On Passive Wireless Device Fingerprinting using Infinite Hidden Markov Random Field

Feng Chen, Qiben Yan, Chowdhury Shahriar, Chang-Tien Lu, Wenjing Lou, and T. Charles Clancy

{chenf, qbyan, cshahria, ctlu, wjlou, tcc}@vt.edu

Virginia Tech, Falls Church, VA, USA

Abstract

This paper presents a new concept of device fingerprinting (or profiling) to enhance wireless security using Infinite Hidden Markov Random Field (iHMRF). Wireless device fingerprinting is an emerging approach for detecting spoofing attacks in wireless network. Existing methods utilize either time-independent features or time-dependent features, but not both concurrently due to the complexity of different dynamic patterns. In this paper, we present a unified approach to fingerprinting based on iHMRF. The proposed approach is able to model both time-independent and time-dependent features, and to automatically detect the number of devices that is dynamically varying. We propose the first iHMRF-based online classification algorithm for wireless environment using variational incremental inference, micro-clustering techniques, and batch updates. Extensive simulation evaluations demonstrate the effectiveness and efficiency of this new approach.

Index Terms

Passive Device Fingerprinting, Hidden Markov Random Field, Nonparametric Bayesian Methods

I. Introduction

Nowadays, the proliferation of mobile devices moves the wireless networks toward “anytime-anywhere” mobile service model. However, the open nature of wireless networks renders them susceptible to various types of spoofing attacks. For example, the adversaries can collect nodes’ identity information by passively monitoring the network traffic, and then masquerade as legitimate nodes to disrupt network operations. Various attacks can be launched, such as packet injection [1], Sybil attack [2], masquerade attack [3], etc. These identify-based attacks may hinder normal communication and result in privacy leakage, which will lead to a huge outbreak of cybercrimes. As a result, how to detect the presence of identity spoofing becomes a critical issue.

Two categories of existing solutions exist to detect identity spoofing attacks, namely, active detection and passive detection. Active detection allows additional messages to be injected into the network, such as challenges and responses used in cryptographic-based schemes for user authentication. In the case that the entire node being compromised such that the cryptographic keys are exposed, location related information can be used to facilitate node authentication. For example, in [4], specific chipset, firmware or the driver of an 802.11 wireless device can be identified by watching its responses to a crafted malformed 802.11 frames. However, the downside of active detection methods lies in its requirement on extra message exchanges, which will accelerate the energy usage and consume available bandwidth. In addition, the responses can also be spoofed, if they are device dependent.

In contrast, passive detection methods extract device specific features from message transmissions, which can be categorized as time-independent and time-dependent features. The main strength is that these features are device dependent and hence can be used as an unique pattern to fingerprint a specific device. Particularly, time-independent features include clock skew (observed from message time stamps), sequence number anomalies (in MAC frames), timing (of probe frames for channel scanning), and various RF parameters (transient phases at the onset of transmissions, frequency offsets, phase offsets, I/Q offsets, etc.) [5]. Time-dependent features include radio signal strength (RSS), angle of arrival, time of arrival, differential received signal strength, frequency difference of arrival, etc. Note that, time-independent features refer to the signal measurements that have constant mean values and are only randomized by white noises across the time. Time-dependent features refer to the signal measurements whose mean values are time varying due to the essential dynamical nature.

For the fingerprinting methods based on time-independent features [3], [5]–[9], though with a variety of implementations, basically it is assumed that the features form a cluster for each device, which can be regarded as the unique fingerprinting pattern to identify the device. Two most recent works are conducted by Brik et al. [6] and Nguyen et al. [5]. Brik et al. [6] proposed the Passive RAdio-metric Device Identification System (PARADIS) utilizing modulation domain radio-metrics, such as carrier frequency error, I/Q offset, etc. Nguyen et al. [5] further proposed an unsupervised clustering method based on non-parametric Bayesian method and infinite Gaussian mixture model, which can automatically determine the number of clusters. To summarize, time-independent features can be regarded as accurate and robust wireless signatures for particular devices. However, the fingerprinting methods using time-independent features also have some limitations. For example, these features are much harder to extract. Usually, some high-end measurement devices are required to perform feature extraction. Moreover, the accuracy of these feature relies on the precision of the measurement devices. Therefore, although time-dependent features are accurate wireless signatures, the extracted features might include some errors due to the limitation of wireless measurements.

For time-dependent features, the most popular family of methods for device identification is RSS-based. In [10], a geographic location based identification against masquerading threats was employed, where two approaches are proposed: distance ratio test (DRT), which utilizes the received signal strength (RSS) of a device, and distance difference test (DDT), which relies on the received signal's relative phase difference when the signal is received at different devices. Zhao et al. [11] proposed a radio environment map (REM) which is a comprehensive database of geographical features, available services, spectral regulations, locations, and activities of radio devices and policies. Identification of cognitive radio (CR) node through an analysis of the transmitted signal is investigated in [12] where wavelet transform is utilized to identify the transmitter fingerprint. However, the RSS measurements are time varying and only provide coarse spatial resolution. Therefore, due to the dynamic nature, time-dependent features, such as RSS, cannot be regarded as an accurate and reliable signature alone.

The goal of this paper is to improve existing detection methods by considering additional features that could potentially help improve the fingerprinting performance. Studies have been shown that both time-independent features (e.g., frequency difference and phase shift difference) and time-dependent features (e.g., RSS and time difference of arrival) can be used to do spoofing detection [3], [5]–[9], [13], [14]. In this paper, we propose to concurrently model all the useful features in a unified statistical framework, based on infinite hidden Markov random field (iHMRF). All the device dependent features can be categorized into time-independent and time-dependent features. The autocorrelation on time-dependent features is captured by using the so-called Markov Property in iHMRF, in which data points that are similar on time-dependent features tend to have consistent cluster labels. The time-independent features are captured through embedded Gaussian mxitures in iHMRF. The main contributions of this work can be summarized as follows:

1) **Design of a unified fingerprinting framework.** To the best of our knowledge, this is the first statistical approach to model both time-dependent and time-independent features in a systematic framework for device fingerprinting.
2) **Formulation of the fingerprinting problem via iHMRF modeling.** We propose a novel

application of the iHMRF model to the device fingerprinting problem that captures correlations on time-dependent features using the Markov property, and correlations on time-independent features using an embedded Gaussian mixture model.

3) **Design of an online learning algorithm.** We propose a new online classification algorithm for the fingerprinting problem based on variational incremental inference, micro-clustering techniques, and batch updates.
4) **Comprehensive empirical validations.** We conducted extensive simulations on a variety of scenarios to validate the effectiveness and efficiency of our proposed techniques, competing with existing state-of-the-art methods.

The rest of the article is organized as follows. Section 2 formalizes the fingerprinting problem based on both time-dependent and time-independent features. Section 3 discusses theoretical preliminaries, including Hidden Markov Random Field (HMRF) and infinite Gaussian Mixture Model (iGMM). Section 4 formulates an infinite hidden Markov random field (iHMRF) model for the fingerprinting problem, and Section 5 presents a new incremental inference algorithm for wireless streaming environment. Empirical validations of our proposed fingerprinting framework are presented in Section 6. The paper concludes and discusses our future work in Section 8.

II. Related Work

A large body of literature has been dedicated to the issue of wireless device identification for detecting spoofing attacks. In this section, we review the most relevant work in the literature. Based on different types of features utilized, we classify these methods into two categories, including radio-metric based methods, and radio signal strength (RSS) based methods.

A. Radio-metric Based Device Fingerprinting

In [6], Brik et al. proposed the Passive RAdio-metric Device Identification System (PARADIS) utilizing modulation domain radio-metrics, such as carrier frequency error, I/Q offset, etc. The experimental results show that these device dependent radio-metrics can effectively differentiate devices. However, this method requires a training phase to collect the fingerprints of legitimate nodes. Nguyen et al. [5] further proposed an unsupervised clustering method based on non-parametric Bayesian method and infinite Gaussian mixture model. Without knowing the number of devices, this method can automatically identify different devices by clustering their emitted packets into different clusters.

Our method also builds upon a non-parametric Bayesian framework for unsupervised clustering. However, our method not only considers device dependent radio-metrics, but also takes other device independent features into consideration to greatly improve the device identification performance.

