

# Selection of Sensors for Efficient Transmitter Localization

Arani Bhattacharya<sup>1</sup>, Member, IEEE, Caitao Zhan, Graduate Student Member, IEEE,

Abhishek Maji, Himanshu Gupta, Member, IEEE, Samir R. Das<sup>2</sup>, Member, IEEE,

and Petar M. Djurić<sup>3</sup>, Fellow, IEEE

**Abstract**—We address the problem of localizing an (unauthorized) transmitter using a distributed set of sensors. Our focus is on developing techniques that perform the transmitter localization in an efficient manner, wherein the efficiency is defined in terms of the number of sensors used to localize. Localization of unauthorized transmitters is an important problem which arises in many important applications, e.g., in patrolling of shared spectrum systems for any unauthorized users. Localization of transmitters is generally done based on observations from a deployed set of sensors with limited resources, thus it is imperative to design techniques that minimize the sensors' energy resources. In this paper, we design greedy approximation algorithms for the optimization problem of selecting a given number of sensors in order to maximize an appropriately defined objective function of localization accuracy. The obvious greedy algorithm delivers a constant-factor approximation only for the special case of two hypotheses (potential locations). For the general case of multiple hypotheses, we design a greedy algorithm based on an appropriate auxiliary objective function—and show that it delivers a provably approximate solution for the general case. We develop techniques to significantly reduce the time complexity of the designed algorithms by incorporating certain observations and reasonable assumptions. We evaluate our techniques over multiple simulation platforms, including an indoor as well as an outdoor testbed, and demonstrate the effectiveness of our designed techniques—our techniques easily outperform prior and other approaches by up to 50-60% in large-scale simulations and up to 16% in small-scale testbeds.

**Index Terms**—Approximation algorithms, cognitive radio, energy efficiency, radio spectrum management.

## I. INTRODUCTION

**W**IRELESS transmitter localization via analysis of the received signal from multiple receivers or sensors is an important problem. While the problem has been widely explored, it exposes new challenges in many emerging applications due to the constraints of the application. In this

Manuscript received March 31, 2020; revised February 12, 2021 and June 8, 2021; accepted July 30, 2021; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor Y. Chen. This work was supported in part by the NSF Grant CNS-1642965. (Corresponding author: Arani Bhattacharya.)

Arani Bhattacharya is with the IIIT-Delhi, New Delhi 110020, India (e-mail: arani@iiitd.ac.in).

Caitao Zhan, Himanshu Gupta, and Samir R. Das are with the Department of Computer Science, Stony Brook University, Stony Brook, NY 11794 USA (e-mail: cbzhan@cs.stonybrook.edu; hgupta@cs.stonybrook.edu; samir@cs.stonybrook.edu).

Abhishek Maji is with Hitachi ABB Power Grids, Uno Lamm HVDC Center, Dalarnas Lan, 771 80 Ludvika, Sweden (e-mail: amaji@kth.se).

Petar M. Djurić is with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794 USA (e-mail: petar.djuric@stonybrook.edu).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TNET.2021.3104000>, provided by the authors.

Digital Object Identifier 10.1109/TNET.2021.3104000

work, we are specifically interested in a distributed monitoring system where a set of distributed RF sensors are tasked to detect and localize transmitters. These transmitters could be of various type. For example, in certain spectrum allocation scenarios, unknown primary transmitters need to be detected/localized, or in spectrum patrolling scenarios, unauthorized transmitters need to be detected/localized [1]. Recent work has explored new approaches for such monitoring where the RF sensors are crowdsourced, perhaps using various low-cost spectrum sensing platforms [2], [3]. The crowdsourcing deploys a large number of sensors. Fine grained spectrum sensing is implemented by creating suitable incentive mechanisms [2], [4].

Crowdsourcing makes the sensing *cost-conscious*. The cost here could be incentivization cost, cost of power, backhaul bandwidth on the part of the spectrum owner or the opportunity cost – being low-cost platform, the sensors may be able to only sense smaller spectrum bands at a time. Thus, involving only a small number of sensors or sensors with low overall cost budget (for a suitable cost model) for sufficiently accurate localization performance is critical. Prior work that discusses sensor selection in this context only presents heuristics without any performance guarantees [2].

We do not use geometric approaches which rely on hard-to-model mapping of received power to distance. Instead, we use a hypothesis-driven, Bayesian approach for localization [5]. We focus on the optimization problem of selecting a certain number of sensors from among the deployed sensors such that an appropriately defined objective of localization accuracy is maximized. This optimization problem can also be used to solve the dual problem of selecting a minimum number of sensors (or sensors with the minimum total cost budget) to ensure at least a given localization accuracy. We adopt the framework of a hypothesis-driven localization approach wherein each hypothesis represents a configuration (location, power, etc.) of the potential transmitters and then the localization is equivalent to determining the most-likely prevailing hypothesis. See Figure 1. The hypothesis-driven framework does not require an assumption of a propagation model, and works for arbitrary signal propagation characteristics. The framework does, however, require prior training to build joint probability distributions of observation vectors for each hypothesis.

### A. Our Contributions

In the above hypothesis-based framework, we develop an overall approach that enables selection of sensors that are

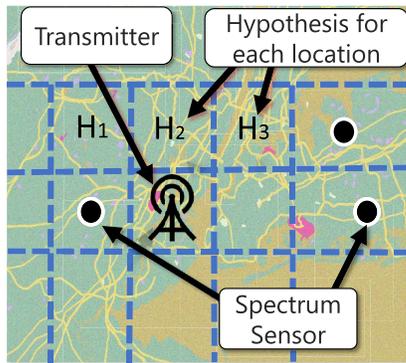


Fig. 1. Hypothesis-driven localization. The figure shows the simple case of localizing a single transmitter with fixed power; thus, there is a hypothesis created for each potential location. Observations from deployed sensors are analyzed to determine the most likely prevailing hypothesis (and thus, location).

most relevant to localize transmitters. In particular, we develop algorithms that aim to maximize localization accuracy for a given budget of number of sensors to be used for localization. More specifically, we make the following contributions in the paper.

- 1) We design a greedy algorithm (GA) that selects sensors iteratively to maximize the objective function of localization accuracy, under the constraint of number of sensors selected. We prove that GA yields a constant-factor approximate solution for the special case of the problem wherein there are only two hypotheses.
- 2) For the general case of more than two hypotheses, we design an alternate greedy scheme (called AGA) based on maximizing an auxiliary objective function. We prove that AGA delivers a solution that has (i) an auxiliary objective value within a constant factor of the optimal auxiliary objective value, as well as (ii) a localization error within a certain factor of the optimal localization error.
- 3) We optimize the time complexity of our developed algorithms by a substantial factor, based on certain observations and reasonable assumptions. In addition, we generalize our techniques to more practical and useful settings.
- 4) We evaluate the performance of the developed algorithms over multiple evaluation platforms: (1) large-scale simulation using synthetically generated data using established signal propagation models (with 100 sensors and 1600 hypothesis with each hypothesis of area  $100\text{ m} \times 100\text{ m}$ ), and (2) publicly available experimental data trace collected over an indoor WiFi network with 44 sensors (with 43 sensors and 44 hypotheses, with each hypothesis of area  $1\text{ m} \times 1\text{ m}$ ), and (3) our own data collection using 18 outdoor software radio sensors in the 915 MHz band with a custom transmitter (with 18 sensors and 100 hypotheses, with each hypothesis of area  $3.2\text{ m} \times 3.2\text{ m}$ ). In each of these cases, the sensors collect RSSI data for each location of the transmitter. Results show that our techniques outperform other state-of-the-art algorithm [2] up to a factor of 50-60% in the

large-scale simulation, and up to a factor of 16% on the indoor WiFi network and our own outdoor network.

A preliminary version of this paper has been accepted for publication at IEEE Infocom 2020 [6]. This version of the paper describes additional results about the performance of the algorithm, and it provides more details about the experiments. It also contains proofs of multiple lemmas and theorems that had been omitted from the preliminary version.

## II. BACKGROUND AND MOTIVATION

### A. Problem Setting

The overall setting of the transmitter localization problem is as follows. Consider a geographic area, with a number of spectrum sensors deployed or available (if attached to mobile devices) at known locations. At any instant, one or more transmitters are allowed to transmit signals (on a common frequency). Each deployed/available spectrum sensor senses and processes the aggregate received signal, and reports appropriate metric (i.e., total received power or signal strength) to a central server which estimates the location of the transmitter(s) using the maximum-likelihood hypothesis algorithm as described below. The overall objective of our paper is to develop techniques to select an optimal subset of sensors in order to accurately localize any present transmitters. Though our developed techniques naturally extend to the case of multiple transmitters, for simplicity, we assume at most a single transmitter present at any instant. We start with defining basic notations used throughout the paper.

### B. Hypotheses, Observations, and Inputs

We discretize the given space into locations  $l_1, l_2, \dots$ , and transmit power of a potential transmitter is similarly discretized into levels  $p_1, p_2, \dots$ . We represent potential “configurations” of the possible transmitter by hypotheses  $H_0, H_1, \dots, H_m$ , where each hypothesis  $H_i$  represents a configuration  $(l_i, p_i)$  of location  $l_i$  and transmit power  $p_i$  of a potential transmitter (see Figure 1). We use the convention that hypothesis  $H_0$  corresponds to no transmitter being present. Localizing any potential transmitter is thus equivalent to determining the prevailing hypothesis. To do this, we use observations from a set of deployed sensors. We denote the observation vector of a subset of sensors  $\mathbf{T}$  by  $\mathbf{x}_{\mathbf{T}}$  (we usually drop the subscript  $\mathbf{T}$ , as it is clear from the context). In our setting, a sensor observation can be any type of reading that may be indicative of the transmitter’s location. In this work, we focus on RSSI, as this is a very common measure and this also allows for direct comparison with other prior works. In principle, any other parameter, such as ToA or AoA, is possible but not very relevant in a crowdsourced setting as they typically need more complex hardware to measure.

