A Framework for Analyzing Spectrum Characteristics in Large Spatio-temporal Scales

Yijing Zeng, Varun Chandrasekaran, Suman Banerjee, Domenico Giustiniano[§]
UW-Madison, [§]IMDEA Networks Institute
{yijingzeng,chandrasekaran,suman}@cs.wisc.edu,domenico.giustiniano@imdea.org

ABSTRACT

Understanding spectrum characteristics with little prior knowledge requires fine-grained spectrum data in the frequency, spatial, and temporal domains; gathering such a diverse set of measurements results in a large data volume. Analysis of the resulting dataset poses unique challenges; methods in the status quo are tailored for specific spectrum-related applications (apps), and are ill equipped to process data of this magnitude. In this paper, we design BigSpec, a general-purpose framework that allows for fast processing of apps. The key idea is to reduce computation costs by performing computation extensively on compressed data that preserves signal features. Adhering to this guideline, we build solutions for three apps, i.e., energy detection, spatio-temporal spectrum estimation, and anomaly detection. These apps were chosen to highlight BigSpec's efficiency, scalability, and extensibility. To evaluate BigSpec's performance, we collect more than 1 terabyte of spectrum data spanning a year, across 300MHz-4GHz, covering 400 km^2. Compared with baselines and prior works, we achieve 17× run time efficiency, sublinear rather than linear run time scalability, and extend the definition of anomaly to different domains (frequency & spatio-temporal). We also obtain high-level insights from the data to provide valuable advice on future spectrum measurement and data analysis.

CCS CONCEPTS

• **Information systems** **Spatial-temporal systems**; **Data analytics**; • **Networks** *Network measurement*.

KEYWORDS

Spectrum measurement; Spatio-temporal data analysis

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MobiCom '19, October 21–25, 2019, Los Cabos, Mexico
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6169-9/19/10…$15.00
https://doi.org/10.1145/3300061.3345450

ACM Reference Format:
Yijing Zeng, Varun Chandrasekaran, Suman Banerjee, Domenico Giustiniano. 2019. A Framework for Analyzing Spectrum Characteristics in Large Spatio-temporal Scales. In *The 25th Annual International Conference on Mobile Computing and Networking (MobiCom '19), October 21–25, 2019, Los Cabos, Mexico.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3300061.3345450

1 INTRODUCTION

The Federal Communications Commission (FCC) believes that the scarcely available spectrum will soon be insufficient to meet the ever-growing demand for mobile broadband services [9]. Therefore, there is a strong requirement to better understand spectrum characteristics. Prior work targeted at understanding spectrum utilization primarily involved large longitudinal (temporal) studies [6, 7, 40, 53, 57] or made measurements across specific (spatial) environments [14, 39, 46, 59, 61]. These efforts disregard analyzing spectrum properties that span both spatial and temporal domains. However, understanding spectrum characteristics across large areas - such as across a city or nation - and over long time periods - multiple months or years is critical. For example, a regulation authority would like to find out which frequency bands are inactive for a long period in a large area and consider opening these bands for secondary users.

We fill this crucial gap by mounting a spectrum analyzer on a bus, and collecting data as the bus travels around a city for a year. Compared with a fixed spectrum analyzer, e.g. in Microsoft Spectrum Observatory (MSO) [6], mobility does not introduce extra monetary cost but covers significantly more locations, whose spectrum characteristics can be drastically different from each other. Thus, our approach achieves a better tradeoff between cost and spatial coverage.

Nevertheless, analyzing the resulting data is even more challenging. A commercial spectrum analyzer nowadays can easily record measurement of 100MHz band with kHz resolution in several seconds; sensors used to record spatial and temporal variations can generate measurements continuously for months, at numerous locations covering a city scale area or larger. Thus, the data gathered is on the order of terabytes (TB) or more. In comparison, prior spectrum data analysis methods [14, 39, 58, 61] often operates on datasets recording channel level information and in the order of a few gigabytes (GB) or less. Therefore, they are not geared to operate on data size of this magnitude, i.e. *they are inefficient on or hardly scalable to high dimensional data of TB scale.*

Two projects based on MSO [50, 66] have attempted to infer high-level insights from large volumes of spectrum data. SpecInsight [50] analyzes the signal patterns in each band, and TxMiner [66] identifies transmitters of active signals. While these systems produce accurate results through recursively aggregating observations from small segments of spectrum data, they tend to be customized to a specific task, i.e. *they are not always designed to be extensible.*

To address these limitations, our goal is to design a general-purpose spectrum data analysis framework, through which a user can efficiently enable various applications (apps) to answer his/her spectrum related queries of interest, even if the user is not very familiar with sophisticated (big) data analytics. BigSpec achieves its desired performance through the interaction between a distributed data store, a scalable execution engine, and an extensible data pipeline (comprising of several modules). The key idea behind designing BigSpec is, in addition to utilizing standard big data techniques, *to transform the raw data into a dimension reduced space that preserves useful signal features apps can leverage, and perform extensive computations in this space.* BigSpec is easily extensible to enable various spectrum related apps, for example:

- **App1 (energy detection):** *What frequency ranges are usually active in different spectrum bands?* Inactive bands in large spatio-temporal scales are ideal for secondary users.
- **App2 (spatio-temporal spectrum estimation):** *Can the system estimate spectrum activities at an unmeasured time/ location?* We cannot measure every location and time given a limited budget of sensing platform.
- **App3 (anomaly detection):** *Are there any obvious violations that mismatch how spectrum is normally used?* We would like to detect anomalous usage of the spectrum that has limited usage of legal users, e.g. in TV channel 51 [2] and military bands.

In this paper, we discuss solutions implemented for these apps. These apps were specifically chosen to highlight BigSpec's efficiency, scalability, and extensibility.

To evaluate the performance of BigSpec, we collected spectrum data over a year, and obtained more than 1 terabyte data measured across a wide swath of spectrum (between 300MHz and 4GHz) and covering a 400 km^2 area. We also obtain high-level insights from the data, which provide valuable advice on future spectrum measurement and data analysis. The data and code are available at https://wings.cs.wisc.edu/projects/ for future extension and analysis.

In summary, the key technical contributions of this paper are the following:

- This is *the first study that analyzes long-term wideband spectrum measurement data with large-scale spatial variations* captured by a mobile spectrum analyzer.
- For **App1**, BigSpec offers *the first solution that quickly detects the active frequency ranges of different spectrum bands in large volumes spectrum data* instead of gradually aggregating observations from single measurements. Efficient algorithms that operate on compressed data are proposed to detect both spatio-temporal long-lived and short-lived energy. Compared with a baseline method (K-Means), we provide finer-grained information in spectrum utilization and 17× improvement in run time efficiency (§7.1.2).
- For **App2**, given a spectrum band and GPS information that includes time and location, BigSpec provides *the first spectrum estimation method*, using a neural network, *that considers both spatial and temporal domains with the original frequency resolution.* It achieves accuracy comparable to the state-of-the-art method (Kriging, which works for spatial or temporal estimation only), but significantly outperforms in the scalability of run time for high dimensional spectrum data (sublinear rather than linear). Furthermore, it is more robust to GPS noise (§7.1.3).
- For **App3**, BigSpec is *the first to detect two kinds of anomalies - frequency domain anomalies and spatio-temporal domain anomalies*, via extending the analysis for **App1** & **App2**. Real world examples from our dataset show that frequency domain anomalies have the potential to differentiate anomalous users from legal users that rarely use the band, and spatio-temporal domain anomalies have the potential to detect unusual usage pattern due to special events, which are impossible for previous work (§7.1.4).

Analyzing the data we collected for the three apps via BigSpec yielded the following new insights:

- Common spectrum utilization patterns observed may not comply with prior knowledge. (§7.2).
- Fine-grained spectrum estimation in large spatio-temporal scales can be hard; to improve estimation accuracy, we need a larger sensing platform of both static and mobile wideband sensing devices. (§7.2.2).
- Anomalies can be caused by sporadic legal users; a unified platform including accurate and fine-grained rule/allocation database, spectrum measurement and data analysis is needed to do illegal user detection. (§7.2.3).

2 MOTIVATION & CHALLENGES

Motivation: *"There exists rich prior knowledge as to how the spectrum is utilized"* - this fundamental assumption guides prior work [14, 39, 46, 57, 59, 61]. According to channel allocations and other rules made by the regulatory authorities, one can simply record the sum power of a given channel and make simple assumptions about its utilization pattern. For example, it is common to assume that up to a predefined distance, two locations have similar channel utilization pattern. As a result, measurements needed for prior work are in the order of a few gigabytes or less - a relatively small dataset.

However, one cannot guarantee that every spectrum user respects these rules. Hence, measurements collected could be inconsistent with any prior assumptions. Additionally, assumptions made are not universally applicable. For example, two locations within the predefined distance may, in reality, not close enough to maintain similar channel utilization pattern due to path loss or blockage of buildings. Thus, in our operational ecosystem, we assume the opposite, i.e. *there exists little prior knowledge about how spectrum is utilized.*

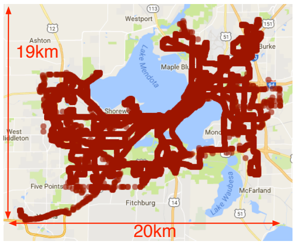


Figure 1: Locations of measurements.

ID	No. of single measurements/100MHz	Time period	Raw data size
Dataset 1	~50k	10 months	~1.1TB
Dataset 2	~20k	4 months	~470GB

Table 1: Summary of the datasets used in this paper.