B. RSS Based Device Fingerprinting

Compared with radio-metric features, RSS feature is much easier to obtain, which makes RSS a popular feature for device fingerprinting. Faria et al. [3] demonstrated strong correlations between RSS signals and the physical location of devices, and proposed to use signalprint, a vector of RSS values measured by surrounding Access Points (APs), to identify wireless devices for detecting spoofing attacks. Sheng et al. [7] extended [3] and applied Gaussian mixture model to identify clusters of the RSS readings. Chen et al. [8] used RSS and K-means cluster analysis. In both [7] and [8], the number of clusters needs to be predefined. Later, Yang et al. [9] proposed two cluster-based mechanisms that can automatically determine cluster numbers.

However, the aforementioned methods [3], [7]–[9] only work in a static network (e.g., each device is fixed in a specific location) and may raise a large number of false alarms in a mobile network. The RSS profiles may change over time due to the nature of wireless device mobility. To capture the RSS time-dependent property, Yang et al. [13] proposed the DEMOTE system that partition the RSS trace of a node identity into two separate RSS traces, in which one trace is related to a genuine node, and the other is related to a potential attacker. If the correlation between the two traces is lower than a threshold, an alarm is alerted. They focused on two-class situations where one genuine node and one attacker share a single identity (e.g., MAC address). This solution may not be applicable to situations with multiple attackers sharing the same identity. Zeng et al. [14] proposed a reciprocal channel variation-based identification (RCVI) technique to detect spoofing attacks in mobile wireless networks. RCVI applies location de-correlation and reciprocal channel variation to detect the original devices of all packets. However, this method assumes a bidirectional communication between the genuine and the victim nodes. Therefore, it is not a completely passive detection and requires senders to send the RSS information, which may cause unnecessary network overheads.

Our paper also focuses on dynamic mobile networks. We observe that the RSS based solution for mobile networks share two more limitations. First, they assume that wireless devices and access points (AP) communicate periodically, and hence high sample rate location features (e.g., RSS, TDOA) could be extracted. Second, they consider device identification (e.g., MAC address) into their fingerprinting process. The use of forgeable user identity information (UII) may make the methods vulnerable to advanced spoofing attacks. For example, an attacker may inject packets with randomly assigned device MAC addresses into the wireless network. This attack will be hard to detect if these MAC-addresses related victim devices are evaluated separately. On contrary, our method takes the low sampling rate into consideration. In addition, we neglect the forgeable UII in our fingerprinting framework.

III. Features for Device Fingerprinting

Device fingerprinting means utilizing a set of unique features of devices that when exploited can be used to differentiate wireless devices. Fingerprinting features can be classified in several ways. For example, it can be categorize as time-dependent or time-independent features. Some of the features varies over the time, whereas the others remain unchanged. There can be device dependent and device-independent features as well. There can be transmitter fingerprinting and receiver fingerprinting. The transmitter fingerprints are different than receiver's radio-metric parameters' such as received power and are unique to the transmitter and not altered by the channel condition and receiver structure.

In this section we briefly discuss about notable features that can be exploited for iHMRF based device fingerprinting. Typically some common features of signal measurements/classifications are: angle-of-arrival (AOA), received signal strength (RSS), time-of-arrival (TOA) and frequency-of-arrival (FOA). However, sometimes difference measurement features are well suited for creating traces for particular applications. For example, time-difference-of-arrival (TDOA), frequency-difference-of-arrival (FDOA), differential received signal strength (DRSS), phase shift difference (PSD) etc.

A. *Time Measurement*

The time required for a signal to travel from the transmitter (client or node) to the receiver (anchor or access point) is directly proportional to the distance between them. The TOA and TDOA follows this principle [15]. Propagation time measurement requires synchronization between transmitter and receiver and knowledge of transmission and reception times at one position. On the other hand, time difference measurements eliminates need for node to be synchronized to anchors, but requires synchronization between anchors and doesn't directly give the distance between transmitter and receiver. The *trilateration*, conversion of the observations to distances, from TOA or TDOA is done by $d = c\tau$, where d is the distance, τ is the observed time of flight (transmit time - receive time), and c is the propagation speed. The distance (from observations) related to positions

$$d_m = \|(x, y) - (x_m, y_m)\|\,, \quad m = 1, 2, 3. \tag{1}$$

where (x, y) is the client position, (x_1, y_1), (x_2, y_2), and (x_3, y_3) are anchor positions, and $\|(x, y)\| = \sqrt{x^2 + y^2}$. Here we have three non-linear equations with two unknowns, and it can be shown that there is a single solution. Solving the equations requires more advanced algorithm unless linearization technique applied. Using two observation points, TDOA can be calculated by

$$d = d_1 - d_2 = \|(x, y) - (x_1, y_1)\| - \|(x, y) - (x_2, y_2)\|\,.$$

The key sources of errors are: 1) synchronization error for imperfect reference clock, measurement error to determine the signal's exact time of arrival and signal fading (i.e., multipath), and environmental errors (e.g., non-line-of-sight (NLOS) propagation) that adds delay not related to distance.

B. Frequency Measurement

Measuring Δf, the difference between the carrier frequency of the received signal and the one of the transmitted signal, can provide estimation about the device's whereabout. The frequency difference is a strong feature since each wireless transmitter has its own oscillator, and each oscillator creates a unique carrier frequency. Frequency shift of the received signal is related to the velocity vector of the transmitter relative to the receiver. Note that this mobility of transmitter introduces *Doppler effect* in the signal that smears signal frequency that can be measured. Frequency difference are more commonly used and obtained from Cross Ambiguity Function

$$C\left(\Delta f, \Delta t\right) = \int_0^T x\left(t\right) x^*\left(t + \Delta t\right) e^{-j\pi\Delta f t} dt. \tag{2}$$

It differs from time dependent features in that the frequency/phase shift feature observation points must be in relative motion with respect to each other and the source, and FDOA is calculated by

$$f = f_1 - f_2 = \frac{v_1}{\lambda}\cos\theta_1 - \frac{v_2}{\lambda}\cos\theta_2. \tag{3}$$

A major drawback of this feature is that a large amounts of data must be moved between observation points or to a central position to do the cross-correlation that is necessary to estimate the frequency shift. Other common source of measurement errors are: 1) imperfect frequency reference, 2) measurement errors such as noise, multipath etc., and non-stationary nature of the frequency.

C. Phase Shift Difference Measurement

On top of aforementioned method, one can differentiate devices by looking into device's I-Q phase characteristic [5], [16]. Ideally the phase shift from one constellation to a neighbor one is 180° for BPSK modulation and 90° for QPSK modulation. I-Q phase characteristics are different for I-phase and Q-phase. The constellation may deviate from original position due to hardware variability, and different devices have different constellations. Therefore, this feature can be measured and used as classifier as well. Figure 1 shows an illustrative example of device signal constellations. In this example we used QPSK as modulation of choice and considered feature extracted from the constellation of QPSK. In QPSK, four symbols with different phases are transmitted where each symbol represents two bits. Mathematically the transmitted symbol can be represented as

$$s_i\left(t\right) = \sqrt{\frac{2E_s}{T}} \cos\left(2\pi f_c t + \left(2n-1\right)\frac{\pi}{4}\right), \tag{4}$$

where E_s is transmission power, T is symbol period, f_c is carrier frequency, and n is constellation index. By changing n, we can vary the phases of signal, creating four phases $\frac{\pi}{4}$, $\frac{3\pi}{4}$, $\frac{5\pi}{4}$, and $\frac{7\pi}{4}$. In ideal case, the phase shift from one symbol to its neighbor is 90°. However, the transmitter amplifiers for I-phase and Q-phase might be different. Consequently, the degree shift can have variations.

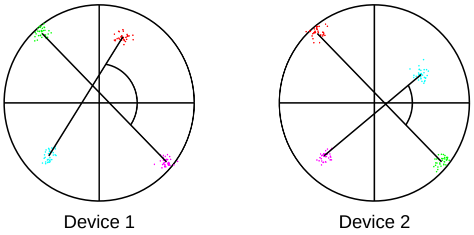


Fig. 1: Illustration of phase shift difference for constellation of QPSK symbols of two transmitters

D. Angle of Arrival Measurement

The direction of the nodes (or devices) relative to the AP (or anchor) is equal to the observed received angle-of-arrival (AOA or DOA), that can be used to create trace of device by calculating the position of the nodes, or determining the angle of the position of node relative to the access point. This process is called '*triangulation*' where a minimum of two anchors (i.e., location (x_1, y_1) and (x_2, y_2)) and reference coordinate are needed and can be calculated by two linear equations

$$\begin{aligned} y &= \tan\theta_1 x + \left(y_1 - \tan\theta_1 x_1\right), \\ y &= \tan\theta_2 x + \left(y_2 - \tan\theta_2 x_2\right), \end{aligned} \tag{5}$$

where θ are angle between device and anchor. Possible source of AOA errors are reference error (what is east?), measurement error for thermal noise, environmental error (i.e., NLOS).