**Inputs.** For a given set of sensors deployed over an area, we assume the following available inputs, obtained via a priori training, data gathering and/or analysis<sup>1</sup>:

<sup>1</sup>In our prior work [7], we discuss novel interpolation techniques to minimize such training cost.

- Prior probabilities of the hypotheses, i.e.  $P(H_i)$ , for each hypothesis  $H_i$ . Since we do not assume any propagation model, the probabilities of hypotheses at adjacent locations may arbitrarily vary. Our technique, therefore, can naturally model the presence of radio obstructions, such as buildings, terrain and vegetation.
- Joint probability distribution (JPD) of sensors' observations for each hypothesis. More formally, for each hypothesis  $H_j$ , we assume  $P(\mathbf{x}_S|H_j)$  to be known for each observation  $\mathbf{x}_S$  for the entire set  $\mathbf{S}$  of deployed sensor. Note that this also gives us the JPD's of each subset  $\mathbf{T} \subseteq \mathbf{S}$ .

### C. Maximum a Posteriori Localization (MAP) Algorithm

We use Bayes' rule to compute the likelihood probability of each hypothesis, from a given observation vector  $\mathbf{x}_\mathbf{T}$  for a subset of sensors  $\mathbf{T}$ :

$$P(H_i|\mathbf{x}_\mathbf{T}) = \frac{P(\mathbf{x}_\mathbf{T}|H_i)P(H_i)}{\sum_{j=0}^m P(\mathbf{x}_\mathbf{T}|H_j)P(H_j)} \quad (1)$$

We select the hypothesis that has the highest probability, for given observations of a set of sensors. Formally, the MAP algorithm returns the hypotheses based on the following equation:

$$\arg \max_{i=0}^m P(H_i|\mathbf{x}_\mathbf{T}) \quad (2)$$

The above MAP algorithm to determine the prevailing hypothesis is known to be *optimal* [8], i.e., it yields minimum probability of (misclassification) error under the zero-one cost function. The above hypothesis-based approach to localization works for arbitrary signal propagation characteristics, and in particular, obviates the need to assume a propagation model. However, it does incur a one-time training cost to obtain the JPDs, which can be optimized via independent techniques [9]. The above approach based on fingerprints has already been used for localization [10].

### D. Selection of Sensors for Localization

As mentioned above, in a typical setting, spectrum sensors may be deployed at pre-determined locations or available at certain locations (if part of mobile devices) to sense unauthorized signals and thus localize any unauthorized transmitters. Two immediate problems of interest in this context are: where to deploy given a number of sensors, and once deployed/available, which subset of sensors to select for localization. The latter problem of selection of sensors is motivated by the fact that, in most realistic settings, the sensors (or their mobile devices) are not tethered to AC power outlets and hence **have limited energy resources**. Moreover, spectrum sensors **also incur cost** in transmitting sensing data to the fusion/cloud center [11]. Thus, it is critical to optimize resources and costs incurred in localization of unauthorized transmitters, e.g., via the selection of an optimal set of sensors. Note that the sensor-selection problem can also be used to effectively *deploy* a given number of sensors, by assuming sensors available at all potential locations.

## III. OPTIMAL SENSOR SELECTION FOR INTRUDER LOCALIZATION

In this section, we address the problem of sensor selection for transmitter localization; informally, the problem is to select an optimal set of  $B$  sensors such that the overall probability of error of localizing a transmitter is minimized, given appropriate JPDs as discussed in the previous section. We start with formulating the problem in the following subsection. In following subsection, we present a greedy algorithm for it and prove that it is guaranteed to deliver an approximation solution for the special case of two hypotheses. However, as shown, the greedy algorithm can perform arbitrarily bad for the general case of multiple hypotheses. Thus, we then modify our algorithm to use an "auxiliary" objective function and show that the modified algorithm delivers an approximation solution for the general case of multiple hypotheses albeit with a slightly worse approximation ratio. Finally, we discuss optimizing the computation complexity of the designed algorithms, certain extensions and other issues.

### A. LSS Problem Formulation

We start with formally defining the optimization objective (probability of error or misclassification) for a given subset of sensors. Then, we formally define the sensor selection problem, hereto referred to as *Localization Sensor Selection (LSS)* problem. Throughout this section, we use hypotheses  $H_0$  to represent the hypotheses with no transmitters present, and  $H_i$  to represent the hypotheses wherein a transmitter is present in  $i^{\text{th}}$  configuration.

1) *Probability of Error* ( $P_{\text{err}}(\mathbf{T})$ ): Recall that, for a given observation vector, the MAP localization algorithm outputs the hypothesis that has the most likelihood among the given hypotheses. Thus, MAP can also be looked upon as a classification technique. Given a subset of sensors  $\mathbf{T}$ , we define the *probability of error* or misclassification as the probability of the MAP algorithm outputting a hypothesis different from the actual *ground truth* (i.e., prevailing hypothesis). The expected or overall probability of error is an expectation of the probability of error over all possible prevailing hypotheses and/or observation vectors  $\mathbf{x}_\mathbf{T}$  from  $\mathbf{T}$ . Our techniques generalize to the notion of distance-based localization error, as discussed in §III-G.

Formally, let  $\text{MAP}(\mathbf{x})$  be the output of the MAP algorithm on observation vector  $\mathbf{x}$  from a given subset of sensors  $\mathbf{T}$ . Given  $H_i$  as the ground truth and  $\mathbf{x}$  as the observation vector, the probability of error  $P_{\text{err}}(\mathbf{T}|H_i, \mathbf{x})$  can be written as:

$$P_{\text{err}}(\mathbf{T}|H_i, \mathbf{x}) = \mathbf{1}[\text{MAP}(\mathbf{x}) \neq i|H_i], \quad (3)$$

where  $\mathbf{1}$  is an indicator function which is equal to 1 if the predicate is true, and 0 otherwise. Since expectation over the data point of an indicator function is its probability, we take the expectation over  $\mathbf{x}$  on both sides to get:

$$P_{\text{err}}(\mathbf{T}|H_i) = P(\text{MAP}(\mathbf{x}) \neq i|H_i) \quad (4)$$

Above, the probability is over the random variable  $\mathbf{x}$ . Now, if the ground truth hypothesis is also not given, we can compute an expectation over all possible hypotheses. Thus,

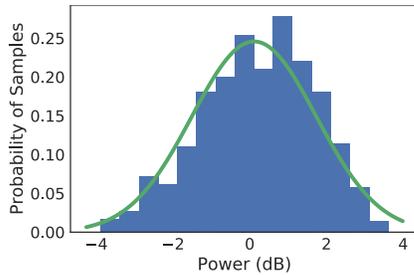


Fig. 2. Distribution of the received power from a transmitter at an RTL-SDR sensor, and the Gaussian fit (green line) of the observed distribution. The transmitter and the sensor are kept in the corridor of a large building at the same height, 10m apart from each other.

the (overall) *probability of error* for a given set of sensors  $\mathbf{T}$  is given by:

$$P_{\text{err}}(\mathbf{T}) = \sum_{i=0}^{m-1} P(\text{MAP}(\mathbf{x}) \neq i | H_i) P(H_i) \quad (5)$$

2) *Localization Accuracy Function*,  $O_{\text{acc}}(\mathbf{T})$ : To facilitate a greedy approximation solution, we formulate our sensor selection as a maximization problem—and thus, define a corresponding maximization objective. In particular, we define the *localization accuracy*  $O_{\text{acc}}(\mathbf{T})$  as  $1 - P_{\text{err}}(\mathbf{T})$ . Based on the above equation Eqn. 5, we get the expression for  $O_{\text{acc}}(\mathbf{T})$  as:

$$O_{\text{acc}}(\mathbf{T}) = 1 - P_{\text{err}}(\mathbf{T}) = \sum_{i=0}^{m-1} P(\text{MAP}(\mathbf{x}) = i | H_i) P(H_i) \quad (6)$$

3) *Localization Sensor Selection (LSS) Problem*: Consider a geographic area with a set of sensors  $\mathbf{S}$  deployed. Given a set of hypotheses and JPD's, as defined in previous section, the OSS problem is to select a subset  $\mathbf{T} \subseteq \mathbf{S}$  of sensors with minimum probability of error  $P_{\text{err}}(\mathbf{T})$  (or maximum localization accuracy  $O_{\text{acc}}(\mathbf{T})$ ), under the constraint that  $|\mathbf{T}|$  is at most a given budget  $B$ . Formally, the formulation is:

$$\text{Maximize } O_{\text{acc}}(\mathbf{T}) \text{ subject to } |\mathbf{T}| \leq B. \quad (7)$$

The above formulation implicitly assumes a uniform cost for each sensor; we generalize our techniques to handle non-uniform sensor costs (see §III-G).

We show that the above LSS problem is NP-hard, via reduction from the well-known maximum-coverage problem (Appendix A). Thus, we develop approximation algorithms below; in particular, our focus is on developing greedy approximation algorithms. The key challenge lies in showing that the objective function satisfies certain desired properties that ensure the approximability of the algorithm.