Data Collection: To compensate for the aforementioned limitations, spectrum measurements in our ecosystem *record a very wide band with high resolution*, and *cover both temporal and spatial variations.* Similar to V-Scope [61], we deploy a commercial spectrum analyzer, WSA4000 from ThinkRF [11], on a metro bus traveling in and around a mid-sized U.S. city. The spectrum measurement is carried out on a per 100MHz basis (defined as a single measurement), looping from 300MHz to 4GHz (37 100MHz bands in total, defined as a single sweep). A single measurement takes 3 seconds[1] and a single sweep takes 2 minutes. For a single measurement, we record power spectrum density (PSD) data with 26215 energy readings (frequency bins). We also record the time and location information of each measurement using a GPS module. Since the bus changes its route regularly, a large fraction of outdoor city roads in and around the city is covered. Fig. 1 illustrates the position of each measurement with each dot on the map representing a single measurement, and we can see that the spatial distance between two locations can be arbitrarily small to maintain the similar channel utilization pattern. We deployed the spectrum analyzer for more than a year, and Table 1 summarizes the datasets we have gathered and used in this paper. Compared with regular wardriving, e.g. [61], we record fine-grained wideband measurements over a long period rather than channel level information for a short duration, or for specific technologies covering at least one order of magnitude less spectrum. For simplicity we denote each 100MHz band by its start frequency, i.e. 300MHz-400MHz as 300MHz band, hereafter.

Challenges: Analyzing wideband spectrum data in a large spatio-temporal scale has the following challenges.

- *High dimensional data of large size.* As one can see, the spectrum data we gathered is of high dimensionality (26215), and large size (TB). This property renders previous methods untangible to analyze the data efficiently.
- *Burstiness in temporal domain and unevenness in spatial domain.* Although we can get denser data by adding more sensors, the measurements captured using mobile sensors are always bursty in temporal domain and uneven in spatial domain. In our case, this is because measurements are only captured when the bus is operational (e.g. almost no measurements between 12AM-6AM) and the locations along popular routes (e.g. downtown area) are covered more frequently. Thus, it requires considering spatial and temporal domain as a whole.
- *Lack of prior knowledge.* For some apps, it can be hard to get the ground truth. This makes applying supervised learning techniques challenging and unsupervised/semi-supervised learning techniques more desirable.

3 BIGSPEC OVERVIEW

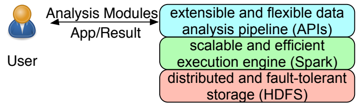

Analysis Modules
App/Result
User
extensible and flexible data analysis pipeline (APIs)
scalable and efficient execution engine (Spark)
distributed and fault-tolerant storage (HDFS)

Figure 2: Architecture of BigSpec.

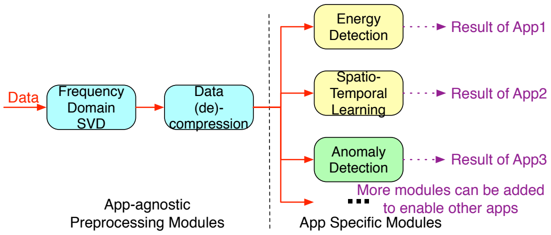


Figure 3: Data analysis pipeline of BigSpec.

We aim to design a general-purpose framework, BigSpec, that is efficient, scalable, and extensible to analyze spectrum data in large spatio-temporal scales. Performing computations on data of TB magnitude is usually ineffective on a single computer - it is bottlenecked by both relatively small amount of memory and CPU cores; prior work e.g. Electrosense [40] leverages the parallelism and consequent scalability offered by standard big data techniques running in clusters, yet limited to processing data from static sensors. BigSpec starts from a similar architecture, which is shown in Fig. 2, but makes it versatile to fulfill the challenges in § 2. At the bottom layer, we utilize a distributed and fault-tolerant file system to store the raw data. In the middle, we leverage an efficient and scalable execution engine to perform

[1] The location change during pure measurement can be ignored; most of a single measurement's time is for data recording, and sensor reconfiguration.

computation on the raw data. At the top layer, users submit their code through APIs and form an extensible and flexible data analysis pipeline. Nevertheless, a general guideline of designing the data analysis pipeline is still missing and poor data pipeline design will still result in slow computation.

The key idea of our data analysis pipeline design is to *perform dimension reduction on the raw data to transform it to a less complex space that preserves signal features, and perform computation extensively in this space.* For example, as shown in Fig. 3, in order to enable the three apps in §1, BigSpec's data analysis pipeline relies on 5 modules – (i) frequency domain singular value decomposition (SVD) (§ 4.1), (ii) data (de)compression (§ 4.2), (iii) energy detection (§ 5.1), (iv) spatio-temporal structure learning (& estimation) (§ 5.2), and (v) anomaly detection (§ 5.3). The first two modules (in blue) are for preprocessing to reduce dimensionality, while the remainder are to perform app specific computations in the dimension-reduced space (yellow blocks) or in both the dimension-reduced and the original space (green block). The data analysis pipeline for each app is a combination of preprocessing modules, and app specific modules. We describe these modules in detail in § 4 & § 5, and new modules following the key idea can always be added to the pipeline to support other apps efficiently. Moreover, although app specific modules can be affected by whether sensors are mobile or static, the app-agnostic preprocessing modules are not affected. The reason is that the preprocessing modules reduce the dimensionality while maintaining almost all useful information no matter if the sensor is mobile or static, although the meaning of each reduced dimension can be changed.

While some modules, e.g. data compression, support real-time streaming data and can be migrated to sensors, we focus on batch data in this paper because we would like to know how spectrum is utilized over the entire spatio-temporal space rather than within a short time window. However, we still have a requirement on how much time the computation should be finished, though not as strong as real-time. Since almost no measurement is collected between 12AM to 6AM, we would like the computation to be finished within a few hours so that no data backlog is built up. Moreover, because the monetary cost to run computations in cluster is usually related to the configuration of machines (memory and CPU cores), number of machines, and the duration of utilization, we believe leveraging our key idea for different apps to reduce computational cost is a good direction.

4 BIGSPEC PREPROCESSING MODULES

Preprocessing is a critical step in data mining. Since our spectrum data is high dimensional, *dimension reduction*, a form of feature extraction, is essential to realize fast analysis. Our contributions here are (i) how to determine the number of preserved dimensions to achieve a good balance between compression ratio and information loss, compared with channel allocation based compression and lossless compression, and (ii) how to interpret the preserved dimensions.

4.1 Frequency Domain SVD

We would like to have an app-agnostic dimension reduction technique so that various app specific modules can leverage its result. This requires the features (preserved dimensions) somehow reflect the wireless signals in the data so that most app specific modules can utilize them in common. Among well-known dimension reduction methods, we find truncated singular value decomposition (SVD) uniquely meets our requirement. It outputs the orthogonal directions with largest variations in data, and the large variations are actually caused by the variations of signal strength at different time and location. Thus, these features capture the wireless signals, which are frequently sensed, in several ways. This can be observed from Fig. 4, and we will explain the meaning of these features in more detail at the end of this subsection. Although truncated SVD is fairly standard, BigSpec is the first system to apply it to fine-grained measurements rather than channel occupancy [33] to the best of our knowledge, thanks to a distributed implementation that enables fast computation. Note that calculating the total power based on the channel allocation is also a method for dimension reduction. However, it has deficiencies: (i) it cannot distinguish different signals that occupy the same channel, e.g. the primary and secondary users of TV channels, and (ii) one must assume the channel allocation scheme is known beforehand and every user follows this scheme - assumptions that are not always true. We now provide more details about truncated SVD.

Truncated SVD: Consider the PSD data of each 100MHz band as a real matrix $D_{m\times n}$ with m rows and n columns, where m is the number of measurements and n is the number of frequency bins. Its truncated SVD with dimension k is

$$D_{m\times n} \approx U_{m\times k} \cdot S_{k\times k} \cdot (V_{n\times k})^T. \quad (1)$$

We focus on $V_{n\times k} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k]$, i.e. the top k right singular vectors, because they contain information about the spectrum utilization. We refer these k column vectors as the first k principal components (PCs) of $D_{m\times n}$. [2] If $k \ll m, n$ and an app specific algorithm operates on these PCs and/or the corresponding projections, it can be computed quickly.

Determining k: The challenge of using truncated SVD, and dimension reduction techniques in general, is to determine the appropriate value of k. If k is too small, important spectrum utilization information is lost; if k is very large, it severely increases the computation time without obtaining any additional useful information. Our contribution here is how to determine the appropriate k, which is as follows:

[2] Compared with PCA, truncated SVD does not remove the mean of each column. In addition, if $\mathbf{v}_i$ is the i^{th} PC of $D_{m\times n}$, $-\mathbf{v}_i$ is also the i^{th} PC of $D_{m\times n}$. To avoid this ambiguity, we multiply $\mathbf{v}_i$ by -1 if the sum of the projections of $D_{m\times n}$ on $\mathbf{v}_i$ is less than 0.

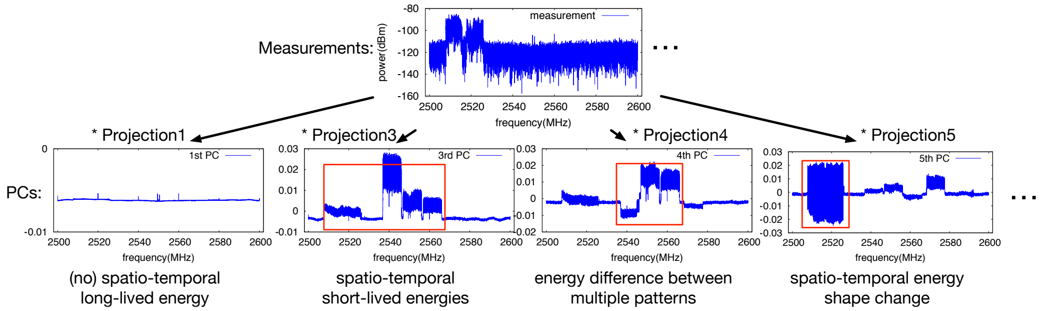
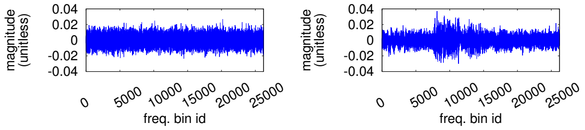


Figure 4: Illustration of BigSpec preprocessing. PCs contain spatio-temporal long-lived energies, short-lived energies, energy shape change(s), and energy difference(s) between multiple energy patterns. Each measurement is compressed as the projections on PCs.