E. Radio Signal Strength (RSS) Measurement

In free space signal power decays exponentially with distance that can be roughly estimated by received signal strength. Translation of RSS measurement to distance requires knowledge of the transmit power (i.e., reference value) and and knowledge of the relationship between distance and power decay (propagation model). The received signal power can be expressed as

$$P_r\left(d\right) = P_0 + 10n\log_{10}\left(\frac{d}{d_0}\right) + X_\sigma, \tag{6}$$

where P_0 is the received power at reference distance d_0 and P_r is the observed received power, d is the distances, and n is the path loss exponent. The *trilateration* from RSS is done in the same way as time measurement, except the conversion of the observations to distances is done by

$$d_0 = d10^{\left(\frac{P_0 - P_r}{10n}\right)}. \tag{7}$$

Differential RSS measurements eliminate need for transmit power knowledge and can provide improved performance in correlated shadowing. The key limitations of this feature are: 1) imperfect knowledge of the transmit power or antenna gain, 2) measurement error such as signal fading (i.e., multipath), interference, and thermal noise, and 3) environmental errors (e.g., non-line-of-sight propagation) such as shadowing, biases the resulting distance estimate, and imperfect knowledge of the propagation exponent (model error).

Interestingly, the channel gain can be used as trait as well. The amplitude of received signal is proportional to the channel gain, A_p. The general consensus is that the signals transmitted from the same device over a short duration tend to have similar amplitude or effect of channel, even though the absolute value of the amplitude is generally unknown. If the channel is Rayleigh faded multipath channel, the channel gain can be expressed as

$$A_p \cong d^{-\beta}|h|, \tag{8}$$

where $|h|$ is the fading component that is normally distributed with $\mathcal{N}(0, \sigma_h^2)$, d is distance from a transmitted device to the sensing device, β is the path loss exponent. Thus the received signal gain

A_p can be described by the distribution

$$A_p \sim \mathcal{N}(0, d^{-2\beta}\sigma_h^2). \tag{9}$$

A notable difference is that by looking into channel characteristics only does not infer the locations of devices directly, rather A_p as one more feature for the identification. Note that the aforementioned features are generic and there are other features that can be used for specific radio technologies. For example, second-order cyclostationary feature of OFDM can be used for identification.

Features	**Time Independent**	**Time Dependent**
Device Dependent	Frequency-of-arrival (FOA) I/Q Offset	Radio Signal Strength (RSS) Signal Noise Ratio (SNR)
Device Independent	Phase Shift Difference (FSD) Carrier Frequency Offset (CFO)	Time-Difference-Of-Arrival (TDOA) Time-Of-Arrival (TOA) Angle-Of-Arrival (AOA) Frequency-Difference-Of-Arrival (FDOA)

TABLE I: Device Fingerprinting Features

IV. Problem Formulation

Suppose we are given a sequence of N packet feature vectors $\{(\mathbf{x}_1, \mathbf{s}_1, t_1), \cdots, (\mathbf{x}_N, \mathbf{s}_N, t_N)\}$, where $\mathbf{x}_i \in \mathcal{R}^p$, $\mathbf{s}_i \in \mathcal{R}^d$, p and d refer to the numbers of time-independent and time-dependent features, respectively, and t_i refers to the arrival time of the ith packet on an AP. The goal is to identity the sequence of hidden states (device labels): $z_1, \cdots, z_N$, where $z_i \in \{1, 2, \cdots, C\}$ refers to the hidden state of the packet feature vector $(\mathbf{x}_i, \mathbf{s}_i, t_i)$, and C refers to the total number of hidden states. There may exist some t_is, in which the time distance between t_i and t_{i+1} is large, and the dependence between $\mathbf{s}_i$ and $\mathbf{s}_{i+1}$ may be highly degraded because of the low collection rate. The number C of hidden states is unknown and will be estimated using nonparametric Bayesian techniques.

The process of feature extraction is shown in Figure 2. Suppose multiple access points (APs) are deployed across the network environment, which collect and send traffic information to a centralized server, called a wireless appliance (WA). Each AP reports the RSS measurement for each packet received, as well as other device dependent features, such as frequency difference and phase shift difference. WA receives all the information and creates a fingerprint feature vector for each packet. Note that, there may be some duplicated features reported by APs, such as frequency differences of the repeated packets received by different APs. We will randomly select and keep one version, since for device dependent features all different versions should exhibit similar patterns.

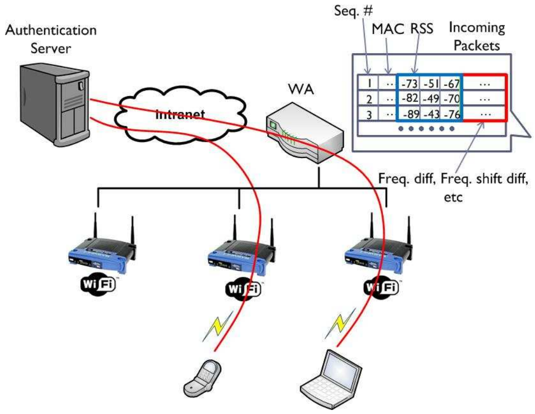


Fig. 2: Features extraction from packets

Several assumptions and constraints are stated as follows:

1) There is no training data about the fingerprints of legitimate devices available. The problem will be addressed in a completely unsupervised manner.
2) The collection rate of RSS measurements may be unstable. Sometimes the collection rate will be low, e.g., some devices are in standby status and there are no communications between the devices and access points. Sometimes the collection rate will be high, e.g., the device users are using calling services, sending text messages, or serving internet.
3) The number of clients (devices) is unknown and dynamic. Current clients may leave the network and new clients may join the network in any time.
4) A wireless network may have a large number of concurrent clients. We will need to evaluate the impact of the number of concurrent clients on the fingerprinting performance.
5) It is not allowed to add any additional out-band message exchanges. The problem will be addressed using passive detection strategies.
6) Attackers have the ability to adjust transmission powers to increase localization uncertainties.
7) Attackers have the ability to masquerade as a large number of clients. Hence, we will not trust device identity information and only consider device dependent features for fingerprinting.

V. Theoretical Backgrounds

This section introduces two basic statistical models: Hidden Markov Random Field (HMRF) and infinite Gaussian Mixture Model (iGMM). These two models provide theoretical fundamentals to Infinite Hidden Markov Random Field (iHMRF) that will be applied to do wireless device fingerprinting.

A. Hidden Markov Random Field

Suppose we have a set of observations $\{(\mathbf{x}_1, \mathbf{s}_1), \cdots, (\mathbf{x}_N, \mathbf{s}_N)\}$, where each observation $(\mathbf{x}_i, \mathbf{s}_i)$ has p features ($\mathbf{x}_i \in \mathcal{R}^p$) and d spatial coordinates ($\mathbf{s}_i \in \mathcal{R}^d$). Denote $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ and $\mathbf{S} = \{\mathbf{s}_1, \cdots, \mathbf{s}_N\}$. The objective is to infer the latent variables $\mathbf{Z} = \{z_1, \cdots, z_n\}$ based on $\mathbf{X}$ and $\mathbf{S}$, where $z_i \in \mathbb{C}$, and $\mathbb{C} = \{1, \cdots, C\}$ denotes the set of class labels.