### B. Transmitter and Sensor Model

We now formally define the assumptions that would allow us to ensure the approximability of our algorithm. First, we assume that the joint probability distribution (JPD) follow a joint Gaussian distribution with the means  $(\mathbf{p}_i, \Sigma)$  for all hypotheses  $H_i$ ,  $\forall i = 0, \dots, m-1$ . We empirically verify

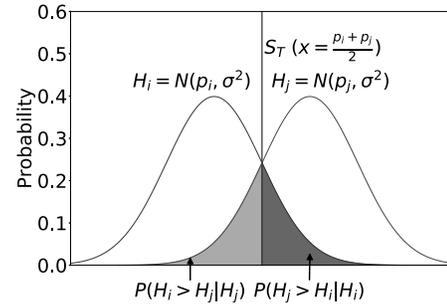


Fig. 3. Classification of a data point between two Gaussians using a threshold.

these assumptions using our own sensor in the wild (as shown in Figure 2). These assumptions have also been made by multiple prior studies [12], [13]. The covariance matrix remains same across hypotheses, since the correlation and noise are properties of the sensors. The means  $\mathbf{p}_i$  can be different, as different power values are received by the sensors depending on the location of the transmitter.

### C. Properties of MAP Algorithm

To explain our sensor selection algorithm, we first need to explain a few properties of MAP algorithm. Assume that there are two hypotheses  $H_i$  and  $H_j$ , with distributions  $(p_i, \sigma^2)$  and  $(p_j, \sigma^2)$  as well as priors  $P(H_i)$  and  $P(H_j)$  respectively, where  $p_i, p_j \in \mathbb{R}$ . Without loss of generality, we assume that  $p_i < p_j$ . In this case, given a data point  $X$ , the MAP algorithm works by comparing it with a fixed threshold  $S_T$  (shown in Figure 3). If  $X \leq S_T$ , then MAP classifies  $X$  as  $H_i$ , i.e.  $\text{MAP}(X) = i$ , otherwise it classifies  $X$  as  $H_j$ , i.e.  $\text{MAP}(X) = j$ . Note that because this is a stochastic decision, there will always be some probability of classification error, depending on the value of  $S_T$ . The MAP algorithm uses the threshold value of  $S_T = \frac{p_i + p_j}{2}$ , and it is well-known that this value of  $S_T$  provides the lowest probability of classification error. Formally, we write this as:

$$X \underset{H_i}{\overset{H_j}{\geq}} \frac{p_i + p_j}{2} + \log \frac{P(H_i)}{P(H_j)} \quad (8)$$

We now explain the case for multidimensional distributions, where  $H_i$  and  $H_j$  are given by  $(\mathbf{p}_i, \Sigma)$  and  $(\mathbf{p}_j, \Sigma)$  respectively ( $\mathbf{p}_i, \mathbf{p}_j \in \mathbb{R}, \Sigma \in \mathbb{R} \times \mathbb{R}$ ). In this case, the classification of a given data vector can be done by comparing with a hyperplane. However, this problem of classification between distributions with multiple dimensions can be *reduced* to classification between distributions with single dimensions, using the following theorem:

*Theorem 1: Given the hypotheses  $H_i \sim N(\mathbf{p}_i, \Sigma)$  and  $H_j \sim N(\mathbf{p}_j, \Sigma)$ , a data vector  $\mathbf{x} = [x_1 \dots x_n]$  can be classified by applying the following threshold test:*

$$\mathbf{x}^T \Sigma^{-1} (\mathbf{p}_j - \mathbf{p}_i) \underset{H_i}{\geq} \frac{1}{2} (\mathbf{p}_i + \mathbf{p}_j)^T \Sigma^{-1} (\mathbf{p}_j - \mathbf{p}_i) + \log \frac{P(H_i)}{P(H_j)} \quad (9)$$

We prove this in Appendix B. We call the LHS of Eqn (9) as test statistic  $T(\mathbf{x})$ . We also show as a corollary of the

theorem that the test statistic itself follows a Gaussian distribution of  $N(\mathbf{p}_i \Sigma^{-1}(\mathbf{p}_j - \mathbf{p}_i), (\mathbf{p}_j - \mathbf{p}_i)^T \Sigma^{-1}(\mathbf{p}_j - \mathbf{p}_i))$  and  $N(\mathbf{p}_j \Sigma^{-1}(\mathbf{p}_j - \mathbf{p}_i), (\mathbf{p}_j - \mathbf{p}_i)^T \Sigma^{-1}(\mathbf{p}_j - \mathbf{p}_i))$  if  $\mathbf{x}$  is from  $H_i$  and  $H_j$  respectively. Thus, our problem is now exactly equivalent to classification using MAP to classify a data point between two Gaussians with known means and same variance.

#### D. Greedy Algorithm (GA)

In this subsection, we analyze a simple greedy approach and show that it delivers a constant-factor approximate solution for the special case of two hypotheses. In the next subsection, we present a modified greedy algorithm for the general case of more than two hypotheses.

1) *Greedy Algorithm (GA)*: A straightforward algorithm for the LSS problem is a greedy approach wherein we iteratively select a single sensor at each stage. At each stage, we select the sensor that improves the localization accuracy  $O_{\text{acc}}(\mathbf{T})$  the most. The algorithm iterates until the given budget  $B$  is reached. We call this algorithm Greedy Algorithm (GA); see Algorithm 1 for the pseudo-code.

2) *Constant-Factor Approximation for 2 Hypotheses*: The approximation result of GA depends on two lemmas, which we prove in the appendix. The first lemma says that addition of a sensor to a given subset never reduces the value of  $O_{\text{acc}}$ :

*Lemma 1: The objective  $O_{\text{acc}}(\mathbf{T})$  is monotone in nature, i.e. if some sensor  $s_k \in \mathbf{S} \setminus \mathbf{T}$ , then  $O_{\text{acc}}(\mathbf{T} \cup \{s_k\}) \geq O_{\text{acc}}(\mathbf{T})$ .* The second lemma says that the amount of increase in accuracy follows a law of diminishing returns:

*Lemma 2: The objective  $O_{\text{acc}} \mathbf{T}$  is submodular in nature, i.e. if some sensor  $s_k \in \mathbf{S} \setminus \mathbf{T}_2$ , where  $\forall \mathbf{T}_1 \subseteq \mathbf{T}_2 \subseteq \mathbf{S}$ , we have:*

$$O_{\text{acc}}(\mathbf{T}_1 \cup \{s_k\}) - O_{\text{acc}}(\mathbf{T}_1) \geq O_{\text{acc}}(\mathbf{T}_2 \cup \{s_k\}) - O_{\text{acc}}(\mathbf{T}_2) \quad (10)$$

Intuitively, these lemmas follow from Theorem 1, where we showed that the problem of identifying the right hypothesis is equivalent to classifying between two unidimensional Gaussians. It is well-known that if an objective is monotone and submodular, then GA gives an approximation result [14], [15]. Thus, the following theorem on the performance of GA now holds:

*Theorem 2: For the special case of two hypotheses, GA gives a subset  $\mathbf{T}$  of sensors whose localization accuracy is at least 63% of the optimal.*  $\square$

3) *Performance of GA for More Than Two Hypotheses*: For the case of more than two hypotheses, GA no longer provides a constant-factor approximation. In fact, we can show via a counter-example that the  $O_{\text{acc}}()$  is not submodular for more than 2 hypotheses. We show this by providing a counter-example in Appendix C.

#### E. Auxiliary Greedy Algorithm (AGA)

In the section, we design an approximation algorithm for the general case of multiple hypotheses based on an auxiliary objective function. To do so, we first analyze the proof of Theorem 2 and see why it does not generalize if the number

---

#### Algorithm 1 Greedy Algorithm (GA)

---

**INPUT:** Set of available sensors  $\mathbf{S}$ , budget  $B$ , objective  $O_{\text{acc}}$   
**OUTPUT:** Subset of sensors  $\mathbf{T}$

```

1:  $\mathbf{T} \leftarrow \phi$  ▷ Start with empty subset of sensors
2: while  $|\mathbf{T}| \leq B$  do
3:    $L \leftarrow O_{\text{acc}}(\mathbf{T})$ 
4:    $\max \leftarrow 0$ 
5:   for all  $s \in \mathbf{S} \setminus \mathbf{T}$  do ▷ Iterate across all available sensors
6:      $M = O_{\text{acc}}(\mathbf{T} \cup \{s\}) - L$  ▷ Compute gain of sensor
7:     if  $M > \max$  then
8:        $\max \leftarrow M$  ▷ Pick sensor with highest gain
9:        $r \leftarrow s$ 
10:    end if
11:  end for
12:   $\mathbf{T} \leftarrow \mathbf{T} \cup \{r\}$  ▷ Add sensor with highest gain to subset
13: end while
14: return  $\mathbf{T}$ 

```

---

of hypotheses is greater than 2. This insight helps in defining an ‘‘auxiliary’’ objective function that is the key to designing the approximation algorithm for the general case.