(a) The 2nd PC of 3100MHz. Test significance is 0.9999.

(b) The 136th PC of 700MHz. Test significance is 0.9901.

Figure 5: The test significance of the PCs.

Step 1: Forward estimation based on history. Consider the optimal k, k_0, as a function of number of measurements m, i.e. $k_0(m)$. Assume that with m_1 number of measurements until time t_1, the optimal k (determined by *Step 2* below) is $k_0(m_1)$. Then, for m_2 number of measurements until time $t_2 > t_1$ (including the ones obtained up to t_1), the dimension k for truncated SVD at time t_2 is $(k_0(m_1) \cdot m_2)/m_1$. Since $k_0(m)$ is a sublinear function (refer § 7.1.5 for more details), this estimation is conservative. At the beginning when m_1 is 0 and m_2 is small (< 1000), one can set $k = m_2$, i.e. complete decomposition is computed, because in this case the computation does not consume too much time.

Step 2: Backward estimation correction to find the optimal dimension $k_0(m_2)$ that should have been used. Although we perform truncated SVD with dimension k determined in the previous step, we only keep those PCs with meaningful information, the number of which is smaller than k. We find that there exists a $k_0(m_2) \geq 1$ such that the $(k_0(m_2) + 1)^{th}$ PC through the k^{th} PC are all very similar to Gaussian white noise, and we are not interested in these PCs. Fig. 5(a) shows an example of such PC. To check if a PC is noise, we compute the Shapiro-Wilk test significance [43, 48] for the n elements of the PC. This determines if the PC can be well modeled by a Gaussian distribution. A test significance (ranging from 0 to 1) which is higher implies that the samples are more probably drawn from a normal distribution.

We compute the test significance in the reverse order, i.e., $k, k-1, \ldots, 2$. We filter out the PCs until we reach the first one whose test significance is less than the threshold of 0.99. We choose this threshold because we observe that the PCs whose test significance exceed this value do not contain noticeable signals. Fig. 5(b) shows the PC we filter out with least test significance using our data. After encountering the first one whose test significance is less than the threshold, we stop computing the test significance and keep all the remaining $k_0(m_2)$ PCs. This backward estimation correction provides the foundation for future forward estimation.

State-of-the-art SVD related works usually use one of the following two ways to determine k: (i) do a full SVD and check how many dimensions to retain so that the top k singular values contain a portion of matrix energy larger than a predefined threshold [18], and (ii) blindly apply a k beforehand [33, 47]. Compared with these two methods, our method is more suitable for noisy fine-grained spectrum measurement because it only retains PCs that are statistically not noise.

Interpretation: From Fig. 4[3] we observe that the first PC reflects the average utilization of the corresponding 100MHz band and thus can be used to detect spatio-temporal long-lived energies. For the second to the k_0^{th} PC, they usually capture three metrics - (i) the variance of the spatio-temporal long-lived energies with no notable shape change or spatio-temporal short-lived energies if there is no long-lived energy in a 100MHz band, (ii) the shape change of the spatio-temporal energy, and (iii) if there are multiple spatio-temporal energy patterns occupying different frequencies in a 100MHz band, a PC can also capture the energy differences between multiple energy patterns, which is common for bands that have energy detected. We offer our approach to detect notable spatio-temporal energy from the PCs in § 5.1.

4.2 Data (De)compression

The purpose of our compression module is not to efficiently reduce the size of spectrum data only. More importantly, it enables transformation of the data into a less complex space where we can obtain insights efficiently. Note that

[3] The 2nd PC in Fig. 4 is omitted because it only contains one spatio-temporal short-lived energy occupying 2505-2525MHz but not multiples ones, which is less representative compared with the 3rd PC.

Algorithm 1 Spatio-temporal long-lived energy detection

```
Input: $\widetilde{\mathbf{v}}_{300}, \widetilde{\mathbf{v}}_{400}, ..., \widetilde{\mathbf{v}}_{3900}, N$
Output: $L$
$i \leftarrow 1, L \leftarrow \emptyset$
while true do
    Run K-Means($\widetilde{\mathbf{v}}_{300}, \widetilde{\mathbf{v}}_{400}, ..., \widetilde{\mathbf{v}}_{3900}$) with $i$ centroids
    if all $\widetilde{\mathbf{v}}_j, j \in N$ are in the same cluster (say, id $k$) then
        $i \leftarrow i + 1$
        $L \leftarrow$ all 100MHz bands not in cluster $k$
    else
        return $L$
    end if
end while
```

traditional lossless compression does not work well on our dataset because of noise in the measurements. Moreover, decompression is also needed for visualization and some apps, e.g. reconstruction error based anomaly detection.

Compression: After frequency domain SVD and the steps described in § 4.1, we obtain a matrix $V_{n \times k_0}$, which consists of all retained PCs, for each 100MHz band. We can then compress the dataset and reduce its dimension by calculating the projection of $D_{m \times n}$ on each PC using

$$C_{m \times k_0} = D_{m \times n} \cdot V_{n \times k_0}. \tag{2}$$

Ideally, the compression ratio of keeping $C_{m \times k_0}$ and $V_{n \times k_0}$ instead of $D_{m \times n}$ is $mn/((m+n)k_0)$. However, we cannot achieve this in practice because we need to retain the exact GPS information of each measurement.

Decompression: To decompress, we compute

$$D'_{m \times n} = C_{m \times k_0} \cdot (V_{n \times k_0})^T. \tag{3}$$

Since the compression is lossy, $D'_{m \times n}$ is only approximately equal to $D_{m \times n}$. In fact, it is the best rank k_0 approximation of $D_{m \times n}$. The reconstruction error $E_{m \times n}$ is defined to be

$$E_{m \times n} = D'_{m \times n} - D_{m \times n}. \tag{4}$$

5 BIGSPEC APP SPECIFIC MODULES

In this section, we present our algorithms for the three apps proposed in §1. Although the algorithms are app specific, they follow the same key idea, i.e. performing most of the computation in the dimension-reduced space.

5.1 Energy Detection

Detecting energies of wireless transmitters from a single spectrum measurement has been widely studied in previous work. However, little work has been done on directly detecting energies from a large number of measurements. Following the key idea of BigSpec, our technique directly infers the existence of energies for each 100MHz band from the PCs, and significantly outperforms the algorithms that operate in the uncompressed space in run time efficiency. We classify the energies of wireless transmitters into two categories - (i) spatio-temporal long-lived energies are those that persist regardless of variations in time or location[4], and (ii) spatio-temporal short-lived energies are those that can be observed frequently, but not at every location/time. Note that "long-lived" and "short-lived" are not solely referring to the temporal domain, but the spatio-temporal domain.

Long-lived energies: We first remove any artifact introduced by the sensing device, i.e. perform noise floor shape extraction, before we detect long-lived energies based on the first PCs of all 100MHz bands. As we will show in § 7.1.2, this has a large impact on detecting the active frequency range of low power long-lived energies[5]. To summarize the method of noise floor shape extraction, we identify the 100MHz bands that have only one PC retained (meaning no spectrum activity is observed), and form a matrix with n columns using the first PCs of these 100MHz bands. We then calculate the first PC of this matrix using SVD, similar to the method in § 4.1, to get the common noise floor of the sensing device. We finally remove its effect on each 100MHz band by subtracting its projection from every first PC of all 100MHz bands.

We now explain how we obtain the 100MHz bands that contain long-lived energies. Let $\widetilde{\mathbf{v}}_{300}, \widetilde{\mathbf{v}}_{400}, ..., \widetilde{\mathbf{v}}_{3900}$ represent the first PC of 300MHz, 400MHz, ..., 3900MHz after noise floor shape extraction respectively. Assume set N contains the 100MHz bands that retain only the first PC. We get the set L that includes all 100MHz bands that have long-lived energies using Algorithm 1. The intuition of Algorithm 1 is that the first PC after noise floor shape extraction of a 100MHz band with long-lived energies should be significantly different from that of a 100MHz band with no spectrum activities observed. We use the K-Means algorithm to capture the difference, and stop increasing the number of centroids when the similarity within the set of all 100MHz bands with no spectrum activities observed breaks down.

Short-lived energies: We focus on the 100MHz bands for which no long-lived energy is detected and multiple PCs are retained. Let $C_{m \times k_0} = [\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_{k_0}]$, where $\mathbf{c}_i$ represents the projections of $D_{m \times n}$ on the i^{th} PC. Consider the m elements in $\mathbf{c}_i$ as the m samples of a random variable $\mathbb{C}_i$. We observe that the correlation between $\mathbb{C}_i (i > 1)$ and $\mathbb{C}_1$ is relatively stronger (absolute value no less than 0.1) if the i^{th} PC captures short-lived energies. Otherwise, the correlation is relatively weaker (absolute value smaller than 0.1). This is because the projections on the PCs capturing spatio-temporal energy shape changes are only weakly correlated with the projections on the first PC, if they are correlated at all. However, if the i^{th} PC captures short-lived energies, then the bigger projection on the i^{th} PC requires the smaller projection on the first PC in order to make up for the stable noise

[4] We do not distinguish energies that have the same frequency domain features from different transmitters for this module.

[5] More likely they are short-lived energies in the band with long-lived energies. However, our goal is to detect active frequency ranges. Thus, we do not distinguish these two carefully.

floor value. This also implies that the stronger correlation is the one that is negative.

Note that the above algorithms only output the PCs that contain spatio-temporal long-lived and short-lived energy patterns. Currently, we manually get the active frequency ranges these patterns occupy from the PCs. Obtaining the active frequency ranges automatically needs further analysis and we leave it as future work; with little prior knowledge, it is hard to find a general method that works well for all kinds of energy patterns, e.g. narrowband or wideband, low-power or high-power, etc.