Hidden Markov Random Field (HMRF) can be described as a two-layer hierarchical model, including the latent layer $\mathbf{Z}$ and the observation layer $\mathbf{X}$. For the latent layer $\mathbf{Z}$, HMRF considers spatial dependencies between the observations $\mathbf{Z}$. Nearby variables will have higher correlations than distant ones. The neighborhood relationship is decided based on their closeness on spatial coordinates $\{\mathbf{s}_1, \cdots, \mathbf{s}_n\}$, such as by the K-nearest neighbors rule. This Markov property can be formulated as

$$p(z_i = c|\mathbb{N}(z_i); \gamma) = \frac{1}{\mathcal{Z}(\gamma)} exp\left(-\sum_{c \in \mathbb{C}_i} V_c(z_i = c, \mathbb{N}(z_i)|\beta)\right), \tag{10}$$

where $\mathcal{Z}(\beta)$ refers to a normalization constant, β is called the inverse temperature of the model, $\mathbb{N}(z_i)$ refers to the neighbors of z_i, and $\mathbb{C}_i$ refers to the set of cliques, each of which contains z_i as a member. A clique c is defined as any set of variables such that all the variables in c are neighbors to each other. $V_c(\cdot)$ is called clique potential, which is a measure of the consistence of the variables in c. A clique potential $V_c(\mathbf{Z}|\beta)$ can be defined as

$$V_c(\mathbf{Z}|\gamma) = \beta \prod_{i,j \in c} \delta(z_i - z_j). \tag{11}$$

The joint distribution $p(\mathbf{Z}|\beta)$ of an HMRF model is

$$p(\mathbf{Z}) = \prod_i p(z_i|\mathbb{N}(z_i); \gamma) = \frac{1}{\mathcal{Z}(\gamma)} exp\left(-\sum_{c \in \mathcal{C}} V_c(\mathbf{Z}|\beta)\right), \tag{12}$$

where $\mathcal{Z}(\beta)$ is a normalization constant.

For the observation layer, HMRF defines the conditional distribution $p(\mathbf{X}|\mathbf{Z})$ as

$$\begin{aligned} p(\mathbf{X}|\mathbf{Z}; \Theta) &= \prod_{i=1}^{N} p(\mathbf{x}_i|z_i; \Theta_{z_i}), & (13) \\ p(\mathbf{x}_i|z_i; \Theta_{z_i}) &= \mathcal{N}(\mathbf{x}_i|\mu_{z_i}, \mathbf{\Sigma}_{z_i}), & (14) \end{aligned}$$

where each observation $\mathbf{x}_i$ follows a Gaussian distribution conditioned on the latent variable z_i. Each class is related to a distinct Gaussian distribution, and we have totally C Gaussian mixtures. Denote the parameters $\Theta = \{\Theta_c\}_{c=1}^{C}$, and $\Theta_c = \{\mu_c, \mathbf{\Sigma}_c\}$.

B. *Infinite Gaussian Mixture Model*

Infinite Gaussian Mixture Model (iGMM), also named Dirichlet Process Gaussian Mixture Model (DPGMM), is an extension of the traditional Gaussian Mixture Model (GMM) to support an finite number of Gaussian mixtures. Denote $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ as observations, and $\mathbf{Z} = z_1, \cdots, z_N$ as latent class labels, where $z_i \in \mathbb{C}_i = \{1, \cdots, C\}$. Note that, different from HMRF, spatial coordinates (attributes) are not considered here. The iGMM model can be defined as

$$\begin{aligned} v_c|\alpha &\sim Beta(1, \alpha), c = 1, \cdots, \infty, &(15)\\ \Theta_c|G_0 &\sim G_0, c = 1, \cdots, \infty, &(16)\\ \mathbf{x}_i|z_i = c; \Theta_c &\sim \mathcal{N}(\mu_c, \mathbf{\Sigma}_c), &(17)\\ z_i|\pi(\mathbf{v}) &\sim Multi(\pi(\mathbf{v})), &(18) \end{aligned}$$

where $\pi_c(\mathbf{v}) = v_c \prod_{i=1}^{c-1}(1 - v_i)$. To interpret this model, we can look at its data generating process:

1) Draw $v_c|\alpha \sim Beta(1, \alpha), c = \{1, 2, \cdots\}$,
2) Draw $\Theta_c = \{\mu_c, \mathbf{\Sigma}_c\}|G_0 \sim G_0, c = \{1, 2, \cdots\}$,
3) For the ith data point
   a) Draw $z_i|\{v_1, v_2, \cdots\} \sim Multi(\pi(\mathbf{v}))$,
   b) Draw $\mathbf{x}_i|z_i = c \sim \mathcal{N}(, \mu_c, \mathbf{\Sigma}_c)$.

Particularly, step 1 samples a countably infinite set of random variables $\mathbf{v}$ from a beta distribution $Beta(1, \alpha)$, where α is a hyper-parameter. The prior probabilities $\pi(\mathbf{v})$ can then be calculated as

$$\pi_c(\mathbf{v}) = v_c \prod_{i=1}^{c-1}(1 - v_i), c = 1, 2, \cdots. \tag{19}$$

Step 2 samples the model parameters Θ_c for each mixture c from a base distribution G_0, which is

$$\begin{aligned} \mathbf{\Sigma}_c &\sim InverseWishart_{v_0}(\Lambda_0), &(20)\\ \mu_c &\sim \mathcal{N}(\mu_0, \Sigma_c/K_0), &(21) \end{aligned}$$

where $\{v, \mu_0, \Lambda_0\}$ are the hyper-parameters. Steps 1 and 2 are called the stick-breaking construction of a dirichlet process (DP). Given the prior probabilities $\pi(\mathbf{v})$ and the Gaussian distribution parameters

$\{\Theta_1, \Theta_2, \cdots\}$, the last step (Step 3) is to i.i.d. sample N observations $\{\mathbf{x}_i, z_i\}$, $i = 1, 2, \cdots, N$. For each point i, step 3.1 samples its class label from $Multi(\pi(\mathbf{v}))$, and step 3.2 samples its features $\mathbf{x}_i$ from the corresponding Gaussian distribution $\mathcal{N}(\mu_c, \mathbf{\Sigma}_c)$.

VI. Infinite Hidden Markov Random Field (iHMRF)

Given the data set $\mathbf{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$, $\mathbf{S} = \{\mathbf{s}_1, \cdots, \mathbf{s}_N\}$, and $\mathbf{T} = \{t_1, t_2, \cdots, t_N\}$, with the unknown class labels $\mathbf{Z} = \{z_1, \cdots, z_N\}$. The iHMRF model can be represented by a graphical model as shown in Figure 3. Each node represents a random variable (or vector), and each dot represents a hyper-parameter. The filled nodes refer to observations and blank nodes refer to latent variables. Basically, we first use spatio-temporal features $\{(\mathbf{s}_1, t_1), \cdots, (\mathbf{s}_N, t_N)\}$ to build a neighborhood graph for the latent state variables $\{z_1, \cdots, z_N\}$, in which states z_i and z_j are connected by an undirected edge if they are spatial temporal neighbors. Each latent state variable z_i will emit an observation $\mathbf{x}_i$. The iHMRF model is designed by this manner. According to the key property of a hidden Markov random field, the hidden states should be consistent if they are neighbors to each other. However, two neighbor nodes z_i and z_j could be assigned different cluster labels if their emission observations $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to two different Gaussian distributions.

The iHMRF model can be defined as follows:

Definition 1. *Infinite Hidden Markov Random Field (iHMRF)*

$$\begin{aligned}
\alpha|\lambda_1, \lambda_2 &\sim Gamma(\lambda_1, \lambda_2) && (22)\\
\beta_c|\alpha &\sim Beta(1, \alpha), c = 1, \cdots, \infty, && (23)\\
\Theta_c|G_0 &\sim G_0, c = 1, \cdots, \infty, && (24)\\
\mathbf{x}_i|z_i = c; \Theta_c &\sim \mathcal{N}(\mu_c, \mathbf{\Sigma}_c), && (25)\\
z_i|\pi(\beta) &\sim Multi(\pi(\beta)), && (26)\\
p(\mathbf{Z}) &= \prod_{i=1}^{N} p(z_i|\pi(\beta), \{z_i, \mathbb{N}(z_i)\}), && (27)\\
p(z_i|\pi(\beta), z_i, \mathbb{N}(z_i)) &= p(z_i = c|\pi(\beta)) \\
&\quad \times p(z_i = c|z_i, \mathbb{N}(z_i); \gamma), && (28)
\end{aligned}$$

where $\Theta_c|G_0$ *stands for:*

$$\begin{aligned} \mathbf{\Sigma}_c &\sim InverseWishart_{\upsilon_0}(\Lambda_0), & (29)\\ \mu_c &\sim \mathcal{N}(\mathbf{g}_0, \Sigma_c/\eta_0), & (30) \end{aligned}$$

and

$$p(z_i = c|\mathbb{N}(z_i);\gamma) = \frac{1}{\mathcal{Z}(\gamma)} exp\left(-\sum_{c\in\mathbb{C}_i} V_c(z_i = c, \mathbb{N}(z_i);\gamma)\right), \quad (31)$$

where $\{\lambda_1, \lambda_2, \gamma, \upsilon_0, \mathbf{g}_0, \Lambda_0, \eta_0\}$ *are hyper-parameters.*