1) *Auxiliary Function*: Let us consider a special case of MAP algorithm, viz.,  $\text{MAP}_{ij}$  which compares the likelihood of only two hypothesis  $H_i$  and  $H_j$  and returns the one with a higher likelihood. It is easy to formulate the objective function  $O_{\text{acc}}$  in terms of  $\text{MAP}_{ij}$  too. From Equation 6, we easily get:

$$O_{\text{acc}}(\mathbf{T}) = \sum_{i=0}^{m-1} P\left(\bigcap_{j \neq i} \text{MAP}_{ij}(\mathbf{x}) = i | H_i\right) P(H_i) \quad (11)$$

$$O_{\text{acc}}(\mathbf{T}) = \sum_{i=0}^{m-1} [1 - P\left(\bigcup_{j \neq i} \text{MAP}_{ij}(\mathbf{x}) = j | H_i\right)] P(H_i) \quad (12)$$

Above,  $\mathbf{x}$  represents the observation vector for the set of sensors  $\mathbf{T}$ . For the case of two hypothesis, the above expression is just  $\sum_{i=0}^1 [1 - P(\text{MAP}_{ij}(\mathbf{x}) = j | H_i)] P(H_i)$  where  $j$  is 1 if  $i$  is 0 and vice-versa; Theorem 2 essential shows that the term  $P(\text{MAP}_{ij}(\mathbf{x}) = i | H_i)$  is submodular. However, for the case of multiple hypothesis, computing the probability for a union of events involves product (and sum) of appropriate probability terms. Note that product of submodular functions need not be submodular, while sum of submodular functions is submodular. Thus, we *approximate* the above  $O_{\text{acc}}()$  expression as follows, so that it is a sum of submodular terms. In effect, in defining the auxiliary objective  $O_{\text{aux}}()$ , we estimate the probability of union of events in the above equation by just taking a summation of the probability of events, i.e., we ignore the other terms involving subsets of events. Formally, we define the auxiliary objective  $O_{\text{aux}}()$  for a set of sensors  $\mathbf{T}$  as:

$$O_{\text{aux}}(\mathbf{T}) = 1 - \sum_{i=0}^{m-1} \sum_{j \neq i} P(\text{MAP}_{ij}(\mathbf{x}) = j | H_i) P(H_i) \quad (13)$$

The above auxiliary objection function is submodular if the JPDs are Gaussian, as it is a sum of submodular functions

( $P(\text{MAP}_{ij}(\mathbf{x}) = i|H_i)$  is submodular, as per Theorem 2's proof). Note that, for a competitive algorithm for the original LSS problem, we also need to show that maximizing  $O_{\text{aux}}()$  also maximizes the original objective function  $O_{\text{acc}}()$ .

2) *Auxiliary Greedy Algorithm (AGA)*: We now modify our Greedy Algorithm (Algorithm 1) to iteratively maximize the auxiliary objective  $O_{\text{aux}}()$  instead of the original objective  $O_{\text{acc}}()$ . We call this algorithm as Auxiliary Greedy Algorithm (AGA). From the submodularity of the  $O_{\text{aux}}()$  for Gaussian JPDs, it is easy to see that AGA delivers a solution  $\mathbf{T}$  s.t.  $O_{\text{aux}}(\mathbf{T})$  is within 63% of the optimal  $O_{\text{aux}}()$  possible. The following lemma states that maximizing  $O_{\text{aux}}$  also maximizes  $O_{\text{acc}}$ . See Appendix E for a proof.

*Lemma 3: Let  $\mathbf{T}$  be a subset of sensors already selected by AGA at some iteration. We claim that  $O_{\text{aux}}(\mathbf{T}) \leq O_{\text{acc}}(\mathbf{T}) \leq 1 - \frac{1}{k}(1 - O_{\text{aux}}(\mathbf{T}))$ , where  $k$  is a value less than  $m$  that decreases as  $\mathbf{T}$  grows (i.e., over AGA's iterations).  $\square$*

We empirically evaluate the value of  $k$  defined above in §IV. The above lemma yields the below theorem, whose proof is shown in Appendix F.

*Theorem 3: For Gaussian JPDs, AGA delivers a subset  $\mathbf{T}$  of sensors such that*

$$P_{\text{err}}(\mathbf{T}) \leq 0.37 + 0.63kP_{\text{err}}(\text{OPT}),$$

where  $k$  is as defined in the above Lemma and  $\text{OPT}$  is the optimal solution.  $\square$

#### F. Optimizing AGA's Computation Cost

In a straightforward implementation of AGA (akin to Algorithm 1 for GA),  $O_{\text{aux}}$  function is computed (using Eqn. (13))  $Bn$  number of times where  $n$  is the total number of sensors. Eqn. (13) requires  $m^2$  computations of the expectation  $P(\text{MAP}_{ij}(\mathbf{x}) = j|H_i)$ , which, for Gaussian distributions, effectively requires computing the formula shown in Eqn. (11) of auxiliary material, and thus takes  $O(B^2)$  time as it involves matrix multiplication of the observation vector of dimension  $B$  with the covariance matrix of dimension  $B \times B$ . Thus, the overall time complexity of a straightforward implementation of AGA is  $O(m^2nB^3)$ . As mentioned before (and in §II), the number of hypotheses  $m$  can be large due to the large number of potential transmitter locations and power values; however, we can reduce the time complexity to  $O(Bn)$  as discussed below, based on some observations and optimizations.

1) *Reducing Number of Comparisons*: Consider a sensor  $s$  whose benefit is to be computed in the `for` loop of Algorithm 1. Below, we show that effectively we only need to consider a constant number of  $(H_i, H_j)$  pairs in Eqn. (13) when computing  $s$ 's benefit, and thus removing the  $m^2$  factor from the time complexity. We implicitly assume a single transmitter in the below discussion, and later extend our argument to multiple transmitters. Let us use  $R$  to denote the maximum transmission ‘‘range’’ of the transmitter; formally,  $R$  is such that, beyond  $R$ , the probability distribution of the signal received from the transmitter is similar to the signal received when there is no transmitter present. We stipulate that any observation  $x_s$  at  $s$ ,  $P(x_s|H_{i1}) = P(x_s|H_{i2})$  for any two

hypotheses  $H_{i1}$  and  $H_{i2}$  whose corresponding locations  $l_{i1}$  and  $l_{i2}$  are more than  $R$  distance away from  $s$ . The implication of the above observation is that, for a given sensor  $s$ , we can group all the hypotheses  $H_i$  with corresponding location  $l_i$  more than  $R$  distance away from  $s$  into one single ‘‘super’’ hypothesis  $H_s$ . Then, if the total number of hypotheses with corresponding locations within a distance of  $R$  from  $s$  is say  $G_R$ , then the total number of comparisons between pairs of hypotheses in Eqn. (13) is effectively only  $(G_R + 2)^2$ , involving  $G_R$  hypotheses,  $H_0$ , and  $H_s$ . The above brings down the overall time complexity of AGA to  $O(G_R^2nB^3)$ . Note that  $G_R$  is essentially equal to the number of grid locations within a circle of radius  $R$  times the total number of power levels, and thus, can be considered as constant (does not vary across problem instances)—which reduces AGA's time complexity to  $O(nB^3)$ .

2) *Independent Sensor Observations*: If we assume that the observations across sensors are conditionally independent, the JPDs can be instead represented by independent probability distributions at each sensor location. In this case, the covariance matrix is purely diagonal, which allows us to ‘‘incrementally’’ compute the benefit of a sensor from one AGA iteration to another and thus reduce AGA's time complexity by an additional factor of  $B^2$ —and thus to  $O(nB)$ . See Appendix G for details.

#### G. Generalizations

1) *Weighted (Distance-Based) Objective Function*: The probability of error  $P_{\text{err}}$  function penalizes *uniformly* for each misclassification. However, in general, it would be useful to assign different penalties or weights for different misclassifications. E.g., Eqn (13) should be generalized to:

$$O'_{\text{aux}}(\mathbf{T}) = 1 - \sum_{i=0}^m \sum_{j \neq i} w_{ij} P(\text{MAP}_{ij}(\mathbf{x}) = j|H_i) P(H_i)$$

Above,  $w_{ij}$  is the weight function. We note that our techniques and proofs of performance guarantees generalize easily to the above generalized function, irrespective of the weight function. In particular, weight  $w_{ij}$  can be the Euclidean distance between the locations  $l_i$  and  $l_j$  corresponding to the hypotheses  $H_i$  and  $H_j$ . For the general case of multiple transmitters, where  $H_i$  and  $H_j$  may represent configuration of multiple transmitters, a minimum-cost matching based objective can be used to define the weight  $w_{ij}$ ; if the number of transmitters in  $H_i$  and  $H_j$  are different, then an appropriately penalty for misses or false-alarms can be added to the weight.

2) *Non-Uniform Sensor Cost*: Another generalization of interest is to allow non-uniform cost for sensors, e.g., to prefer sensors with more (remaining) battery resources. Here, each sensor  $s$  may have a different cost  $c(s)$ , and the LSS problem constraint becomes: total cost of the selected set of sensors must be less than a given cost budget. For this version of the LSS problem, our algorithms need to be slightly modified in that we should pick the sensor that offers the highest improvement in the objective function per unit cost. To ensure a theoretical performance guarantee, we also need to use the ‘‘knapsack trick,’’ i.e., to pick better of the two solutions:

one returned by the modified algorithm, and the other the best one-sensor solution [16]. It can be shown the overall algorithm still offers a theoretical performance guarantee for submodular functions, but the performance ratio is reduced by a multiplicative factor of 2. The above model is useful when designing a “load-balanced” strategy to maximize network lifetime of a system—therein, the sensor-selection algorithm must be run periodically based on the remaining battery resources.

#### IV. EVALUATION

In this section, we evaluate the performance of our algorithms developed in the previous sections. We start with a description of the evaluation platforms used in our experiments.

