and identification of opportunities for dynamic spectrum access. Our future research efforts will open doors toward efficient use and better understanding of spectrum bands.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2017.2716936, IEEE Transactions on Mobile Computing

8 Acknowledgements

This work was in part supported through a NSF CISE Research Initiation Initiative (CRII) grant CNS-1657476.

References

[1] NSF Workshop on Spectrum Measurement Infrastructures. http://www.cs.albany.edu/~mariya/nsf_smsmw/, April, 2016.
[2] A. Blake, P. Kohli, and C. Rother. Advances in Markov Random Fields for Vision and Image Processing. MIT Press, 2011.
[3] D. Chen, S. Yin, Q. Zhang, M. Liu, and S. Li. Mining spectrum usage data: A large-scale spectrum measurement study. MobiCom '09, Beijing, China, 2009.
[4] A. Chowdhery, R. Chandra, P. Garnett, and P. Mitchell. Characterizing Spectrum Goodness for Dynamic Spectrum Access. 50th Allerton Conference, Monticello, Illinois, October, 2012.
[5] A. Goldsmith. *Wireless communications*. Cambridge Univ Pr, 2005.
[6] J. Gorin. Detector selection for spectrum analyzer measurements, February 2003.
[7] S. S. Hong and S. R. Katti. DOF: A local wireless information plane. SIGCOMM, Toronto, Canada, 2011.
[8] A. Iyer, K. K. Chintalapudi, V. Navda, R. Ramjee, V. Padmanabhan, and C. Murthy. Specnet: Spectrum sensing sans frontières. NSDI, Boston, MA, 2011.
[9] J. Kim and K. Grauman. Observe locally, infer globally: A space-time mrf for detecting abnormal activities with incremental updates. In *IEEE CVPR*, June 2009.
[10] V. Kone, L. Yang, X. Yang, B. Y. Zhao, and H. Zheng. On the feasbility of effective opportunistic spectrum access. IMC, Melbourne, Australia, November, 2010.
[11] 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. In *TAPAS*, Aug. 2006.
[12] I. Naim and D. Gildea. Convergence of the EM Algorithm for Gaussian Mixtures with Unbalanced Mixing Coefficients. ICML'12, Edinburgh, Scotland, UK, 2012.
[13] N. T. Nguyen, R. Zheng, and Z. Han. On identifying primary user emulation attacks in cognitive radio systems using nonparametric bayesian classification. *Signal Processing, IEEE Transactions on*, 60(3), 2012.
[14] S. Rayanchu, A. Patro, and S. Banerjee. Airshark: Detecting non-WiFi RF Devices Using Commodity WiFi Hardware. IMC '11, Berlin, Germany, 2011.
[15] L. Shi, P. Bahl, and D. Katabi. Beyond Sensing: Multi-GHz Realtime Spectrum Analytics. In *NSDI'15*, Oakland, CA, USA, April 2015.
[16] M. Souryal, M. Ranganathan, J. Mink, and N. E. Ouni. Real-time centralized spectrum monitoring: Feasibility, architecture, and latency. In *IEEE DySPAN'15*, Stockholm, Sweden, 29 September - 2 October 2015.
[17] M. Wellens and P. Mahonen. Lessons learned from an extensive spectrum occupancy measurement campaign and a stochastic duty cycle model. TRIDENTCOM, Washington D.C., April, 2009.
[18] L. Xu and M. I. Jordan. On Convergence Properties of the EM Algorithm for Gaussian Mixtures. *Neural Computation*, 8:129–151, 1995.
[19] L. Yang, W. Hou, L. Cao, B. Y. Zhao, and H. Zheng. Supporting Demanding Wireless Applications with Frequency-Agile Radios. NSDI'10, San Jose, CA, 2010.
[20] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. IEEE Transactions on Information Theory 51 (7), pages 2282—–2312, July 2005.

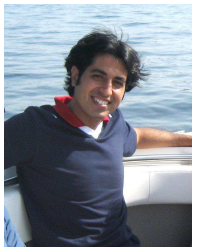
Dr. Mariya Zheleva is an assistant professor in the Department of Computer Science at University at Albany SUNY. Mariya's research interest is in the intersection of wireless networks and Information and Communication Technology for Development. She has done work on small local cellular networks, Dynamic Spectrum Access, spectrum management and sensing and network performance and characterization. She is the founder and director of the UbiNet Lab at University at Albany. Mariya is active in the spectrum research community and has served on the organizing committee of IEEE DySPAN 2017 and on multiple technical program committees.

Dr. Ranveer Chandra holds a PhD from Cornell University, 2005. He is a Principal Researcher at Microsoft Research where he is leading an Incubation on IoT Applications. Ranveer is also leading the battery research project, and co-leading the white space networking project at Microsoft Research. His research was shipped in multiple Microsoft products, including VirtualWiFi , low-power Wi-Fi, Energy Profiler in Visual Studio, and the Wireless Controller Protocol in XBOX One. He is active in the networking and systems research community, and has served as the Program Committee Chair of IEEE DySPAN 2012, and ACM MobiCom 2013.

Dr. Aakanksha Chowdhery is an Associate Research Scholar at Princeton University. She completed her PhD in Electrical Engineering from Stanford University in 2013 and was a postdoctoral researcher at Microsoft Research in the Mobility and Networking Group. Her research focuses on the network architectures and data analytics for next-generation Internet-of-Things (IoT) applications. Her work has contributed to industry standards and consortia, such DSL standards and OpenFog Consortium.

Paul Garnett is the Director of Affordable Access Initiatives at Microsoft, where he focuses on new technologies and business models that will enable billions more people to affordably get online. Paul and his team work with Internet access providers and other partners to deploy new last-mile access technologies, cloud-based services and applications, and business models that reduce the cost and improve the quality of Internet access. Prior to joining Microsoft, Paul spent 17 years in Washington, DC, where he focused on telecommunications law and policy. Paul earned his bachelor's degree in political science at Union College and his law degree at the Catholic University of America, Columbus School of Law.

Anoop Gupta currently works on infrastructure problems related to wireless communications, cloud efficiency, and scalable experimentation platforms. He manages a software development team of within Amazon working on Amazon's Forecasting Platform. Prior to joining Amazon, Anoop managed the software development team for Microsoft Research's Technology Policy Group.



Dr. Ashish Kapoor holds a PhD from the MIT Media Lab. He is a researcher with the Adaptive Systems and Interaction Group at Microsoft Research, Redmond. His research interests are in Machine Learning, Quantum Computation and Computer Vision with applications in User Modeling, Affective Computing and HCI. Currently, he is focusing on Aerial Informatics and Robotics with applications to Weather Sensing, Monitoring for Precision Agriculture and Safe Cyber-Physical Systems.

Matt Valerio is a senior software engineer at Microsoft.