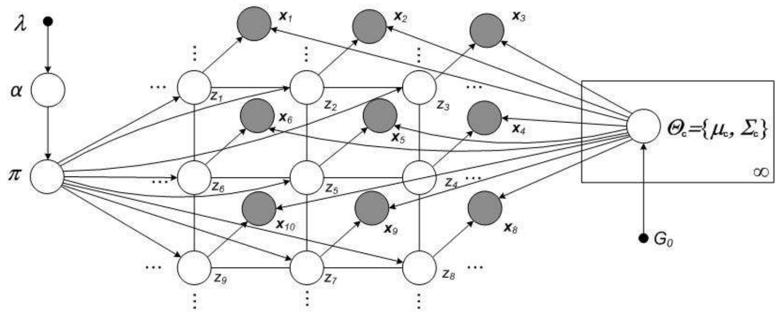


Fig. 3: Graphical Model Representation of iHMRF

Compared with HMRF and iGMM, the iHMRF model has three major advantages: First, iHMRF is able to capture Gaussian mixtures information and spatial dependencies between latent variables $\{z_i\}_{i=1}^N$ concurrently, through Equations (25) and (28). As a result, iHMRF tends to decide the value of z_i both based on its neighbors and its closest Gaussian mixture. When conflicts occur, that means the class labels of its spatial neighbors are not consistent with its closest Gaussian mixture, we can adjust the inverse temperature parameter γ to decide the weight we put on each side. A smaller value of γ implies that the model will favor more on the Gaussian mixtures information. In the extreme when $\gamma = 0$, the model will degenerate and become equivalent to iGMM. Second, iHMRF is able to automatically estimate the number of class labels (clusters), since Dirichlet Process (DP) is used as the prior distribution for z_i and $\mathbf{x}_i$. Third, iHMRF is robust to transmission power changes. When a device changes its transmission power, it tends to increase the spatial entropy and makes its spatial trajectory more highlighted than those of other devices. We observe that iHMRF inherits the advantages of both HMRF and iGMM.

Based on the above iHMRF model specification, the fingerprinting problem can be reformulated as a maximum-a-posterior (MAP) problem. It is to estimate the latent variables $\{z_1, \cdots, z_N\}$, such that their joint posterior probability based on the observations $\{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ can be maximized:

$$\{z_1, \cdots, z_N\} = \underset{\{z_1, \cdots, z_N\}}{\operatorname{argmin}} \; p(z_1, \cdots, z_N | \mathbf{x}_1, \cdots, \mathbf{x}_N). \tag{32}$$

Because the wireless device environment under study is a streaming environment, it is more appropriate to do incremental inference (or classification). We will introduce efficient incremental techniques in the next section VII.

VII. Incremental Variational Inference for the IHMRF Model

Inference for the iHMRF model can be conducted based on variational inference, Markov chain Monte Carlo (MCMC), and other methods. In this paper, we are focused on variational inference, because it is computationally more scalable than MCMC techniques, and hence more applicable to wireless streaming environment. Denote $\Phi = \{\mathbf{Z}, \Theta, \mathbf{v}\}$ as the set of all latent random variables, and $\theta = \{\gamma, \lambda_1, \lambda_2, \upsilon_0, \mathbf{g}_0, \mathbf{\Lambda}_0\}$ as the set of hyper-parameters. The objective is to infer the latent Φ given the observations $\mathbf{X}$ and hyper-parameters θ. Because it is intractable to calculate the posterior $p(\mathbf{\Phi}|\mathbf{X}, \theta)$, variance inference is applied to approximate the posterior with a parametric family of factorized distributions $q(\mathbf{\Phi}|\mathbf{X}, \theta)$ of the form

$$\begin{aligned} q(\mathbf{\Phi}|\mathbf{X}, \theta) &= q(\mathbf{Z}) q(\alpha; \lambda_1, \lambda_2) \prod_{c=1}^{C-1} q(\beta_c; \zeta_{c,1}, \zeta_{c,2}) \\ &\quad \times \prod_{c=1}^{C} q(\mu_c, \Sigma_c; \tilde{\upsilon}_c, \tilde{\eta}_c, \tilde{\mathbf{g}}_c, \tilde{\Lambda}_c). \end{aligned} \tag{33}$$

Denote the variational Free Energy functional as

$$F(q; \mathbf{X}, \theta) = \int q(\Phi; \theta) \log \frac{p(\mathbf{\Phi}|\mathbf{X}, \theta)}{q(\Phi; \theta)} d\Phi, \tag{34}$$

which is a lower bound of the original log-evidence $\ln p(\mathbf{X}|\theta)$. The optimal solution based on the parametric family can be obtained by maximizing the Free Energy functional:

$$\underset{\tilde{\theta}}{\operatorname{minimize}} \; F(q(\tilde{\theta}); \mathbf{X}, \theta), \tag{35}$$

where the variational parameters to be estimated include $\tilde{\theta} = \{\lambda_1, \lambda_2, \zeta_{c,1}, \zeta_{c,2}, \upsilon_c, \eta_c, \mathbf{g}_c, \Lambda_c, \}_{c=1}^{C}$. These parameters can be optimized iteratively by coordinate accent until convergence to a local optimum. The results have been derived by Chatzis et.al. [17].

In this section, we will focus on incremental inference, instead of the above offline inference (35). Incremental inference is more suitable for a streaming environment as existing in our device fingerprinting problem. Assume that we have a buffer bucket with a limited size (e.g., N) to store the streaming observations. When the bucket is full, it will be processed and all the observations in the bucket will be classified. Then the bucket is cleaned and is ready to accept new incoming observations. We may consider multiple buckets in the process line, such that when one bucket is being processed, other buckets are ready to store new incoming observations. Denote a bucket data as $\mathbf{B}^{(i)} = \{(\mathbf{x}_1^{(i)}, \mathbf{s}_1^{(i)}, t_1^{(i)}), \cdots, (\mathbf{x}_N^{(i)}, \mathbf{s}_N^{(i)}, t_N^{(i)})\}$, where i refers to the bucket sequence number. The incremental inference problem is to process the incoming buckets $\mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \cdots$ incrementally. We consider a similar strategy as used in iGMM [18], [19], and propose an incremental inference framework for iHMRF. The key components are summarized as follows:

1) Compression Phase: When the observations have been classified to different clusters, each cluster is separated into a number of microclusters that tend to have consistent cluster labels, even when the clusters have been reformed due to the process of new bucket data. For each microcluster, its sufficient statistics are stored and the data points inside are discarded to save memory space and improve computational efficiency.
2) Model Building Phase: The incremental inference will be conducted based on microclusters, instead of data points. Some microclusters are allowed to be isolated data points.
3) Incremental Batch Update Phase: The incremental model updates based on the new bucket and previous buckets need not to start from scratch. The model information estimated based on previous buckets will be considered to improve the incremental update efficiency.

The technical details of the above three components are discussed in Sections VII-A, VII-B, and VII-C, respectively.

A. Model Building Phase

This phase assumes that the observations in the current buckets have already been grouped to a set of microclusters. When this phase is first run (as the initialization step), each observation will be regarded a microcluster. For later iterations, the microclusters are generated from the previous iterations (see Section VII-B). Denote A as a specific microcluster, n_A as the cluster size, and $\mathbf{x}_A = \frac{1}{n_A} \sum_{x_i \in A} x_i$.