##### A. Implementation

1) *Implementation Technique*: To evaluate whether AGA can be feasibly used, we compute the cost of data collection and then observe the execution time of AGA. The cost of data collection can be evaluated by the total amount of data collection. It is possible to collect data either manually (as in our testbed) or using drones/robots [17]. In our case, since there are  $m$  hypotheses, and propagation is symmetric in nature, we have collected data from a total of  $m^2/2$  grid cells. For each of these cases, we collect a total of 250 KB of IQ data, and then perform FFT on the sensors themselves with a bin size of 256 before sending it to the server. Thus, the server only receives around 1 KB of data per transmitter location per sensor. The total data that the server collects is therefore, also equal to  $m^2/2$  KB. Computing the joint probability density functions using this data is trivial, as it involves only computing the mean and standard deviation of each sensor-transmitter location paper.

To compute execution time, we implement two distinct versions of AGA using python. The first version, called AGA-Basic, does not utilize the optimizations discussed in Section III-F. The second version, called AGA-OPT, includes these optimizations. Each version utilizes multiple cores of a CPU using joblib library [18] to compute the gain of each available sensor in parallel. It also uses the numpy library to vectorize operations wherever possible to make execution fast. We run three different instances of AGA – with 100, 1600 and 4096 hypotheses. Each of these instances have 100 available sensors and a budget of 20. We execute this on a Core i9-7900X CPU having a frequency of 3.30GHz and 20 cores.

2) *Implementation on CPU*: Figure 4(i) shows the execution time of these three instances. We note that for small instances, the execution time is relatively small. For example, for 100 hypotheses, AGA-basic only takes 13s to execute. However, this rises to 28 minutes for 1600 hypotheses and to over 10 hours for 4096 hypotheses. We also find that for small instances, the optimizations do not lead to much improvement due to the overhead of maintaining the data structures. However, there is a large improvement for 4096 hypotheses, where we get an execution time of 150 minutes using the optimized version.

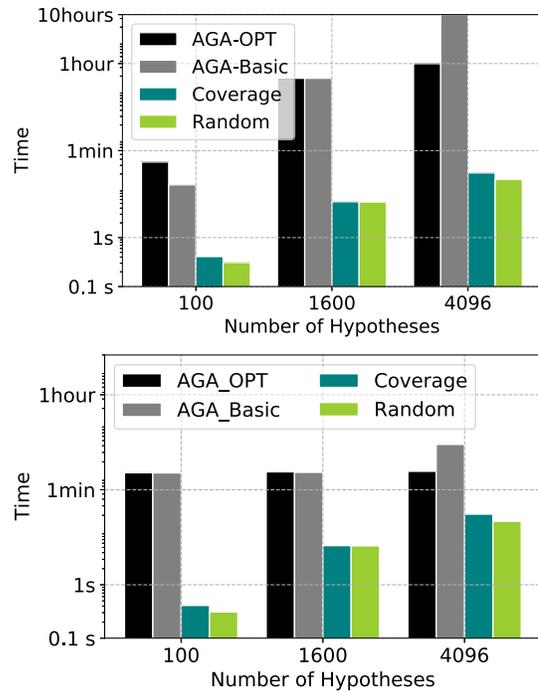


Fig. 4. Execution time of AGA and baseline techniques both with and without the optimizations on a (i) CPU and on a (ii) GPU.

3) *Implementation on GPU*: Although execution on CPU’s using our optimizations is feasible, we further note that the bulk of execution time is spent on matrix operations. This suggests that execution on a GPU can lead to much better utilization of data-level parallelism, and further speed up execution. To evaluate this, we optimize the computation of the gain of the sensors using numba library [19] to execute it on a GPU. While using numba library, we ensure that the computation requests from the GPU for all the sensor gains are batched into a single request, in order to reduce movement of data between the CPU and GPU memory. This optimizes the computation of the sensor gains. We utilize an nVidia GTX 2080Ti GPU having 4352 cores with a processor clock of 1.545 GHz. We then note the execution time for each of the three instances of both AGA-Basic and AGA-OPT.

Figure 4(ii) shows the execution times on a GPU. We note that execution is much faster on a GPU than on a CPU. For example, AGA-OPT now runs in 123s, 130s and 133s for 100, 1600 and 4096 hypotheses respectively. This shows that AGA can run very fast on a system with GPU, with a speedup of up to 155 times on the large instances, compared to AGA-Basic. While this is still slower than the baseline techniques, it is still feasible to use it in realistic settings.

##### B. Evaluation Platforms

We use the following three evaluation platforms with varying fidelity of signal propagation characteristics, to demonstrate the performance of our techniques.

- *Simulation based on synthetic data*. To demonstrate the scalability of our techniques and the sensitivity of our

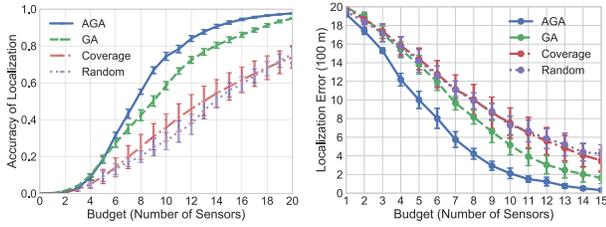


Fig. 5. Comparison of various techniques for (i) Localization accuracy ( $O_{acc}$  ()), and (ii) Weighted localization error, for varying available budget (number of sensors).

algorithms to changes in settings, we consider a large geographic area of 4km by 4km, with signal path-loss values generated using the SPLAT! application for the Longley-Rice [20]. We use the noise in the sensor measurements (measured independently) to generate the required JPDs. We assume observations to be conditionally independent, thus representing the JPDs as set of probability distributions, one for each sensor and intruder configuration pair. Unless otherwise stated, for this large-scale platform, we use 100m x 100m grid cells giving 1600 potential locations, randomly deploy a transmitter at the height of 30m at a random power between 27-33 dBm which corresponds to roughly 250-750m of transmission range. We randomly deploy 100 spectrum sensors in the area.

- *Indoor Data.* We use publicly available data [21], which deploys transmitters and receivers at 44 locations at an indoor building of an area of  $14m \times 14m$ . Here, we use 1m x 1m grid cells, thus giving us a total of 196 potential locations and hypotheses. The transmitters transmit at a frequency of 2.4GHz, with a transmit power of 10mW, and antenna gains of 1.1 dBi.
- *Outdoor Testbed.* Finally, to evaluate our techniques in a more practical outdoor setting, we deploy our own testbed in a parking area of dimension  $32m \times 32m$ .<sup>2</sup> Each grid cell has size of 3.2m x 3.2m, thus giving us a total of 100 grid cells. We place a total of 18 sensors on the ground. The sensors consist of single-board computers such as Raspberry Pi's and Odroid-C2's connected to an RTL-SDR dongle. The RTL-SDRs use dipole antennas. We collect raw Inphase-Quadrature (I/Q) samples from the RTL-SDRs [22], while transmitting data using a USRP-based transmitter from each grid cell at a height of 1.5m. We perform FFT on the I/Q samples with a bin size of 256 samples to get the signal power values, and then utilize the mean and standard deviation of the power reported for each of the sensors.

1) *Metrics:* We evaluate the performance of a localization strategy in terms of two key metrics: (i) Localization accuracy, i.e.,  $O_{acc}(\mathbf{T})$ , and (ii) Weighted localization error, which weights the misclassification error by the Euclidean distance between the actual and the predicted location (§III-G).

2) *Compared Algorithms:* We implement both of our designed algorithms, AGA and GA. We also implement two

<sup>2</sup>We have made this dataset publicly available at: <https://github.com/Wings-Lab/IPSN-2020-data>.

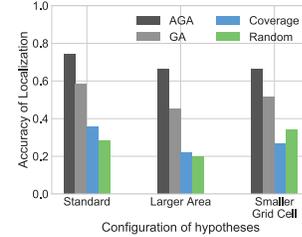


Fig. 6. Comparison for configurations with different number of hypotheses, with a fixed budget of 10 sensors.

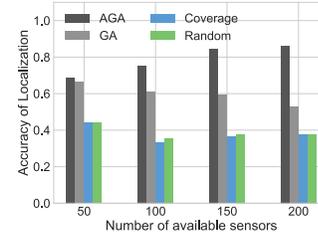


Fig. 7. Comparison for varying number of available sensors, with a fixed budget of 10 sensors.

other techniques for comparison purposes. The first technique, called *Coverage*, is the selection heuristic from the recent work [2], which essentially tries to maximize the “coverage” of the sensors in a greedy manner.<sup>3</sup> We also implement a *Random* algorithm which selects the required sensors randomly. We implement these algorithms in python, with extensive use of *numpy* library for vectorized operations. To ensure that our results are statistically significant, we run each of the algorithms 100 times; the range of values is plotted in each of the figures.

### C. Simulation Based on Synthetic Data

1) *Varying Budget:* Figure 5 shows the performance of our techniques for budgets varying from 1 to 20 sensors. We observe that AGA and GA easily outperform other two algorithms in terms of both metrics, with AGA outperforming even GA quite significantly. For example, AGA outperform *Coverage* by up to 39% and 56% for localization accuracy and error respectively, while outperforming GA by 15% and 50% for the two metrics respectively.