5.2 Spatio-Temporal Structure Learning

Considering spatial and temporal domains as a whole is necessary for spectrum estimation due to the fact that the data is bursty in temporal domain and uneven in spatial domain. However, previous spectrum estimation efforts e.g. [35, 57] either estimate the sum power or occupancy of a particular channel at different locations without considering temporal variations, or estimate its variations at different timestamps regardless of spatial variations. Considering spatial and temporal domains together is difficult using state-of-the-art interpolation methods such as Kriging [14, 39, 58]. These methods are all based on the assumption that a location's reading can be inferred from measurements that are at a small distance from this location. Although the distance here is clearly defined only for the temporal domain or the spatial domain, it is hard to accurately define distance in spatio-temporal domain based on (time, location) coordinates. In addition, Kriging has several other limitations. To the best of our knowledge, Kriging does not have a distributed implementation that works well for large data volumes. It also fails if two measurements have the same spatio-temporal coordinates since the Kriging matrix is non-invertible in this case. Last and most importantly, it cannot be parallelized if the number of output dimensions is more than one, which makes estimation at a frequency bin level challenging.

NN Configurations: In order to address the limitations of current spectrum estimation methods, we decide to use a feedforward neural network (NN) for each 100MHz band to learn the spatio-temporal structure by formulating it as a regression problem. We do not use more advanced NN structures, e.g., Convolutional LSTM [55], because these structures usually cannot handle the burstiness in temporal domain and unevenness in spatial domain. For the inputs, we convert the Unix timestamp of each measurement into time of day, day of week, date in the month, and the month. We also add latitude and longitude as spatial-related input features and exclude other GPS readings, i.e. altitude and speed. For the outputs, we choose the projections of the dataset on each PC. Since we have reduced the number of dimensions of the data in the earlier preprocessing step, it provides superior scalability in run time. An essential design choice in using a NN is selecting the activation function for the neurons, and the loss function for stochastic gradient descent (SGD). We choose Rectified Linear Unit (ReLU) as the activation function for fast convergence rate in all layers except for the output layer, which uses a linear function. For the loss function, we first decompress the estimation error vector in the compressed space and then use the sum of squared error (SSE) of the estimation error vector in the uncompressed space. Decompression is important as the estimation accuracy in the uncompressed space is what we really care about.

Benefits of NN: Using a NN overcomes the limitations of Kriging. First, via feature standardization, we need not worry about how to define distance in spatio-temporal domain. Second, NNs have efficient and distributed implementations e.g. Tensorflow [13]. Third, for two measurements having the same spatio-temporal coordinates, gradients can still be computed during the SGD process. Finally, since each output neuron shares the same previous layers, the learning process can be parallelized if the number of output dimension is more than one. Another advantage of using a NN compared to Kriging is that the NN is more robust to GPS noise because input noise is equivalent to a form of regularization [17].

5.3 Anomaly Detection

We would like to stress that anomaly detection is not equivalent to illegal user detection because of little prior knowledge. Illegal user detection needs further verification, which requires a human-in-the-loop, so naive strategies such as observing all measurements for every detected energy are not scalable. Therefore, filtering energy patterns which are more likely to come from illegal users based on statistics is necessary. This is the goal of our anomaly detection.

Although anomaly detection has been well studied in the context of cooperative spectrum sensing [15, 24, 34, 54], previous work focuses on the data that records the total power of a particular channel. We, however, investigate how to detect anomalies in high dimensional spectrum data. We define two types of anomalies by extending previous analysis: (a) a frequency domain anomaly is a point anomaly, where a single measurement can be considered anomalous with respect to the rest of the data, and (b) a spatio-temporal domain anomaly is a contextual anomaly, where a single measurement is anomalous in a specific spatio-temporal context. The benefit of distinguishing these two types of anomalies is that frequency domain anomaly has the potential to differentiate anomalous users from legitimate ones that seldom use the band (i.e. sporadic legal users), and spatio-temporal domain anomaly has the potential to detect unusual usage pattern due to special event.

Frequency domain anomaly: We detect frequency domain anomaly for each 100MHz band based on the reconstruction error (eq. 4). Let $\mathbf{e}_i$ be the i^{th} row of $E_{m\times n}$ in eq. 4, $\bar{\mathbf{e}}$ be the mean of all $\mathbf{e}_i$, and $\mathbf{e}_{std}$ be the standard deviation of all $\mathbf{e}_i$. Anomaly detection is typically computed as some function of the L2 or L1 norm of $\mathbf{e}_i$ [21, 41] with normal distribution assumption of anomaly score; we, however, use a different metric and define anomaly score

$$\tilde{e}_i = \|(\mathbf{e}_i - \bar{\mathbf{e}}) \oslash \mathbf{e}_{std}\|^2, \tag{5}$$

where $\oslash$ denotes element-wise division and $\|\cdot\|$ represents the L2 norm. The reason for this is that $\tilde{e}$ should follow a Chi-square distribution with degree of freedom n as $\mathbf{e}_i - \bar{\mathbf{e}}$ should be a Gaussian random vector in the normal case, similar to [32]. We can then apply a threshold on the anomaly score and determine a measurement as anomalous if its anomaly score is greater than the threshold.

Spatio-temporal domain anomaly: We directly detect spatio-temporal domain anomaly for each band based on the estimation error in compressed space produced by the NN. The method for detecting spatio-temporal domain anomaly is similar to frequency domain anomaly detection, but simply modifying the dimension of the vectors to length k_0 is not accurate. In the frequency domain anomaly case, $\mathbf{e}_i - \bar{\mathbf{e}}$ is a Gaussian random vector, where we assume that the errors added to different dimensions are not correlated. However, this is not true in the spatio-temporal anomaly setting; in the NN, each output neuron shares the same previous layers, introducing correlation. To address this issue, we need to whiten (decorrelate) the errors of different dimensions before calculating the anomaly score, and we use ZCA whitening [16] to achieve this in our system.

Note that the optimal thresholds on the anomaly scores are not determined due to the unsupervised nature. Initially, one can determine the threshold by indicating a significance factor that represents the probability that an anomaly has occurred, and then check the value of the inverse survival function of Chi-square distribution corresponding to this significance factor to get the threshold. After having an initial impression of the optimal thresholds, more advanced techniques can be used to determine the optimal thresholds using (semi-)supervised learning, e.g. SVM.

6 IMPLEMENTATION

Implementation of BigSpec requires a cloud infrastructure described in Fig. 2. While any compatible systems can be used to implement the various modules and layers of BigSpec, we discuss the specifics of our implementation.

We configure an 8-node cluster in CloudLab [1, 44]; each node has two 14-core 2.00 GHz Intel CPUs, 256GB RAM, and dual-port Intel 10GbE NIC. On top of the cluster, we install CDH5 from Cloudera [22], which integrates distributed fault-tolerant storage HDFS [51], scalable in-memory execution engine Spark [8], etc. We also install Tensorflow [13] as the execution engine for running the NNs. The data pipeline is realized using a combination of Scala and Python. Note that Spark supports both Scala and Python, but the choice of language has a direct implication for the run time. If a module requires direct operations on the uncompressed data, we choose Scala because of its superior run time. However, if a module does the computation on the compressed data, we can implement the module using Python because of greater flexibility and the various packages offered. Additionally, the performance of Spark is related to several other issues, e.g. the number of partitions of the data, the memory configuration. We do not present the details here because it is beyond the scope of this paper.

7 EVALUATION AND RESULTS

The purpose of our evaluation is two-fold. First, we evaluate BigSpec using the datasets gathered (§ 7.1) to highlight the ease of the respective solutions we built, which is impossible with earlier work. Second, we present high-level insights from our data (§ 7.2). Due to space limitations, we only report results using dataset 1, except for § 7.1.5 and § 7.1.6.

7.1 Evaluation of BigSpec

7.1.1 System Level Performance.

Run time breakdown: Recall that we do not collect data between 12AM-6AM every day; during this period, the data analysis is performed. Thus, there is an implicit constraint that the data analysis pipeline should be completed within several hours. Fig. 6 shows the run time breakdown of each module. From Fig. 6 we see that the total data analysis time can meet our requirement, and SVD is the most time-consuming module - it involves operations on the uncompressed data. Moreover, energy detection can be completed within 3 minutes because its computation is performed purely in the compressed space. Note that the computation time of spatio-temporal learning and estimation is adjustable based on the batch size of SGD and the number of epochs that learning is performed, such that it is not the bottleneck.

7.1.2 Energy Detection.

Ground truth and baseline method: We obtain ground truth by manually investigating all the retained PCs of all 100MHz bands, and determining whether there is a signal, and its starting and ending frequencies if there are any. We also confirm our observations using the FCC allocation chart [3] and an online spectrum wiki [10]. As a baseline (running in cloud), we run K-Means on the columns of the data matrix $D_{m\times n}$ of each 100MHz band for $k = 1, 2$ respectively, and calculate the within set sum of squared errors (WSSSE), the sum of the distances from each observation to its centroids in all partitions, for each k. If the WSSSE when $k = 2$ is smaller than the WSSSE when $k = 1$ times a constant between 0 and 1 (0.95 in practice, which is empirically optimized based on the ground truth), then there are

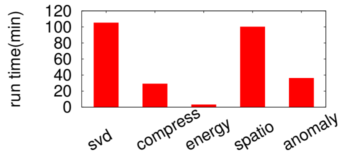


Figure 6: Run time breakdown

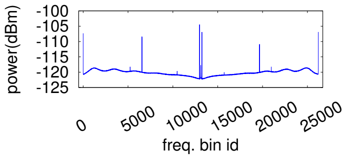


Figure 8: Extracted noise floor

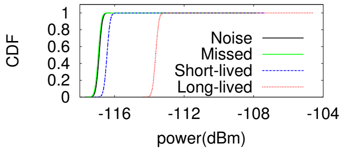
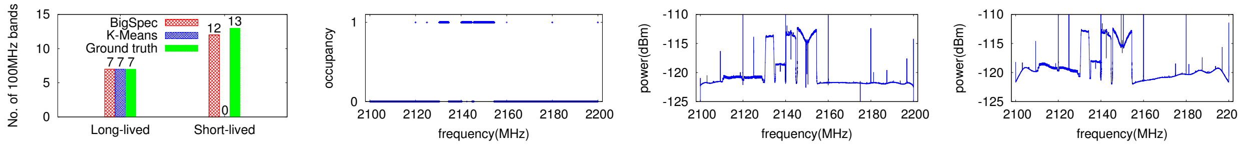


Figure 9: SNR sensitivity



(a) No. of 100MHz band with detected energies.