The model building phase is to solve the following constrained optimization problem

$$\begin{aligned} &\underset{q(\Phi;\theta)}{\text{minimize}} \quad \int q(\Phi;\theta)\log\frac{p(W|X;\theta)}{q(\Phi;\theta)}d\Phi \\ &\text{subject to} \quad q(z_i) = q(z_j), if\ \exists A\ s.t.\ z_i, z_j \in A, \end{aligned} \tag{36}$$

where $q(\Phi;\theta)$ is a factorized parametric form as defined in 33. Notice the difference the above problem (36) and the traditional offline problem (35). New constraints are defined such that the data points in a same microcluster must have identical class labels. Because each microcluster is now summarized by its sufficient statistics, the computational efficiency is greatly improved. The above problem can be optimized iteratively by coordinate accent until convergence to a local optimum. The solution for each iteration can be obtained as

$$\zeta_{c,1} = 1 + \sum_A n_A q(A = c) \tag{37}$$

$$\zeta_{c,2} = \langle\alpha\rangle + \sum_{k=c+1}^{C}\sum_A n_A q(A = k) \tag{38}$$

$$w_c = \sum_A n_A q(A = c) \tag{39}$$

$$\bar{\mathbf{x}}_c = \frac{\sum_A n_A q(A = c)\mathbf{x}_A}{w_c} \tag{40}$$

$$\Xi = \sum_A n_A q(A = c)(\mathbf{x}_A - \bar{\mathbf{x}}_c)(\mathbf{x}_A - \bar{\mathbf{x}}_c)^T \tag{41}$$

$$q(A = c) \propto p(A = c|(N)(A);\gamma)\tilde{\pi}_c(\beta)\tilde{p}(\mathbf{x}_A|\Theta_c), \tag{42}$$

where $\mathbb{N}(A)$ refers to the neighbors of the micro-cluster A, which are defined similar to those based on data points. Here, we use the spatial center point of a microcluster to represent its spatial location, with $\mathbf{s}_A = \frac{1}{n_A}\sum_{\mathbf{s}_i \in A}\mathbf{s}_i$, and use the center time to represent its time domain location, with $t_A = \frac{1}{n_A}\sum_{t_i \in A} t_i$. Note that, only the solution components that are different from the traditional offline solution are presented above. Readers are referred to [17] for the estimation of the other model parameters that have the same result as the offline iHMRF model, including $\tilde{\zeta}_c, \tilde{\Lambda}_c, \tilde{\upsilon}_c$, $\tilde{\eta}_c$, and $\tilde{\mathbf{g}}_c$.

B. Compression Phase

This phase focuses on the generation of microclusters. The microclusters will be generated such that the data points in each microcluster tend to be located in a same cluster, even when the overall clusters have been reformed due to the process of new bucket data. To address this challenge, a straightforward strategy is to generate multiple candidate clusters from different ways and then look for the micoclusters, each of which never overlaps with more than one candidate cluster concurrently. However, this strategy has two potential deficiencies: First, it is computationally expensive since the number of different groups increases exponentially with the data size; Second, it does not consider the behavior of future data points. An optimized strategy is to predict up to Δ future points based on the empirical distribution estimated from existing data ($\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T$):

$$\tilde{p}(\mathbf{x_{T+1}}, \cdots, \mathbf{x}_{T+\Delta}) = \prod_{i=T+1}^{T+\Delta} \frac{1}{T} \sum_{t=1}^{T} \delta(\mathbf{x}_i - \mathbf{x}_t). \tag{43}$$

We define a modified Free Energy functional by taking expectation on Δ unobserved future points as

$$\begin{aligned} \tilde{F}(q; \mathbf{X}, \theta) &= \int d\mathbf{x_{T+1}}, \cdots, d\mathbf{x}_{T+\Delta} F(q; \mathbf{X}, \theta) \\ &\quad \cdot \tilde{p}(\mathbf{x_{T+1}}, \cdots, \mathbf{x}_{T+\Delta}). \end{aligned} \tag{44}$$

The solution by maximizing the above modified Free Energy functional can be obtained as

$$\zeta_{c,1} = 1 + (1 + \frac{\Delta}{T}) \sum_A n_A q(A = c) \tag{45}$$

$$\zeta_{c,2} = \langle \alpha \rangle + (1 + \frac{\Delta}{T}) \sum_{k=c+1}^{C} \sum_A n_A q(A = k) \tag{46}$$

$$w_c = (1 + \frac{\Delta}{T}) \sum_A n_A q(A = c) \tag{47}$$

$$\bar{\mathbf{x}}_c = (1 + \frac{\Delta}{T}) \frac{\sum_A n_A q(A = c) \mathbf{x}_A}{w_c} \tag{48}$$

$$\begin{aligned} \Xi &= (1 + \frac{\Delta}{T}) \sum_A n_A q(A = c)(\mathbf{x}_A - \bar{\mathbf{x}}_c) \\ &\quad (\mathbf{x}_A - \bar{\mathbf{x}}_c)^T \end{aligned} \tag{49}$$

$$q(A = c) \propto p(A = c | (N)(A); \gamma) \tilde{\pi}_c(\beta) \tilde{p}(\mathbf{x}_A | \Theta_c), \tag{50}$$

To conduct the compression phase, we first apply the model building phase to generate clusters. Then for each candidate cluster, we split it into two clusters along its principal component, and refine the clusters based on the above update rules 45, until convergence. The gain on the free energy function is denoted as $\Delta\tilde{F}(q; \mathbf{X}, \theta)$. The cluster with the largest $\Delta\tilde{F}(q; \mathbf{X}, \theta)$ will be selected as the final splitting cluster. Iterate the process until convergence, e.g., the gain $\Delta\tilde{F}(q; \mathbf{X}, \theta)$ is smaller than a predefined threshold or the consumed memory is greater than the memory space limit.

C. Incremental Batch Update Phase

This phase assumes that all previous bucket data have been processed, and we have obtained the estimate variational parameters $\{\eta_c, \Lambda_c, \upsilon_c, \mathbf{g}_c, \zeta_{c,1:2}, \lambda_{1:2}, w_c, \bar{\mathbf{x}}_c, \Xi_c\}_{c=1}^{C}$. Suppose a new bucket data have been arrived, and it is necessary classify the new bucket data points and update all existing clusters. Denote the new bucket data as $\{(\mathbf{x}_1^{(n)}, \mathbf{s}_1^{(n)}, t_1^{(n)}), \cdots, (\mathbf{x}_N^{(n)}, \mathbf{s}_N^{(n)}, t_N^{(n)})\}$. The incremental Batch update phase can be described as

$$\tilde{\zeta}_{c,1} = \zeta_{c,1} + \sum_{i=1}^{N} q(z_i^{(n)} = c) \tag{51}$$

$$\tilde{\zeta}_{c,2} = \zeta_{c,2} + \sum_{k=c+1}^{C} \sum_{i=1}^{N} q(z_i^{(n)} = k) \tag{52}$$

$$\tilde{w}_c = w_c + \sum_{i=1}^{N} q(z_i^{(n)} = c) \tag{53}$$

$$\bar{\mathbf{x}}_c = \frac{\bar{\mathbf{x}}_c w_c + \sum_{i=1}^{N} q(z_i^{(n)} = c)\mathbf{x}_i^{(n)}}{\tilde{w}_c} \tag{54}$$

$$\begin{aligned}\tilde{\Xi} = {} & \Xi + \sum_{i=1}^{N} q(z_i^{(n)} = c)(\mathbf{x}_i^{(n)} - \bar{\mathbf{x}}_c) \\ & (\mathbf{x}_i^{(n)} - \bar{\mathbf{x}}_c)^T\end{aligned} \tag{55}$$

$$q(z_i^{(n)} = c) \propto p(z_i^{(n)} = c|\mathbb{N}(z_i))\tilde{\pi}_c(\beta)\tilde{p}(\mathbf{x}_i^{(n)}|\Theta_c). \tag{56}$$

The basic idea is to apply Equation (56) to estimate $q(z_i^{(n)})$, and apply Equations (51) to (55) to update the variational parameters $\tilde{\zeta}_{c,1:2}$, $\tilde{w}_c$, $\bar{\mathbf{x}}_c$, and $\tilde{\Xi}$. The other parameters that are consistent with the offline iHMRF model are then updated by the equations derived in [17].

VIII. Simulation Result

This section presents an extensive simulation study to validate the effectiveness and efficiency of our proposed techniques, compared with existing solutions, such as Gaussian Mixture Model (GMM) and infinite Gaussian Mixture model (iGMM) [5]. For our fingerprinting framework, we studied the performances of two inference algorithms, including the offline variational inference algorithm [17] and our proposed online (incremental) inference algorithm.

A. Simulation Setup

The simulation data generator includes two components. The first component is the generation of time-independent features. The same simulator design as used in [5] were applied to generate time-independent features. Basically, a number of devices will be chosen randomly in an area of 40×40 in the time-independent feature space, with variances of the clusters chosen random in the range from 0 to 1. We considered two time-independent features, such that the data can be easily visualized. The second component is the generation of time dependent features. We considered RSS features and assumed that the collected RSS features have been triangulated to three dimensional spatial coordinates. This is appropriate for mobile devices, because for different time periods users may travel to different spatial regions and different Access Points (AP) will be able to collect the related RSS traces data. By converting the RSS features to spatial coordinates, we do not need to consider the issue of missing values for different access points. We used UdelModels to generate mobile device traces data, which is a widely used simulator for generating human trajectory data [20]. Changes of transmission power were simulated by shifting a trace segment with a randomly selected distance and direction.