2) *Varying Number of Hypotheses:* We now show the performance of our algorithms in terms of localization accuracy, for varying number of hypotheses. In Figure 6, we plot three different cases: (i) the default configuration of 100m by 100m grid cells, (ii) a larger area of 6km by 6km with 100m by 100m grid cells giving 3600 potential locations, and finally (iii) a configuration with default 4km by 4km area, but smaller 62.5m by 62.5m grid cells. First, we observe that AGA continues to outperform other techniques significantly across different cases, with the performance gap between

<sup>3</sup>Their approach *Metropolis* performs worse than their greedy approach in open areas [2], and hence, not compared. Similarly, [23] selects sensors to measure only spatial phenomena such as temperature, and thus is not applicable to our problem.

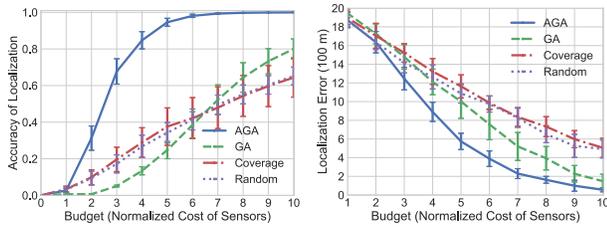
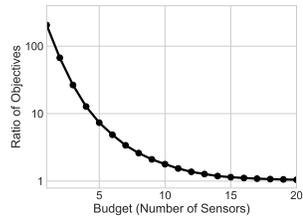


Fig. 8. Comparison of various techniques, for sensors with non-uniform cost.

Fig. 9. Values of  $k$  (from Lemma 3) for varying budget.

AGA and others (especially GA) increasing with increase in number of hypotheses. Also, as expected, with increase in area and thus number of hypotheses, the accuracy of each of the algorithms falls, but deterioration in AGA's accuracy is much less compared to other techniques. Finally, the performance of the Coverage algorithm falls significantly when the number of hypotheses increases. This is because the Coverage algorithm is designed considering indoor localization and thus works well for smaller areas. In fact, for the case of smaller grid cells, the performance of the Coverage algorithm is worse than that of Random.

3) *Varying Sensor Density*: Figure 7 shows the accuracy of localization for varying sensor density (i.e., number of available sensors) with a fixed budget of 10 sensors. We note that the accuracy of localization of AGA significantly improves when we increase the number of sensors. For example, it increases by 16% when the number of sensors increases from 50 to 150. In contrast, the performance of GA and Coverage both actually reduces by 7%. This is because having with an increase in sensor density GA and Coverage select sensors that are too close to one another to be useful. In contrast, AGA has a submodular objective which leads to an increase in accuracy whenever the value of the optimal value increases.

4) *Non-Uniform Sensor Costs*: We also evaluate performance of techniques under the setting of sensors with non-uniform cost. We obtain the costs by computing the energy consumption of each sensor by varying the number of samples from 32 to 2048 in multiples of 2. We then randomly assign some energy and corresponding distributions to each sensor. Figure 8 shows the performance for such heterogeneous sensors. As expected, AGA continues to outperform the other techniques in both localization error. However, GA performs much worse than expected in case of heterogeneous sensors.

5) *Empirical Evaluation of  $k$  Value*: We now evaluate the  $k$  value as defined in Lemma 3. In particular, the performance guarantee of AGA depends on the value of  $k$ , with better performance guarantee for lower  $k$  (ideally,  $k$  should be equal

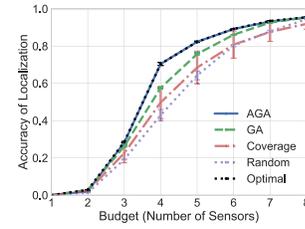


Fig. 10. Comparison with an optimal algorithm, for small instances of the problem.

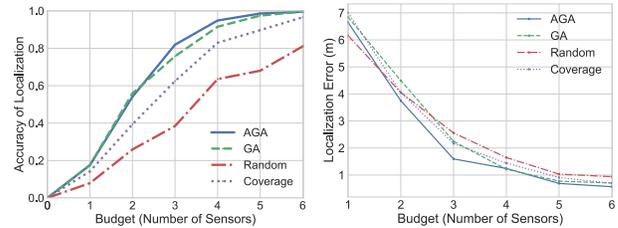


Fig. 11. Performance over public indoor data.

to 1). Figure 9 shows the value of  $k$  for varying budget. We observe that for a very low budget, the value of  $k$  is very large, but it reduces rapidly with increase in budget. In particular, for budgets of 10 and 15 sensors, the value of  $k$  is 1.78 and 1.19 respectively. This shows that AGA's performance guarantee as per Theorem 3 reaches its near-best for a moderately small budget.

6) *Comparison With Optimal in Small Instances*: We further confirm AGA's performance with respect to optimal, we consider small instances of the problem (with 100 hypotheses) and compare AGA with an optimal algorithm. The optimal algorithm uses exhaustive search, which is impossible to execute over larger instances. See Figure 10. We observe that AGA and optimal perform near-identically, with the optimal algorithm yielding at most 0.7% higher localization accuracy than AGA. Note that GA performs worse than AGA and optimal even in this case, albeit by a smaller amount than in cases with larger number of hypotheses. Moreover, even for such small instances, the optimal algorithm takes at least an order of magnitude more execution time compared to both AGA and GA.

#### D. Evaluation in Indoor and Outdoor Testbeds

1) *Indoor Data*: We now evaluate our techniques over a publicly available data-trace taken in an indoor environment, as described in the previous subsection. See Figure 11. We again observe similar performance trends as in previous experiments, for both the performance metrics. The relatively smaller performance gap between AGA and GA is likely due to smaller a number of hypotheses.

2) *Outdoor Testbed*: Figure 12 shows the performance of various algorithms over our outdoor testbed described in the previous subsection. We observe that AGA again performs the best among all techniques in both the metrics, with AGA outperforming Coverage by up to 18%. As in the indoor

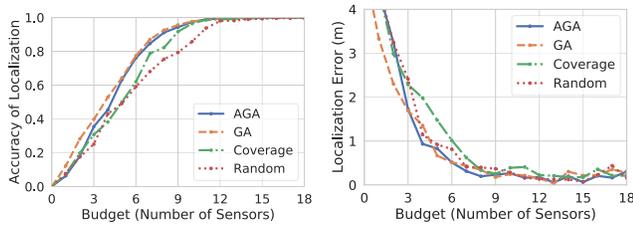


Fig. 12. Performance over outdoor testbed data.

testbed, the performance gap between the AGA and GA is less (close to 1%) compared to the large-scale simulations due to a small number of hypotheses. Also, unlike in the case of the simulations, the performance of Coverage algorithm is significantly better than that of random. This is because the Coverage algorithm is designed in a way that it performs much better when the experiment is performed over a smaller area.

## V. RELATED WORK

### A. Indoor Localization

Indoor localization has been a topic of interest for a long time [24], [25]. Our technique for the hypothesis-based framework utilizes the fingerprinting technique [26] that has been discussed in earlier works. The work [27] fuses IMU sensors and WiFi RSSI measurements to improve the accuracy of indoor localization. Similar techniques have been used using sound waves too [28]. Algorithmic techniques to localize a transmitter include using techniques like multilateration, k-nearest neighbor, bayesian averaging, multi-layer perceptron, apart from maximum a posteriori (MAP) estimate. Our work utilizes the objective of MAP to derive the objective of sensor selection, and does not study the performance of the other existing localization approaches. Note that since our work studies the orthogonal problem of selecting sensors in the context of fingerprint and MAP-based localization, we do not compare our work with these approaches.

### B. Sensor Selection for Transmitter Localization

A large number of works have developed techniques for detecting and localizing transmitters or intruders that emit radio signals [10], [29]. Note that the transmitter localization problem is slightly different from the problem of indoor localization. To the best of our knowledge, none of these prior works on transmitter localization either have addressed the optimization problem addressed in the paper. The closest related works are [1] and [2] as discussed next. The work [1] focuses on *detection* of unauthorized transmitters using low-cost sensors in the context of shared spectrum systems; they consider the problem of selection of sensors in this context, and propose a heuristic with no performance guarantees. The key difference of our work from theirs is that they focus on detection of transmitters, which is a much simpler problem than localization of transmitters. In addition, [2] considers selection of sensors for transmitter localization, but with a objective of maximizing the “coverage” of the region by the sensors. They present heuristics without any performance

guarantees. Nevertheless, we implement their approach and compare with our techniques (§IV).

### C. Sensor Selection in Sensor Networks

Sensor selection is a natural problem to address in the context of wireless sensor networks deployed to detect and/or localize an event or phenomenon (see [30] for a survey). Many of these works have leverage the submodularity property to develop greedy approximation algorithms. The closest work among these is that of [23] which shows approximability of the greedy approach for the problem of minimizing uncertainty in estimating a spatial phenomenon (e.g., temperature). However, in general, the key difference of our work with these works is our desired objective function ( $O_{\text{acc}}$  or  $P_{\text{err}}$ )—and thus, the making the proof of monotonicity and/or submodularity of the objective function very different. In our case, we had to even circumvent the non-submodularity of the objective function  $O_{\text{acc}}$  by considering an appropriate auxiliary objective function.

### D. Online Selection of Sensors

An alternate formulation of our sensor selection problem could be to select sensors adaptively *based* on the observations of previously selected sensors. This online problem is similar to the adaptive stochastic optimization problem addressed in other contexts [31]–[34]. However, in online selection, a sensor is selected based on analysis (which will incur non-trivial latency) of observations of previous sensors. This makes localization based on *near-simultaneous* sensor observations, required to localize intermittent transmitters, infeasible. Also, note that online selection needs to be done anew for each localization, which may be performed very frequently (e.g., every second or fraction of a second) in many applications, e.g., spectrum patrolling. Thus, our focus is on offline selection.