(b) Detected long-lived energies using K-Means.

(c) Detected long-lived energies using BigSpec.

(d) Long-lived energies w/o noise floor extraction.

Figure 7: BigSpec energy detection.

energies from wireless transmitters in this 100MHz band.[6] In this case, the cluster of columns (frequency bins) with higher mean power contains the energies. Note that the baseline does not follow the key idea of BigSpec; it does computations on uncompressed data.

Observations: *1. Detected Energies.* Fig. 7 compares the performance between BigSpec energy detection and K-Means. Fig. 7(a) shows that both BigSpec and K-Means detect all 100MHz bands that contain (multiple) long-lived energies. These 100MHz bands are 300MHz (satellite), 500MHz (TV), 600MHz (TV), 700MHz (LTE), 800MHz (GSM), 1900 MHz (PCS), and 2100MHz (PCS). However, K-Means cannot detect *any* short-lived energies. BigSpec, on the other hand, can detect 12 out of 13 100MHz bands that contain (multiple) short-lived energies. Detailed results of short-lived energies are omitted here due to space constraints. BigSpec misses out on the 3600MHz (CBRS) band, and we observe that the correlation coefficient computed is -0.099, which is slightly greater than the threshold -0.1. We believe that this threshold can be further optimized through supervised learning.

Moreover, even for the long-lived energy detection, K-Means cannot offer as much detail as BigSpec, e.g., signal features and modulation scheme, relative power comparison between multiple signals. Fig. 7(b) shows the result of K-Means for 2100MHz. We can see that it only offers occupancy information and three energy patterns, i.e., 2130-2135MHz, 2140-2145MHz, and 2145-2155MHz, are detected. However, as Fig. 7(c) shows, for BigSpec, three more energy patterns are detected, which are 2110-2120MHz, 2120-2130MHz, and 2135-2140MHz. In addition, frequency bin level details are maintained using BigSpec.

2. Computation Time. The run time of K-Means is 51 minutes, and the run time of BigSpec without counting preprocessing is only 3 minutes, which is 17× smaller than that of K-Means. This exemplifies the efficacy of BigSpec's key idea, i.e., performs computations on compressed data. Even if the run time of preprocessing is counted, the run time of BigSpec and K-Means are still comparable, with the benefit that other apps can access the same compressed data.

Effect of noise floor extraction: Fig. 8 suggests that the noise floor in our device is not flat.[7] We believe this is not a unique phenomenon specific to our device because of the nonlinearity in devices, and the unavoidable effect of adding a time window before performing a FFT. We also notice that there are some peaks in addition to the slowly varying noise floor, which means that the readings of these particular bins may be unreliable. As a result, if the noise floor is not removed for long-lived energy detection, we will lose some information. For example, Fig. 7(d) shows the first PC of 2100MHz without noise floor removal. Compared with Fig. 7(c), it is clear that detecting 2110-2120MHz and 2120-2130MHz energy patterns is hard if the noise floor is not extracted.

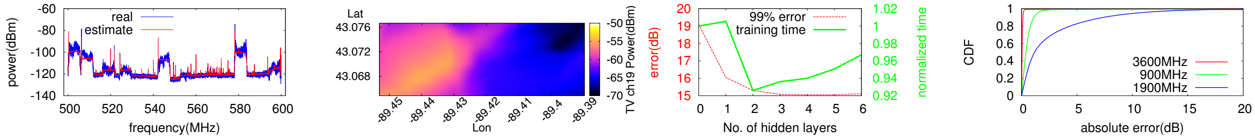
SNR sensitivity: We evaluate BigSpec energy detection's SNR sensitivity using 3600MHz data because it has only one transient energy pattern, 3650-3660MHz, which is missed by BigSpec. We keep artificially adding the same energy to the readings of 3650-3660MHz in all measurements until it can be detected as short-lived and long-lived energy. Fig. 9 shows the result. We can see that the original missed energy pattern has a similar CDF with noise except for a large tail of 8dB. When 0.5dB/3.3dB more energy per measurement is added, it is detected as short/long-lived energy respectively.

7.1.3 Spatio-Temporal Structure Learning.

Method to get bin-level estimation precision: We run 10-fold cross validation on the projection data of a particular band ($C_{m \times k_0}$) as well as the GPS readings of measurements,

[6] This method can be generalized to keep increasing the number of centroids until the WSSSE stops decreasing significantly, and theoretically all energies from wireless transmitters can be found in this way. However, we empirically find that the stop condition is met when $k = 2$.

[7] The algorithm's output is a unit vector. We time it with average projection for better visualization. Similar for Fig. 7(c), 7(d), & 17.



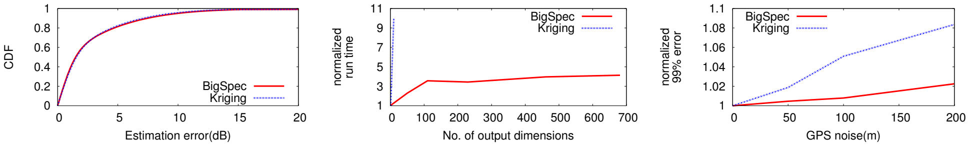
(a) Example of spatio-temporal estimation.

(b) Example of spatial heatmap.

(c) Influence of hidden layers.

(d) Error distribution.

Figure 10: BigSpec spatio-temporal estimation.



(a) Accuracy.

(b) Run time vs. output dimensions.

(c) Robustness against GPS noise.

Figure 11: Performance of BigSpec spatio-temporal estimation compared with Kriging using 500MHz data.

and decompress the estimated projections back into bin powers for each measurement. We then compare the estimated bin powers of each measurement with the power readings of the same measurement after compression/decompression ($D'_{m\times n}$). Fig. 10(a) shows an example of the spatio-temporal estimation as well as the corresponding real measurement after (de)compression. One can see that we can estimate the spectrum usage at the frequency bin-level and preserve details, e.g., pilots of TV signals. Spatial heatmap (e.g. Fig. 10(b)) and temporal waterfall chart (omitted due to space constraint) can also be easily generated based on the NNs.

Number of hidden layers: Adding more hidden layers can improve the accuracy of the NN at the expense of increased training time. To that end, we vary the number of hidden layers in the NN to measure the trade-off between the 99th percentile error and training time. Fig. 10(c) shows the results using 500MHz band data. It can be seen that adding more hidden layers indeed improves the precision of estimation. However, when the number of hidden layers is greater than 3, the 99th percentile error stops improving significantly, and the training time starts to increase unfavourably. For the rest of this subsection, we fix the number of hidden layers at 3.

Estimation accuracy: We apply the aforementioned method to get bin-level estimation precision of all the bands. Fig. 10(d) shows the CDF of the absolute error of spatio-temporal structure learning and estimation. The red, green, and blue lines represent the best, median, and worst case of the bands with long-lived or short-lived energies. The results for other bands, which are better than the best case here, are not shown for clarity of presentation. The definitions of the best, median, and worst cases are in terms of the 99th percentile value. We can see that in the best case, the NN can make near perfect estimations. In the median case, the 99th percentile absolute error is less than 3dB. The NN produces less accurate estimations for the GSM and PCS bands, which have multiple long-lived active and time-varying channels. The 99th percentile absolute error is around 17dB. This estimation error may seem high because these bands have random utilization nature while NN can only output a deterministic estimation. Furthermore, since we use cross validation, which divides dataset randomly rather than spatial/temporal distance-based[8], the estimation result can be affected by outdated history pattern or significantly different future pattern. In terms of the type of PCs, NN makes less accurate predictions of the projections on the PCs that capture energy shape changes, which can be observed from Fig. 10(a). The reason is that the projections on these PCs are relatively random, and independent of input features.

Comparison with Kriging: In practice, Kriging only works for spatial interpolation or temporal interpolation, but not spatio-temporal interpolation. Thus, we choose the 500MHz TV band dataset as it has large spatial variations with little temporal variation. Note that we have multiple preserved dimensions in the data; thus the Kriging method we implement interpolates each preserved dimension, then decompresses the aggregated results to obtain the final results. We also add a small random noise to the latitude and longitude readings if needed to ensure that the Kriging matrix is invertible. Moreover, to enable a fair comparison, the NN to compare with Kriging only takes spatial coordinates as input. Our observations are as follows:

1. Accuracy. Fig. 11(a) compares the CDF of estimation error for BigSpec and Kriging. It can be seen that BigSpec achieves very similar levels of accuracy with Kriging.

2. Run time vs. output dimensions. 500MHz data has 681 dimensions retained after frequency domain SVD; larger the number of dimensions needs longer run time (training and

[8] Dividing the dataset based on spatial/temporal distance and then evaluating estimation accuracy is hard for our dataset due to its unevenness/burstiness in spatial/temporal domain.

inference). Fig. 11(b) shows the normalized run time as a function of the number of output dimensions. From Fig. 11(b), we can see that run time grows linearly for Kriging but sublinearly using the NN. This demonstrates BigSpec's superior scalability and the significance of performing computation on compressed data that retain signal features. For absolute run time, running a NN with 681 dimensions takes 2 hours, which is the same as Kriging with 3 dimensions.

3. Robustness against GPS noise. Another benefit of using NN over Kriging is its robustness against GPS noise. GPS readings can be noisy due to the blocking of signals from buildings. We characterize its influence by injecting noise to the spatial coordinates with uniform random direction from 0 to 360 degrees and uniform radius from 0 to a varying maximum value. As Fig. 11(c) shows, with growing maximum radius of injected GPS noise, the change in 99th percentile error of BigSpec is significantly smaller than that of Kriging.