We considered four major metrics to evaluate the effectiveness of our framework, including precision, recall, F-measure, and rand index (IR). These metrics are defined based on true positive rate (TP), false positive rate (FP), false native rate (FN), and true negative rate (TN), as interpreted in Table II. These metrics are defined as $Precision = TP/(TP+FP); Recall = TP/(TP+FN); F-Measure = \frac{Precision \times Recall}{Precision + Recall}$; and $Rand - Index(RI) = \frac{TP+TN}{TP+TN+FN+FP}$.

TABLE II: Definition of TP, FP, FN, and TN

	Same Cluster	Different Clusters
Same Class	TP	FN
Different Classes	FP	TN

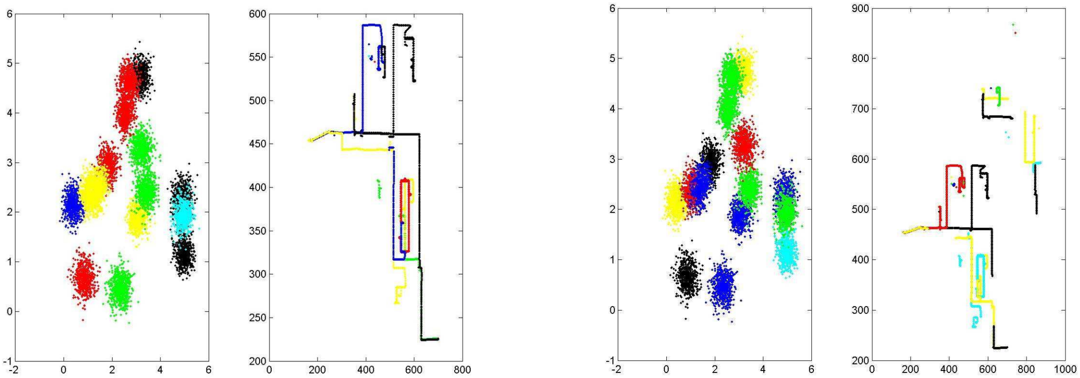
We used UdelModels to generate twelve simulation datasets to cover a variety of scenarios, including indoor and outdoor environments. The basic features of these data sets are summarized in the following table III. For each setting, we generated five different realizations, in order to calculate the uncertainty (standard deviation) of the classification performance. For all our comparison results, we reported the mean and standard deviation values for each method, in order to mitigate potential random effects.

Figure 4 shows the spatial distributions of two simulation datasets under different scenarios, with two time-independent and two time-dependent features. For both datasets, the wireless device carriers are all pedestrians, but the left one has a stable RSS sample rate, and the right one has an unstable sample rate. Each dataset has 15 clusters (devices). For each device, we generated a sequence of 8000 time-stamped observations at 1-minute interval. For each observation, two time-independent features and two time-dependent features were generated, as illustrated in the left and right plots in Figure 4 (a). As shown from the plots, the clusters are not well separable and have overlaps in the time-independent feature space (see the left plot), and also not well separable in the time-dependent feature space (see the right plot). If these features are not clustered jointly, it can be seen that it is very difficult to differentiate different clusters. However, as shown in our followup experiments, by jointly considering all the features into the clustering (fingerprinting) process, our approach significantly improves the accuracy of the clustering (fingerprinting) process. Note that, because there are no enough colors that can be used to display 15 different clusters, in our visualization in Figure 4, we randomly selected a color for each cluster. Therefore, it is observed that some clusters happened to have the same color, but we demonstrate that the clusters are not well separable from each other without any ambiguity.

TABLE III: Simulation Data Settings

Description	# of Penetrations (Peds)	# of Cars
1 Building 10 Floors	5, 10, 15	5, 10, 15
Real City (Chicago9B1k)	5, 10, 15	5, 10, 15

We compared our framework with two existing approaches, including GMM and iGMM. For our framework, we employed two inference algorithms, including the offline variational inference algorithm for iHMRF [17], abbreviated as iHMRF-VI, and our proposed incremental inference algorithm, abbreviated as Inc-iHMRF-VI. For GMM, it is required to predefine the number of clusters. In our simulation study, we set the value as the true number of clusters (devices), in order to study the best performance that a GMM model could achieve. iGMM is a nonparametric method. Although it still needs to set the number of clusters, iGMM is able to automatically determine the number of clusters. Therefore, we randomly set the initial cluster number. All the other hyperparameters were set such that the corresponding parameters are uniform-distributed. Similar strategies were used for the nonparametric methods iHMRF-VI and Inc-iHMRF-VI. One more setting in both iHMRF and



(a) Chicago9B1k Data with Only Pedestrians

(b) Chicago9B1k Data with Unstable RSS Rates

Fig. 4: Spatial Distribution of Simulation Data

Inc-iHMRF-VI is to define spatio-temporal neighborhood relationships. We defined neighbors as the data points that are 5 nearest spatial neighbors to each other and have the time stamp distance smaller than 50. These settings can be loosely decided and we observed that the resulting performance is not rapidly varied. We set the memory bound and the bucket size of Inc-iHMRF-VI to 2000 and 2000, respectively. We observed similar patterns based on different settings of these two parameters.

TABLE IV: Simulation Results Based on UdelModels with 1 Building 10 Floors

Methods	# of Devices	Precision	Recall	F-Measure	Relative Index (RI)
iHMRF-VI	5	**0.97** (0.02)	0.91 (0.13)	**0.93** (0.07)	**0.96** (0.04)
	10	0.72 (0.13)	0.81 (0.13)	0.76 (0.11)	**0.93** (0.04)
	15	**0.73** (0.09)	**0.82** (0.06)	**0.77** (0.07)	**0.96** (0.01)
Inc-iHMRF-VI	5	0.88 (0.10)	**0.94** (0.05)	0.91 (0.05)	0.95 (0.02)
	10	0.65 (0.28)	**0.85** (0.14)	0.72 (0.23)	0.90 (0.09)
	15	0.51 (0.13)	0.79 (0.08)	0.62 (0.12)	0.92 (0.02)
iGMM-VI	5	0.86 (0.09)	0.44 (0.15)	0.57 (0.15)	0.80 (0.09)
	10	**0.73** (0.11)	0.43 (0.10)	0.54 (0.10)	0.91 (0.03)
	15	0.56 (0.01)	0.30 (0.07)	0.38 (0.06)	0.92 (0.01)
GMM-EM	5	0.91 (0.15)	0.85 (0.22)	0.86 (0.16)	0.90 (0.10)
	10	0.72 (0.14)	0.83 (0.13)	**0.77** (0.11)	**0.93** (0.04)
	15	0.64 (0.11)	0.77 (0.06)	0.70 (0.08)	0.94 (0.01)

TABLE V: Simulation Results Based on UdelModels - Chicago9Blk - with Pedestrians and Cars

Methods	# of Devices	Precision	Recall	F-Measure	Relative Index (RI)
iHMRF-VI	5 Peds, 5 Cars	**0.99** (0.01)	0.98 (0.01)	**0.99** (0.01)	**0.99** (0.01)
	10 Peds, 10 Cars	**0.91** (0.10)	**0.99** (0.10)	**0.95** (0.05)	**0.99** (0.01)
	15 Peds, 15 Cars	**0.90** (0.09)	**0.97** (0.02)	**0.94** (0.05)	**0.99** (0.01)
Inc-iHMRF-VI	5 Peds, 5 Cars	0.98 (0.02)	**1.00** (0.00)	**0.99** (0.01)	**0.99** (0.01)
	10 Peds, 10 Cars	0.80 (0.13)	0.97 (0.04)	0.87 (0.08)	0.96 (0.02)
	15 Peds, 15 Cars	0.57 (0.07)	0.92 (0.08)	0.70 (0.07)	0.93 (0.02)
iGMM-VI	5 Peds, 5 Cars	0.90 (0.12)	0.29 (0.05)	0.44 (0.07)	0.80 (0.06
	10 Peds, 10 Cars	0.67 (0.08)	0.31 (0.06)	0.42 (0.06)	0.89 (0.02)
	15 Peds, 15 Cars	0.63 (0.06)	0.29 (0.06)	0.40 (0.06)	0.92 (0.01)
GMM-EM	5 Peds, 5 Cars	0.92 (0.13)	0.89 (0.06)	0.89 (0.07)	0.95 (0.03)
	10 Peds, 10 Cars	0.69 (0.08)	0.79 (0.11)	0.73 (0.09)	0.93 (0.03)
	15 Peds, 15 Cars	0.69 (0.12)	0.78 (0.06)	0.72 (0.08)	0.95 (0.02)