## VI. DISCUSSIONS

We now discuss some of the assumptions made by our study, and how essential these assumptions are to detect unauthorized transmissions in the wild.

### A. Presence of Multiple Transmitters

Our hypothesis-based techniques naturally generalize to the case of multiple transmitters, if we represent each *combination* of configurations of present transmitters by a separate hypothesis. Since the MAP, GA, and AGA algorithms are formulated in terms of hypotheses, they generalize naturally to localization of multiple transmitters. However, the key challenge arises due to the large number of hypotheses—exponential in the number of potential transmitters—and thus, the high time complexity of AGA. In our prior work [7], we develop an efficient MAP-based technique for localization of multiple transmitters. Similarly, our sensor-selection algorithms (GA and AGA) can also be modified to work efficiently for the case of multiple transmitters as follows.

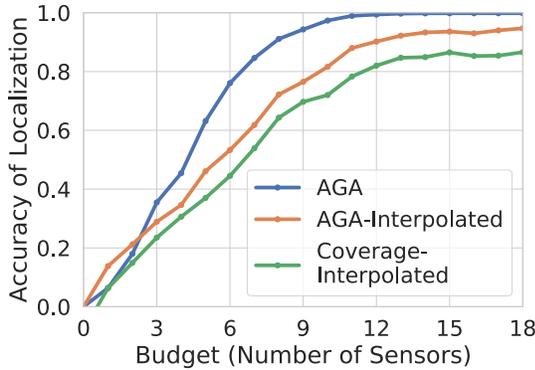


Fig. 13. Performance of AGA and Coverage algorithms when half the JPD's are obtained from empirical measurement, and the other half is obtained by interpolation.

The key observation is that, for a given hypothesis  $H_i$ , the probability distribution of observations at a sensor  $s$  depends only on the configuration of transmitters in  $H_i$  that within a distance of  $R$  of  $s$ . I.e., for any observation  $x_s$  at a sensor  $s$ ,  $P(x_s|H_{i1}) = P(x_s|H_{i2})$  for any two hypotheses  $H_{i1}$  and  $H_{i2}$  that have the same configuration (locations and powers) for transmitters that are within a distance of  $R$  of  $s$ . The implication of the above observation(s) is that, for a given  $s$ , we can group the given hypotheses into *equivalence classes* based on the configuration of transmitters close to  $s$ , and to compute the benefit of a sensor  $s$  with AGA's iteration, we only need to compare pairs of equivalence classes (rather than the original hypotheses, which are exponentially many). The number of such equivalence classes is easily seen to be equal to  $G_R^T$  where  $G_R$  is the number of locations (grid cells) within  $R$  times the number of power levels, and  $T$  is the maximum number of transmitters possible/allowed within a range  $R$  of  $s$  (or any location). Thus, computation of benefit of  $s$  requires consideration of  $G_R^{2T}$  pairs of equivalence classes. If we assume  $T$  to be a small constant, then the overall time complexity of AGA reduces to  $O(nB^3)$  as before, and to  $O(nB)$  if we assume independence of sensor observations.

In our work, we have assumed the existence of only a single transmitter in the area under consideration. The rationale behind this assumption is that in many applications multiple concurrent transmitters do not exist due to the use of an effective multiple access protocol that avoids concurrent transmissions in the same neighborhood. Transmissions from far-away transmitters can be treated as noise.

### B. Presence of Training Data

Our framework assumes that training data for each of the hypothesis is available. This training is usually expensive as it requires a lot of manual effort. While reducing training effort involved in utilizing MAP is not the primary focus of this work, we studied the performance of our techniques when we collected only half the original training data. We obtained the means of the rest of the joint probability distributions (JPD's) by linear interpolation. We then compared (Figure 13)

the performance of AGA and Coverage algorithms with and without interpolations.

We observe that the performance of the algorithms do reduce on reducing the amount of training. The reduction in performance is highest (close to 18% at budget of 7) when the budgeted sensors is moderately high, but it reduces (around 8% at budget of 5) with further increases in the budget. While for clarity we do not show the reduction for the other techniques, this reduction in performance is observed for all the techniques, as they all depend on MAP for the final localization. We leave it to future work to investigate better interpolation techniques to enable more accurate localization.

### C. Knowledge of Selected Sensors by Transmitters

In this work, we have assumed that the transmitters are unaware of the sensors that are selected. This is because our work is evaluated on the prior probabilities of each hypothesis being equal. If the transmitters are aware of the selected sensors, in certain types of applications (e.g., spectrum patrolling problems when the transmitters are unauthorized) they would try to evade the sensors by appearing at locations that are less closely monitored. This in turn would gradually change the prior probabilities, leading to a change in the subset of selected sensors. Studying the changing dynamics of how the unauthorized transmitters and selected sensors can react to changing priors is left for future work.

### D. Validation Over Larger Testbed

In this work, we have validated our algorithms over smaller testbeds and large-scale simulations. Smaller testbeds do provide a good understanding of the algorithms involved, as the power of the transmitter is correspondingly low. Experimental validation by testing over a larger area is currently difficult, as this requires regulatory approval to transmit with a larger power in the wild. However, the propagation models used in our large-scale simulations are known to be used by cellular service providers [35], and so we believe they provide us a good insight into the performance of our algorithms if they are actually deployed.

## VII. CONCLUSION

In this work, we have considered the hypothesis-driven approach for localization of transmitters, and developed techniques to optimize the localization accuracy under a constraint of limited resources. Our developed techniques have been shown to yield provably approximate solutions, while also having low running time. Our work can be instrumental in maximizing the network lifetime of a spectrum monitoring and/or patrolling system. Furthermore, we have evaluated our work using three distinct techniques – large-scale simulation, publicly available dataset and our own testbed. We are making the source code available to the community at the URL “<https://bitbucket.org/arani89/sensorselection-infocom>”. Our future work focuses on improving our theoretical performance guarantee results, and developing similar sensor selection approximation algorithms for other localization approaches that are not hypothesis-driven.