7.1.4 Anomaly Detection.

Comparison with baselines: We compare our (frequency) anomaly detection method against ones using L2 and L1 norm to compute the anomaly score with normal distribution assumption, which are similar to recent work SAIFE [41] (L1 norm based). After computing the anomaly score for each measurement, we order the measurements by their scores and obtain the top 50. We obtain the ground truth by manually comparing each measurement with its reconstructed signal in terms of frequency domain features and use the following two metrics to compare the sensitivity and specificity: (i) number of true positives (TPs) before the first false positive (FP), and (ii) number of false positives (FPs) in the 50 positive (P) outputted anomalies. From Fig. 12(a) we can see that BigSpec achieves better performance in both metrics.

Frequency domain anomaly example: Fig. 12(b) and 12(c) show a frequency domain anomaly and a spatio-temporal domain anomaly in the 600MHz TV band respectively. In 2011, the FCC freezed all future applications for broadcast stations requesting to use channel 51 (692-698MHz) to prevent interference to the A-Block of the 700MHz LTE band [2]. If we are only provided with the sum power of the TV channel, we will be unable to correctly determine whether the anomaly is due to an anomalous user or a legacy TV user. However, as shown in Fig. 12(b) and 12(c), because BigSpec preserves the frequency bin level details, we can manually differentiate these two without difficulties. A TV signal should occupy 6MHz bandwidth and have a rectangular-shape in frequency domain but the frequency domain anomaly shown in Fig. 12(b) contains a 2MHz bandwidth spike-shape signal (in green box). Thus, we are sure the frequency domain anomaly shown in Fig. 12(b) is caused by a non-TV signal. This proves that frequency domain anomaly has the potential to differentiate anomalous users from legal ones that seldom use the band.

Spatio-temporal domain anomaly example: We notice that at 9PM, Aug. 7th, there is a burst of spatio-temporal anomalies in both 700MHz (LTE) band and 800MHz (GSM) band. The real measurements show that these two bands are busier than the expectations of the NNs. Therefore, one convincible explanation is that there is a special event so that a lot of people gathered at a location which is near the measurement location, at that time. We obtained the location of the anomalies, and manually found the special event that is mostly likely to cause these two anomalies among all events documented on the local website. It was an event for LGBT community with over 100 attendants [12] within 250 meters at that time. Fig. 12(d) illustrates the locations of the spatio-temporal anomaly and the local event. Although we are not 100 percent certain this is the root cause of the spatio-temporal domain anomaly burst (which is a common problem for using real world measurements e.g. in [63]), we believe this shows that the spatio-temporal anomaly has the potential to detect unusual usage pattern due to special event.

7.1.5 Frequency Domain SVD.

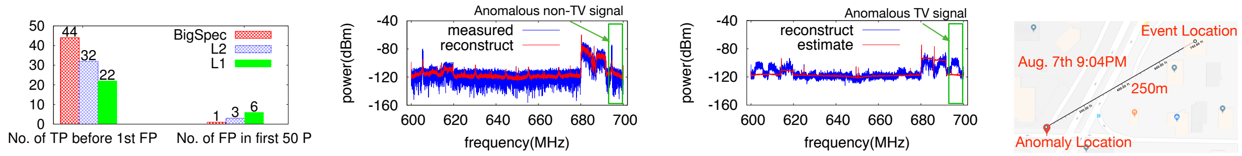
Number of retained PCs after backward estimation correction: Fig. 13 shows the CDF of the number of retained PCs in the two datasets. From Fig. 13, we can see that 80% of the 100MHz bands have less than or equal to 100 PCs retained. In the worst case, dataset 1 has fewer than 700 PCs retained and dataset 2 has fewer than 350 PCs retained. These two numbers are still significantly smaller than the original 26215 dimensions.

Accuracy of forward estimation: Fig. 14 compares the results of forward estimation (red line) and backward estimation correction (blue line) using 500MHz data as an example. The black arrows show the process of forward estimation and backward correction. From Fig. 14, it can be seen that forward estimation is indeed a conservative estimation. When the number of measurements is small, the difference between forward estimation and backward estimation correction is relatively large. This is tolerable because when the number of measurements is small, the time to compute SVD is also relatively small. On the other hand, when the number of measurements is large enough, we can see that forward estimation is a very tight bound of backward estimation correction. Furthermore, the blue line in Fig. 14 also shows that k_0 grows sub-linearly as a function of the number of measurements m. (Note that it also goes through (0,0), (1,1).)

7.1.6 Data Compression.

Performance comparison with baselines: We choose a lossless general compression Gzip [5, 38] and a lossy compression for spectrum data Airpress [65] as the baselines. We evaluate both compression ratio and compression time.

1. Compression ratio. Fig. 15(a) shows the CDF of the compression ratio of each 100MHz band for dataset 2. From Fig. 15(a),



(a) Comparison with baselines for frequency domain anomaly.

(b) Example of frequency domain anomaly.

(c) Example of spatio-temporal domain anomaly.

(d) Anomaly and special event.

Figure 12: BigSpec anomaly detection.

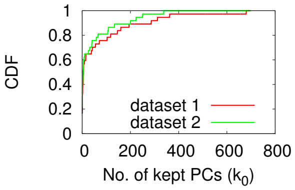


Figure 13: CDF of no. of retained PCs.

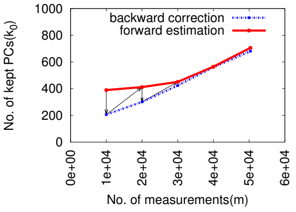


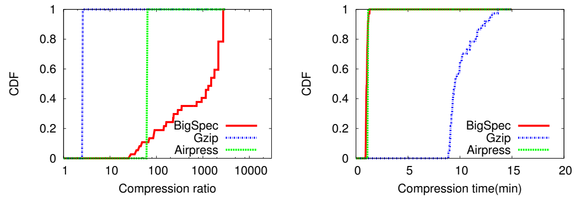
Figure 14: Accuracy of forward estimation for k_0.

it can be seen that for 80% of the 100MHz bands, the compression ratio of BigSpec is greater than 100. In the worst case, the compression error is still around 30. For comparison, the compression ratio of Gzip is only around 2.5 and Airpress is around 64, and our median compression ratio is 10^4× better than Gzip and 25× better than Airpress. This suggests lossy compressions are more suitable than lossless ones for spectrum data. In addition, for 90% 100MHz bands, BigSpec has better compression ratio than Airpress.

2. Compression time. Fig. 15(b) shows the CDF of the compression time of each 100MHz band. We can see that the compression time of BigSpec and Airpress is around 1 minute, and the median compression time of Gzip is 10× higher. This shows that by using lossy compression, we also significantly reduce the compression time.

Compression error: *1. Distribution.* We benchmark the error introduced by compression/decompression using dataset 2 in Fig. 16. Fig. 16(a) illustrates the CDF of the absolute error for each frequency bin. The red line represents the best case, and the green line represents the worst case. The definitions of the best and worst cases are in terms of 99th percentile error. Fig. 16(a) shows that the worst case is very close to the best case, which means the error introduced by compression/decompression is equivalent in all the bands. In Fig. 16(a), the 99th percentile absolute error is around 17dB. Although this number may seem high, the compression still maintains all useful information in the data except for a few frequency domain anomalies, e.g. Fig. 12(b), and the error is introduced almost entirely by noise.

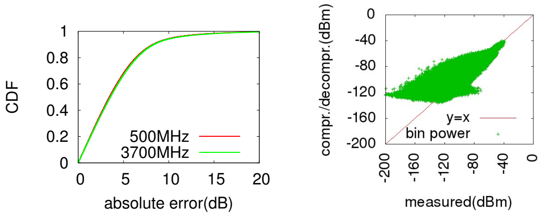
2. Error pattern. We also evaluate the error pattern introduced by compression/decompression. Ideally, we would like the error to be small for measured high energy bins, and we can tolerate more error for very low energy bins. Fig. 16(b) shows



(a) CDF of compression ratio.

(b) CDF of compression time.

Figure 15: Performance comparison against baselines.



(a) CDF of absolute error.

(b) Error pattern of 500MHz.

Figure 16: Compression error.

the error pattern of 500MHz in dataset 2 as an example. The x axis is the measured bin power and the y axis is the bin power after reconstruction. From Fig. 16(b) we can see that BigSpec indeed achieves lower error for high energy bins.

7.2 Insights from the Data

We conclude our evaluation by discussing the new insights obtained using BigSpec, which differ from the assumptions/conclusions of prior works. These insights provide valuable advice on future spectrum measurement and data analysis.

7.2.1 Energy Detection.

It is not unusual that common spectrum utilization pattern does not comply with prior knowledge. Previous work e.g. [14, 39, 57, 61, 65] implicitly assume all legal users follow channel allocations/rules made by the regulation authorities so that this rich prior knowledge makes coarse spectrum measurement good enough. However, we find that this is not the case and use 500MHz TV band as an example, whose first PC is shown in Fig. 17. According to FCC's allocation, starting from 500MHz, each TV channel occupies 6MHz and is adjacent to each other, i.e. 500-506MHz, 506-512MHz, etc. Nevertheless, from Fig. 17 we can see that 5 detected long-lived energy patterns from TV signals (in red boxes) do not

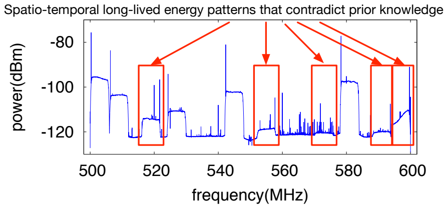


Figure 17: Long-lived energy patterns that contradict the prior knowledge in 500MHz TV band.

comply with this prior knowledge. In other words, if one uniformly samples a location/time in the spatio-temporal space we measured, the expectation of spectrum utilization contains 5 TV signals that do not comply with prior knowledge. Moreover, their pilot tones are close to the upper edges of the channels rather than the lower edges, which also contradicts with the prior knowledge.