## REFERENCES

- [1] A. Bhattacharya, A. Chakraborty, S. R. Das, H. Gupta, and P. M. Djurić, "Spectrum patrolling with crowdsourced spectrum sensors," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 271–281, Mar. 2020, doi: [10.1109/TCCN.2019.2939793](https://doi.org/10.1109/TCCN.2019.2939793).
- [2] M. Khaledi *et al.*, "Simultaneous power-based localization of transmitters for crowdsourced spectrum monitoring," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, Oct. 2017, pp. 235–247, doi: [10.1145/3117811.3117845](https://doi.org/10.1145/3117811.3117845).
- [3] A. Nika *et al.*, "Empirical validation of commodity spectrum monitoring," in *Proc. 14th ACM Conf. Embedded Netw. Sensor Syst. CD-ROM (SenSys)*, Nov. 2016, pp. 96–108, doi: [10.1145/2994551.2994557](https://doi.org/10.1145/2994551.2994557).
- [4] A. Chakraborty, M. S. Rahman, H. Gupta, and S. R. Das, "SpecSense: Crowdsensing for efficient querying of spectrum occupancy," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9, doi: [10.1109/infocom.2017.8057113](https://doi.org/10.1109/infocom.2017.8057113).
- [5] V. Cevher, P. Boufounos, R. G. Baraniuk, A. C. Gilbert, and M. J. Strauss, "Near-optimal Bayesian localization via incoherence and sparsity," in *Proc. IEEE Comput. Soc. Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Apr. 2009, pp. 205–216.
- [6] A. Bhattacharya, C. Zhan, H. Gupta, S. R. Das, and P. M. Djurić, "Selection of sensors for efficient transmitter localization," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jul. 2020, pp. 2410–2419, doi: [10.1109/INFOCOM41043.2020.9155230](https://doi.org/10.1109/INFOCOM41043.2020.9155230).
- [7] C. Zhan, H. Gupta, A. Bhattacharya, and M. Ghaderibaneh, "Efficient localization of multiple intruders in shared spectrum system," in *Proc. 19th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Apr. 2020, pp. 205–216, doi: [10.1109/IPSN48710.2020.00025](https://doi.org/10.1109/IPSN48710.2020.00025).
- [8] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Hoboken, NJ, USA: Wiley, 2012.
- [9] P. K. Penumarthy *et al.*, "Multirobot exploration for building communication maps with prior from communication models," in *Proc. Int. Symp. Multi-Robot Multi-Agent Syst. (MRS)*, Dec. 2017, pp. 90–96, doi: [10.1109/mrs.2017.8250936](https://doi.org/10.1109/mrs.2017.8250936).
- [10] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2568–2599, 3rd Quart., 2019, doi: [10.1109/COMST.2019.2911558](https://doi.org/10.1109/COMST.2019.2911558).
- [11] *NSF Workshop on Spectrum Measurements Infrastructure*, Assoc. Comput. Mach., New York, NY, USA, 2016, pp. 22–23.
- [12] W. Sun, M. Xue, H. Yu, H. Tang, and A. Lin, "Augmentation of fingerprints for indoor WiFi localization based on Gaussian process regression," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10896–10905, Nov. 2018, doi: [10.1109/TVT.2018.2870160](https://doi.org/10.1109/TVT.2018.2870160).
- [13] A. Chakraborty, L. E. Ortiz, and S. R. Das, "Network-side positioning of cellular-band devices with minimal effort," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 2767–2775, doi: [10.1109/INFOCOM.2015.7218669](https://doi.org/10.1109/INFOCOM.2015.7218669).
- [14] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—I," *Math. Program.*, vol. 14, no. 1, pp. 265–294, Dec. 1978, doi: [10.1007/BF01588971](https://doi.org/10.1007/BF01588971).
- [15] J. Kleinberg and E. Tardos, *Algorithm Design*. Chennai, India: Pearson, 2006.
- [16] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Inf. Process. Lett.*, vol. 70, no. 1, pp. 39–45, 1999, doi: [10.1016/S0020-0190\(99\)00031-9](https://doi.org/10.1016/S0020-0190(99)00031-9).
- [17] R. Ayyalasomayajula *et al.*, "Deep learning based wireless localization for indoor navigation," in *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, Apr. 2020, pp. 1–14, doi: [10.1145/3372224.3380894](https://doi.org/10.1145/3372224.3380894).
- [18] Joblib. (Oct. 2018). *Joblib/Joblib*. [Online]. Available: <https://github.com/joblib/joblib>
- [19] S. K. Lam, A. Pitrou, and S. Seibert, "Numba: A LLVM-based Python JIT compiler," in *Proc. 2nd Workshop LLVM Compiler Infrastruct. HPC*, 2015, pp. 1–6, doi: [10.1145/2833157.2833162](https://doi.org/10.1145/2833157.2833162).
- [20] K. A. Chamberlin and R. J. Luebbers, "An evaluation of Longley–Rice and GTD propagation models," *IEEE Trans. Antennas Propag.*, vol. AP-30, no. 6, pp. 1093–1098, Nov. 1982, doi: [10.1109/TAP.1982.1142958](https://doi.org/10.1109/TAP.1982.1142958).
- [21] N. Patwari. (Sep. 2007). *CRAWDAD Dataset Utah/CIR (V. 2007-09-10)*. [Online]. Available: <https://crawdadd.org/utah/CIR/20070910>
- [22] (2018). *RTL-SDR (RTL2832U) and Software Defined Radio News and Projects*. Accessed: Oct. 18, 2018. [Online]. Available: <https://www.rtl-sdr.com/>
- [23] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, vol. 9, no. 2, pp. 235–284, 2008.
- [24] A. Yassin *et al.*, "Recent advances in indoor localization: A survey on theoretical approaches and applications," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1327–1346, 2nd Quart., 2017, doi: [10.1109/COMST.2016.2632427](https://doi.org/10.1109/COMST.2016.2632427).
- [25] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE Conf. Comput. Commun., 19th Annu. Joint Conf. IEEE Comput. Commun. Soc. (INFOCOM)*, vol. 2, Mar. 2000, pp. 775–784, doi: [10.1109/INFCOM.2000.832252](https://doi.org/10.1109/INFCOM.2000.832252).
- [26] R. Liu, T.-N. Do, U.-X. Tan, and Y. Chau, "Fusing similarity-based sequence and dead reckoning for indoor positioning without training," *IEEE Sensors J.*, vol. 17, no. 13, pp. 4197–4207, Jul. 2017, doi: [10.1109/JSEN.2017.2706303](https://doi.org/10.1109/JSEN.2017.2706303).
- [27] R. Liu, C. Yuen, T.-N. Do, M. Zhang, Y. L. Guan, and U.-X. Tan, "Cooperative positioning for emergency responders using self IMU and peer-to-peer radios measurements," *Inf. Fusion*, vol. 56, pp. 93–102, Apr. 2020, doi: [10.1016/j.inffus.2019.10.009](https://doi.org/10.1016/j.inffus.2019.10.009).
- [28] Y. Sun, J. Chen, C. Yuen, and S. Rahardja, "Indoor sound source localization with probabilistic neural network," *IEEE Trans. Ind. Electron.*, vol. 65, no. 8, pp. 6403–6413, Aug. 2018, doi: [10.1109/TIE.2017.2786219](https://doi.org/10.1109/TIE.2017.2786219).
- [29] R. K. Martin and R. Thomas, "Algorithms and bounds for estimating location, directionality, and environmental parameters of primary spectrum users," *IEEE Trans. Wireless Commun.*, vol. 8, no. 11, pp. 5692–5701, Nov. 2009, doi: [10.1109/TWC.2009.090494](https://doi.org/10.1109/TWC.2009.090494).
- [30] H. Rowaihy *et al.*, "A survey of sensor selection schemes in wireless sensor networks," *Proc. SPIE*, vol. 6562, May 2007, Art. no. 65621A.
- [31] H. Wang, K. Yao, G. Pottie, and D. Estrin, "Entropy-based sensor selection heuristic for target localization," in *Proc. 3rd Int. Symp. Inf. Process. Sensor Netw. (IPSN)*, 2004, pp. 36–45, doi: [10.1145/984622.984628](https://doi.org/10.1145/984622.984628).
- [32] Y. Chen, S. H. Hassani, A. Karbasi, and A. Krause, "Sequential information maximization: When is greedy near-optimal?" in *Proc. 28th Conf. Learn. Theory*, vol. 40, Jul. 2015, pp. 338–363. [Online]. Available: <http://proceedings.mlr.press/v40/Chen15b.html>
- [33] A. Krause and C. Guestrin, "Near-optimal nonmyopic value of information in graphical models," in *Proc. 21st Conf. Uncertainty Artif. Intell. (UAI)*, Arlington, VA, USA, 2005, pp. 324–331.
- [34] D. Golovin and A. Krause, "Adaptive submodularity: Theory and applications in active learning and stochastic optimization," *J. Artif. Intell. Res.*, vol. 42, no. 1, pp. 427–486, Nov. 2011.
- [35] J. A. Magliacane. (2008). *SPLAT! A Terrestrial RF Path Analysis Application for Linux/Unix*. [Online]. Available: <https://www.qsl.net/kd2bd/splat.html>



**Arani Bhattacharya** (Member, IEEE) received the B.Tech. degree in computer science from West Bengal University of Technology in 2011, the M.Tech. degree in computer science from the Indian Statistical Institute in 2013, and the Ph.D. degree in computer science from Stony Brook University in 2019. He is currently an Assistant Professor with Indraprastha Institute of Information Technology, Delhi (IIIT-Delhi), India. He is passionate about integrating both model and data-driven techniques into working systems and studying their performance tradeoffs. His research interests include wireless networks, edge computing, studying problems related to performance of traffic surveillance systems, WiFi 6, and mobile systems in developing countries.



**Caitao Zhan** (Graduate Student Member, IEEE) received the B.S. degree in computer science and technology from China University of Geosciences, Wuhan, China, in 2017, and the Ph.D. degree from the Department of Computer Science, Stony Brook University. He does research in the broad area of computer networks and focus at the intersection of computer networks and machine learning. Also, he is actively studying quantum networks and investigates network layer protocols in the quantum network stack.



**Abhishek Maji** received the B.E. degree in electrical engineering from Jadavpur University, India, in 2014, the master's degree from the Indian Institute of Technology Kanpur, India, and the master's degree from the KTH Royal Institute of Technology, Sweden. He currently works as a System Development Engineer with Hitachi ABB Power Grids. His research interest include applying reinforcement learning-based techniques to improve the efficiency of power grids.



**Samir R. Das** (Member, IEEE) received the Ph.D. degree in computer science from Georgia Institute of Technology, Atlanta, GA, USA, in 1994. Since 2002, he has been with Stony Brook University, Stony Brook, NY, USA, where he is currently a Professor with the Department of Computer Science and also works as the Department Chair. His research interests include mobile/wireless networking protocols, systems, and performance evaluation. He was a recipient of the U.S. National Science Foundation's CAREER Award in 1998. He has been on the Editorial Board of several major journals, including the IEEE TRANSACTIONS ON NETWORKING and the IEEE TRANSACTIONS ON MOBILE COMPUTING. He also co-chaired premier conferences, such as ACM MobiCom and ACM MobiHoc. He routinely serves on the program committees of top networking and mobile computing conferences.



**Himanshu Gupta** (Member, IEEE) received the B.Tech. degree in computer science and engineering from the Indian Institute of Technology (IIT) Bombay, Mumbai, India, in 1992, and the M.S. and Ph.D. degrees in computer science from Stanford University, Stanford, CA, USA, in 1999. He is currently a Professor with the Department of Computer Science, Stony Brook University, Stony Brook, NY, USA. His research interests include theoretical issues in wireless networking, sensor networks, and more recently quantum networks.



**Petar M. Djurić** (Fellow, IEEE) received the B.S. and M.S. degrees in electrical engineering from the University of Belgrade, Belgrade, Yugoslavia, and the Ph.D. degree in electrical engineering from the University of Rhode Island, Kingston, RI, USA. He is a SUNY Distinguished Professor and currently a Chair of the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY, USA. His research has been in the area of signal and information processing with primary interests in the theory of machine learning; Monte Carlo-based methods; signal and information processing over networks; signal modeling, detection, and estimation; RFID; and the IoT. He is also a Fellow of EURASIP. He was a recipient of the *IEEE Signal Processing Magazine* Best Paper Award in 2007 and the EURASIP Technical Achievement Award in 2012. He was the Chair of Excellence of Universidad Carlos III de Madrid-Banco de Santander in 2008. He has been on numerous committees of the IEEE Signal Processing Society and of many professional conferences and workshops. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS. From 2008 to 2009, he was a Distinguished Lecturer of the IEEE Signal Processing Society.