7.2.2 Spatio-temporal Structure Learning.

Fine-grained spectrum estimation in large spatio-temporal scales can be hard, and we need a larger sensing platform of both static and mobile wideband sensors to improve accuracy. Previous works [14, 39, 57, 58] show that coarse spectrum estimation is accurate in small spatio-temporal scales. However, this is not always true for fine-grained estimation in large spatio-temporal scales, where we can have large tail error for dynamic bands, e.g. cellular bands. The 99th percentile error for 1900MHz PCS band can be as high as 17dB. To improve accuracy and identify spatio-temporal patterns accurately, we need denser data in spatio-temporal domain; this requires a larger sensing platform with multiple sensors. Current spectrum measurement efforts usually use static sensors. They have good temporal coverage but lack spatial coverage. On the other hand, a sensing platform of a few mobile sensors has the opposite property. Therefore, we need both static and mobile sensors to complement each other. Moreover, there is a tradeoff between cost and bandwidth/resolution of sensors. This further requires the platform can handle data from sensors of different quality, which we will discuss more in § 8.

7.2.3 Anomaly Detection.

Anomalies can be caused by sporadic legal users; a unified platform including accurate and fine-grained rule/allocation database, spectrum measurement and data analysis is necessary to do illegal user detection. Contrary to prior works [15, 19, 24, 29, 34, 54] where anomalies are assumed to be caused by malicious illegal users, a large fraction of detected anomalies is in fact sporadic legal users of the spectrum; for example, frequency domain anomalies detected in 2400MHz band are mostly Bluetooth signals. Thus, for an illegal user detection system, semi-supervised learning with a small percentage of labeled data is more realistic. Our anomaly detection method in fact provides the advice on what data are more desirable to be labeled/further verified, as long as illegal users rarely appear. However, labeling requires accurate and fine-grained prior knowledge, but currently we only have the FCC allocation chart and online documentation about (i) how particular frequency bands are allocated, and (ii) for what services, which is very coarse information. We do not know what types of signals can be transmitted and are being transmitted for every time, location, and frequency band. Therefore, we need a unified platform that combines accurate and fine-grained rule/allocation database, spectrum measurement and data analysis. This accurate and fine-grained rule/allocation database enables us to query about who can utilize a specific band with what regulation constraints are in play at a specific location/time accurately, which is essential for further verification to accurately identify illegal users.

8 DISCUSSION AND FUTURE WORK

Other measurement methods: *1. I/Q samples*: BigSpec can potentially support I/Q samples in addition to energy readings given the fact that SVD can be generalized to complex numbers. Although the current Spark implementation does not support computing SVD for a complex matrix, it is possible to achieve this when only real SVD is available by using its equivalent real matrix [23].

2. Multiple sensing devices and crowdsourcing: If multiple devices all have the same frequency resolution, and measure the same bands with identical bandwidth, we can combine their data. In fact, as long as we fix the number of frequency bins per measurement and the start and end frequency for each band, we can tolerate different bandwidths for each band (and consequently different frequency resolution). If multiple devices have significantly different performance (in terms of resolution, bandwidth), we envision a solution where low resolution devices can detect any anomaly based on the data gathered by high resolution devices and ask the nearest user with a high resolution device to verify this anomaly.

Generalizing to other apps: There are other apps that may be of interest. For example, can BigSpec identify the signal pattern, feature, modulation, and technology of different types of transmitter? What portion of the users in a shared spectrum is primary/secondary users respectively? Generalizing BigSpec to other apps needs to design algorithms of app specific modules. However, we believe an efficient and scalable app specific module that analyzes spectrum utilization over the entire spatio-temporal space rather than within a short time window should always perform computations on compressed data that retain signal features. As shown previously, lossless compression is unhelpful and channel allocation based compression offers coarse information only. Thus, we believe our app-agnostic preprocessing modules are a good example of the entire class of algorithms that reduce the number of dimensions but still preserve (almost) all useful features, so that a balance between high compression ratio and easiness to extract fine-grained information is achieved. Furthermore, if more preprocessing modules are going to be added to BigSpec, we believe they should have the same idea as ours to reach this balance. This is the reason why we think BigSpec is generalizable, and we hope it can form a new spectrum (batch) data analysis paradigm, similar to how classic MapReduce paradigm used to shape the way people do computations on big data.

9 RELATED WORK

Spectrum measurement and spectrum observatory: Previous spectrum measurement efforts either fix the locations and only record the temporal variations, e.g. [6, 28, 57], or record spatial discrepancies and assume time-invariance, e.g. [45, 46, 61]. Our effort, however, assumes little prior knowledge and records both spatial and temporal variations. Recent research efforts also include low cost sensing devices [36, 37, 49, 62] and quick sensing methods [25, 26, 60]. Although some work has been done on indoor measurement [20, 59], outdoor measurement gains more attention and wideband long-term outdoor efforts have led to the work of building spectrum observatories [40, 50, 53, 66]. [50] and [66] also offer solution to analyze signal patterns and to detect transmitters respectively from MSO's data. Compared with these works, we provide a general-purpose framework for efficiently doing spectrum data analysis of massive amount of data, whose key idea is to perform computations on compressed data that retain signal features, and we illustrate this point with three example apps.

Signal detection from spectrum measurement: Signal detection from a single spectrum measurement has been extensively studied. Classical methods can usually be categorized as energy detection or feature detection [31, 56]. [52] gives a survey. Recent work [63] also detects signals in transformed space but it still focuses on single/small spatio-temporal scale measurement(s). We, however, focus on how to directly detect energy of signals from a large number/spatio-temporal scale of spectrum measurements, a direction which has attracted little attention in previous work.

Spectrum estimation: There are two main directions in spectrum estimation, i.e. channel energy/occupancy estimation [4, 35, 57] and transmitter type/location estimation [27, 30, 42, 64]. State-of-the-art method for dealing with time invariant channel energy estimation is Kriging [14, 39, 58]. Our method, however, is different in two respects. First, we consider the spatial and temporal domains together instead of separately. Second, we provide estimated frequency bin level energy rather than channel level energy/occupancy. Moreover, although we do not address transmitter type estimation in this paper, an efficient module following the same key idea can always be added to BigSpec to solve it.

Spectrum anomaly detection: Spectrum anomaly detection has been well studied in the context of cooperative sensing [15, 24, 29, 34, 54]. However, they are all based on the sum power readings of particular channels, hence cannot distinguish between frequency domain anomaly and spatio-temporal domain anomaly. Our method, however, can distinguish these two different anomalies and we have shown the benefits. Recent work SAIFE [41] also can detect anomalies from high dimensional PSD or I/Q data, but it is based on L1 norm of reconstruction error and works on data with temporal variations only.

Spectrum data compression: Airpress [65] also noted the scalability issue of spectrum inventory. Thus, it mainly focuses on how to minimize the size of data with maximal compression ratio 64. We take a step further and consider data compression as a preprocessing step to transfer the data into a less complex space with signal features retained, so that we can enable different apps efficiently.

10 CONCLUSIONS

We have presented BigSpec, a general-purpose framework that can enable different spectrum related apps efficiently on large volume of spectrum data. Although we only evaluate the performance of BigSpec using three example apps in this paper, we believe that the key idea of BigSpec enables us to gain a deeper understanding of spectrum utilization in large spatio-temporal scales with little prior knowledge. We envision BigSpec to be extended with other building blocks to enable more interesting apps by the community in the future. We foresee that the new insights generated using BigSpec are of considerable value in assisting users, service providers, and regulation authorities to better measure and utilize spectrum.

ACKNOWLEDGEMENT

We thank our shepherd Ashutosh Sabharwal and the anonymous reviewers for their detailed feedback. We are grateful to Madison Metro bus for letting us collecting data, and Steve Bauder at Wisconsin Public Broadcasting for answering our questions about TV spectrum. We appreciate our lab mates for setting up the data collection platform, Jerry Zhu for useful discussions at the early stage of this project, and Robin Corcos for proofreading early versions of this paper. Yijing Zeng, Varun Chandrasekaran, and Suman Banerjee were supported in part by US National Science Foundation grant CNS-1838733, CNS-1719336, CNS-1647152, and CNS-1629833. Domenico Giustiniano was sponsored in part by the NATO Science for Peace and Security Programme under grant G5461, and Madrid Regional Government through TAPIR-CM project S2018/TCS-4496.

REFERENCES

[1] Cloudlab. https://www.cloudlab.us/.

[2] Fcc: General freeze of new applications on channel 51. http://hraunfoss.fcc.gov/edocs_public/attachmatch/DA-11-1428A1.pdf.

[3] Fcc online table of frequency allocations. https://www.ntia.doc.gov/files/ntia/publications/january_2016_spectrum_wall_chart.pdf.

[4] Google spectrum database. https://www.google.com/get/spectrumdatabase/.

[5] Gzip. https://www.gzip.org.

[6] Microsoft spectrum observatory. https://observatory.microsoftspectrum.com/.

[7] Shared spectrum company. http://www.sharedspectrum.com/.

[8] Spark. https://spark.apache.org/.

[9] Spectrum crunch. https://obamawhitehouse.archives.gov/sites/default/files/docs/20150122_spectrum_auction_wsj.pdf.

[10] Spectrumwiki. http://www.spectrumwiki.com/.

[11] Thinkrf. http://www.thinkrf.com/.

[12] True colors pride celebration week at five. http://ourlivesmadison.com/event/true-colors-pride-celebration-week-at-five/.

[13] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[14] A. Achtzehn, J. Riihijarvi, and P. Mahonen. Improving accuracy for tvws geolocation databases: Results from measurement-driven estimation approaches. In *Dynamic Spectrum Access Networks (DYSPAN), 2014 IEEE International Symposium on*, pages 392–403. IEEE, 2014.

[15] T. Bansal, B. Chen, and P. Sinha. Fastprobe: Malicious user detection in cognitive radio networks through active transmissions. In *INFOCOM, 2014 Proceedings IEEE*, pages 2517–2525. IEEE, 2014.

[16] A. J. Bell and T. J. Sejnowski. The "independent components" of natural scenes are edge filters. *Vision research*, 37(23):3327–3338, 1997.

[17] C. M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

[18] Z.-Y. Chai, Y.-L. Li, Y.-M. Han, and S.-F. Zhu. Recommendation system based on singular value decomposition and multi-objective immune optimization. *IEEE Access*, 7:6060–6071, 2019.

[19] A. Chakraborty, A. Bhattacharya, S. Kamal, S. R. Das, H. Gupta, and P. M. Djuric. Spectrum patrolling with crowdsourced spectrum sensors. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 1682–1690. IEEE, 2018.

[20] A. Chakraborty and S. R. Das. Designing a cloud-based infrastructure for spectrum sensing: A case study for indoor spaces. In *Distributed Computing in Sensor Systems (DCOSS), 2016 International Conference on*, pages 17–24. IEEE, 2016.

[21] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[22] Cloudera. Cdh5. https://archive.cloudera.com/cm5/installer/latest/.

[23] D. Day and M. A. Heroux. Solving complex-valued linear systems via equivalent real formulations. *SIAM Journal on Scientific Computing*, 23(2):480–498, 2001.

[24] O. Fatemieh, R. Chandra, and C. A. Gunter. Secure collaborative sensing for crowd sourcing spectrum data in white space networks. In *New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on*, pages 1–12. IEEE, 2010.

[25] Y. Guddeti, R. Subbaraman, M. Khazraee, A. Schulman, and D. Bharadia. Sweepsense: Sensing 5 ghz in 5 milliseconds with low-cost radios. In *NSDI*, pages 317–330, 2019.

[26] H. Hassanieh, L. Shi, O. Abari, E. Hamed, and D. Katabi. Ghz-wide sensing and decoding using the sparse fourier transform. In *INFOCOM, 2014 Proceedings IEEE*, pages 2256–2264. IEEE, 2014.

[27] S. S. Hong and S. R. Katti. Dof: a local wireless information plane. In *ACM SIGCOMM Computer Communication Review*, volume 41, pages 230–241. ACM, 2011.

[28] M. Hoyhtya, M. Matinmikko, X. Chen, J. Hallio, J. Auranen, R. Ekman, J. Roning, J. Engelberg, J. Kalliovaara, T. Taher, et al. Measurements and analysis of spectrum occupancy in the 2.3–2.4 ghz band in finland and chicago. In *Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM), 2014 9th International Conference on*, pages 95–101. IEEE, 2014.

[29] P. Kaligineedi, M. Khabbazian, and V. K. Bhargava. Malicious user detection in a cognitive radio cooperative sensing system. *IEEE Transactions on Wireless Communications*, 9(8):2488–2497, 2010.

[30] M. Khaledi, M. Khaledi, S. Sarkar, S. Kasera, N. Patwari, K. Derr, and S. Ramirez. Simultaneous power-based localization of transmitters for crowdsourced spectrum monitoring. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 235–247. ACM, 2017.

[31] H. Kim and K. G. Shin. In-band spectrum sensing in cognitive radio networks: energy detection or feature detection? In *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 14–25. ACM, 2008.

[32] F. Liu, X. Cheng, and D. Chen. Insider attacker detection in wireless sensor networks. In *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*, pages 1937–1945. IEEE, 2007.

[33] L. Liu, H. Li, and Z. Han. Sampling spectrum occupancy data over random fields: A matrix completion approach. In *2012 IEEE International Conference on Communications (ICC)*, pages 1487–1491. IEEE, 2012.

[34] S. Liu, Y. Chen, W. Trappe, and L. J. Greenstein. Aldo: An anomaly detection framework for dynamic spectrum access networks. In *INFOCOM 2009, IEEE*, pages 675–683. IEEE, 2009.

[35] R. Murty, R. Chandra, T. Moscibroda, and P. Bahl. Senseless: A database-driven white spaces network. *IEEE Transactions on Mobile Computing*, 11(2):189–203, 2012.

[36] A. Nika, Z. Li, Y. Zhu, Y. Zhu, B. Y. Zhao, X. Zhou, and H. Zheng. Empirical validation of commodity spectrum monitoring. In *Proceedings of 14th ACM Conference on Embedded Networked Sensor Systems (SenSys 2016). ACM*, 2016.

[37] A. Nika, Z. Zhang, X. Zhou, B. Y. Zhao, and H. Zheng. Towards commoditized real-time spectrum monitoring. In *Proceedings of the 1st ACM workshop on Hot topics in wireless*, pages 25–30. ACM, 2014.

[38] D. Pfammatter, D. Giustiniano, and V. Lenders. A software-defined sensor architecture for large-scale wideband spectrum monitoring. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, pages 71–82. ACM, 2015.

[39] C. Phillips, M. Ton, D. Sicker, and D. Grunwald. Practical radio environment mapping with geostatistics. In *Dynamic Spectrum Access Networks (DYSPAN), 2012 IEEE International Symposium on*, pages 422–433. IEEE, 2012.

[40] S. Rajendran, R. Calvo-Palomino, M. Fuchs, B. Van den Bergh, H. Cordobés, D. Giustiniano, S. Pollin, and V. Lenders. Electrosense: Open and big spectrum data. *IEEE Communications Magazine*, 56(1):210–217, 2018.

[41] S. Rajendran, W. Meert, V. Lenders, and S. Pollin. Saife: Unsupervised wireless spectrum anomaly detection with interpretable features. In *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pages 1–9. IEEE, 2018.

[42] S. Rayanchu, A. Patro, and S. Banerjee. Airshark: detecting non-wifi rf devices using commodity wifi hardware. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 137–154. ACM, 2011.

[43] N. M. Razali, Y. B. Wah, et al. Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. *Journal of*

statistical modeling and analytics, 2(1):21–33, 2011.

[44] R. Ricci, E. Eide, and C. Team. Introducing cloudlab: Scientific infrastructure for advancing cloud architectures and applications. *; login:: the magazine of USENIX & SAGE*, 39(6):36–38, 2014.

[45] A. Saeed, K. A. Harras, E. Zegura, and M. Ammar. Local and low-cost white space detection. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 503–516. IEEE, 2017.

[46] A. Saifullah, M. Rahman, D. Ismail, C. Lu, R. Chandra, and J. Liu. Snow: Sensor network over white spaces. In *Proceedings of the International Conference on Embedded Networked Sensor Systems (ACM SenSys)*, 2016.

[47] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Application of dimensionality reduction in recommender system-a case study. Technical report, Minnesota Univ Minneapolis Dept of Computer Science, 2000.

[48] S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591–611, 1965.

[49] J. Shi, Z. Guan, C. Qiao, T. Melodia, D. Koutsonikolas, and G. Challen. Crowdsourcing access network spectrum allocation using smartphones. In *Proceedings of the 13th ACM workshop on hot topics in networks*, page 17. ACM, 2014.

[50] L. Shi, P. Bahl, and D. Katabi. Beyond sensing: Multi-ghz realtime spectrum analytics. In *NSDI*, pages 159–172, 2015.

[51] K. Shvachko, H. Kuang, S. Radia, and R. Chansler. The hadoop distributed file system. In *2010 IEEE 26th symposium on mass storage systems and technologies (MSST)*, pages 1–10. IEEE, 2010.

[52] M. Subhedar and G. Birajdar. Spectrum sensing techniques in cognitive radio networks: A survey. *International Journal of Next-Generation Networks*, 3(2):37–51, 2011.

[53] T. M. Taher, R. B. Bacchus, K. J. Zdunek, and D. A. Roberson. Long-term spectral occupancy findings in chicago. In *New Frontiers in Dynamic Spectrum Access Networks (DySPAN), 2011 IEEE Symposium on*, pages 100–107. IEEE, 2011.

[54] W. Wang, L. Chen, K. G. Shin, and L. Duan. Secure cooperative spectrum sensing and access against intelligent malicious behaviors. In *INFOCOM, 2014 Proceedings IEEE*, pages 1267–1275. IEEE, 2014.

[55] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pages 802–810, 2015.

[56] L. Yang, W. Hou, L. Cao, B. Y. Zhao, and H. Zheng. Supporting demanding wireless applications with frequency-agile radios. In *NSDI*, pages 65–80, 2010.

[57] S. Yin, D. Chen, Q. Zhang, M. Liu, and S. Li. Mining spectrum usage data: a large-scale spectrum measurement study. *IEEE Transactions on Mobile Computing*, 11(6):1033–1046, 2012.

[58] X. Ying, C. W. Kim, and S. Roy. Revisiting tv coverage estimation with measurement-based statistical interpolation. In *Communication Systems and Networks (COMSNETS), 2015 7th International Conference on*, pages 1–8. IEEE, 2015.

[59] X. Ying, J. Zhang, L. Yan, G. Zhang, M. Chen, and R. Chandra. Exploring indoor white spaces in metropolises. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 255–266. ACM, 2013.

[60] S. Yoon, L. E. Li, S. C. Liew, R. R. Choudhury, I. Rhee, and K. Tan. Quicksense: Fast and energy-efficient channel sensing for dynamic spectrum access networks. In *INFOCOM, 2013 Proceedings IEEE*, pages 2247–2255. IEEE, 2013.

[61] T. Zhang, N. Leng, and S. Banerjee. A vehicle-based measurement framework for enhancing whitespace spectrum databases. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 17–28. ACM, 2014.

[62] T. Zhang, A. Patro, N. Leng, and S. Banerjee. A wireless spectrum analyzer in your pocket. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pages 69–74. ACM, 2015.

[63] M. Zheleva, P. Bogdanov, T. Larock, and P. Schmitt. Airview: Unsupervised transmitter detection for next generation spectrum sensing. In *IEEE International Conference on Computer Communications (INFOCOM2018)*, 2018.

[64] M. Zheleva, R. Chandra, A. Chowdhery, A. Kapoor, and P. Garnett. Txminer: Identifying transmitters in real-world spectrum measurements. In *Dynamic Spectrum Access Networks (DySPAN), 2015 IEEE International Symposium on*, pages 94–105. IEEE, 2015.

[65] M. Zheleva, T. Larock, P. Schmitt, and P. Bogdanov. Airpress: High-accuracy spectrum summarization using compressed scans. In *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pages 1–5. IEEE, 2018.

[66] 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*, 17(2):362–375, 2018.