TABLE VI: Simulation Results Based on UdelModels - Chicago9Blk - with Only Cars

Methods	# of Devices	Precision	Recall	F-Measure	Relative Index (RI)
iHMRF-VI	5 Cars	**0.95** (0.03)	0.59 (0.08)	0.72 (0.06)	0.89 (0.02)
	10 Cars	**0.83** (0.09)	0.55 (0.05)	0.66 (0.05)	0.93 (0.01)
	15 Cars	0.68 (0.08)	0.53 (0.09)	0.59 (0.08)	0.94 (0.01)
Inc-iHMRF-VI	5 Cars	0.89 (0.12)	**0.98** (0.02)	**0.93** (0.02)	**0.97** (0.04)
	10 Cars	0.73 (0.11)	0.77 (0.09)	0.75 (0.06)	0.93 (0.02)
	15 Cars	0.56 (0.08)	**0.83** (0.06)	0.66 (0.07)	0.93 (0.02)
iGMM-VI	5 Cars	0.82 (0.08)	0.30 (0.07)	0.44 (0.07)	0.82 (0.02)
	10 Cars	0.65 (0.10)	0.32 (0.07)	0.43 (0.08)	0.89 (0.01)
	15 Cars	0.55 (0.06)	0.29 (0.05)	0.38 (0.05)	0.92 (0.01)
GMM-EM	5 Cars	0.91 (0.12)	0.87 (0.13)	0.89 (0.12)	0.95 (0.05)
	10 Cars	0.79 (0.07)	**0.81** (0.09)	**0.89** (0.08)	**0.95** (0.02)
	15 Cars	**0.73** (0.04)	0.79 (0.09)	**0.76** (0.06)	**0.96** (0.01)

B. Comparisons on Precision, Recall, and F-Measure

For the simulation data, we considered two scenarios, including indoors and outdoors. For indoors, we generated simulation data with the number of devices 5, 10, and 15, and the sample rate one reading every 20 seconds. The results are shown in Table IV. For outdoors, we simulated mobile traces of a real downtown area in Chicago with 5, 10, and 15 penetrations. The results are shown in Table V. The results on the scenarios with 5, 10, and 15 cars are shown in table VI. Table VII shows the results with concurrent pedestrians and cars. From all these results, we observe that our framework based on the iHMRF model outperformed GMM and iGMM in the majority of cases, especially compared with iGMM. Recall that the GMM method used the true number of clusters (devices) as the initial setting. Its according performance should represent the close-to-the-best performance of general clustering algorithms based on time-independent features.

However, we did notice that as shown in table VI, when the mobile devices are vehicles, the GMM's performance was comparable to our methods. But our methods still outperformed iGMM. This pattern is potentially related to the assumption of the iHMRF model. That is, data points that are spatially and temporally close tend to have consistent class labels. Vehicles are moving mush faster than pedestrians and tend to have lower sample rates and have more overlaps on their spatial traces. When devices have more overlaps spatially and temporally, the overlapped spatial trace features can not be well used to distinguish different mobile devices anymore. However, there still exist some trace segments that are not overlapped together, which can be regarded as useful information for the classification process. It potentially explains why iHMRF's performance was degraded in this situation but still performed better than iGMM.

TABLE VII: Simulation Results Based on UdelModels - Chicago9Blk - with Only Pedestrians

Methods	# of Devices	Precision	Recall	F-Measure	Relative Index (RI)
iHMRF-VI	5 Peds	**0.98** (0.04)	0.83 (0.13)	**0.90** (0.09)	**0.96** (0.03)
	10 Peds	**0.92** (0.08)	0.80 (0.13)	**0.85** (0.10)	**0.97** (0.02)
	15 Peds	**0.91** (0.05)	0.86 (0.05)	**0.88** (0.04)	**0.98** (0.00)
Inc-iHMRF-VI	5 Peds	0.86 (0.10)	**0.92** (0.08)	0.88 (0.07)	0.95 (0.03)
	10 Peds	0.71 (0.08)	**0.89** (0.07)	0.79 (0.05)	0.95 (0.03)
	15 Peds	0.61 (0.08)	**0.92** (0.02)	0.72 (0.06)	0.95 (0.01)
iGMM-VI	5 Peds	0.82 (0.12)	0.31 (0.05)	0.44 (0.06)	0.85 (0.01)
	10 Peds	0.73 (0.11)	0.36 (0.08)	0.48 (0.10)	0.92 (0.01)
	15 Peds	0.63 (0.05)	0.35 (0.06)	0.45 (0.05)	0.94 (0.00)
GMM-EM	5 Peds	0.73 (0.15)	0.90 (0.07)	0.80 (0.11)	0.91 (0.05)
	10 Peds	0.69 (0.12)	0.84 (0.09)	0.75 (0.11)	0.94 (0.03)
	15 Peds	0.68 (0.11)	0.86 (0.04)	0.75 (0.08)	0.96 (0.02)

In overall all, both iHMRF-VI and Inc-iHMRF-VI achieved comparable accuracies, but iHMRF-VI performed slightly better. This can be interpreted as the results of data compression by the use of microclusters in Inc-iHMRF-VI. For all the simulation data sets, the average data size is around 8000 observations. In our implementation, we set the memory bound to 2000 observations. That means, we compressed 8000 observations into 2000 microclusters, which greatly reduced the computational cost and the required memory size, but with slight sacrifices of the accuracy.

C. Impacts of Instable RSS Collection Rates

We evaluated the impacts of instable RSS collection rates baesd on the ChicagoBlk pedestrians data set. We randomly selected 50 percent of devices, segmented each selected device trace into eight segments, and then randomly removed 50 percent of the segments. This process leads to discontinuous RSS trace data. The classification results based on those modified data are shown in table VIII, and a visualization of the generated simulation data is shown in Figure 4. We observe that iHMRF-VI and Inc-iHMRF-VI performed the best in the majority of cases, which is consistent with our observations in previous results. However, by comparing Table VIII and Table VIII, we observe that unstable RSS rates slightly degraded the accuracies. This is potentially due to the reduction of samples size, since we have removed 50 percent of observations from 50 percent randomly selected devices. However, as long as each segment is still composed of spatial and temporally adjacent data points, the iHMRF model can be applied to capture the corresponding autocorrelations.

D. Impacts of Transmission Power Changes

Studies have shown that attackers may hide their actual locations by periodically changing the transmission powers of their mobile devices [21]. To simulate this behavior, we used the ChicagoBlk pedestrians data set. Fifty percent of devices were selected, the trace of each selected device was segmented into 8 same length, and fifty percent of these pieces were shifted to random directions with random spatial distances. The corresponding classification results are shown in Table IX. We observe that the changes of transmission power did not have significant impacts on the accuracies. One potential interpretation is that the changes of transmission power will increase the spatial entropy, making the devices' corresponding traces more separated from other traces. This will reduce the potential overlaps between device traces, and could even help improve the accuracies of iHMRF-VI and Inc-iHMRF-VI.

E. Comparison on Time Costs

We evaluated the time costs of the four algorithms on three data sets, including ”1 Building 10 Floors” (7224 observations), ”Chicago9B1k with 10 Pedestrians and 10 Cars” (4525 observations), and ”Chicago9B1k with 10 Pedestrians” (6000 observations). We set bucket size to 2000. That means, the data will be processed bucket by bucket, 2000 observations each time. The results are summarized in Figure 5. The X axis refers to the titles of the three data sets and the Y axis refers to running

TABLE VIII: Unstable RSS Rates (UdelModels - Chicago9Blk - with Only Pedestrians)

Methods	# of Devices	Precision	Recall	F-Measure	Relative Index (RI)
iHMRF-VI	5 Peds	**0.91** (0.11)	0.77 (0.08)	0.83 (0.05)	0.93 (0.02)
	10 Peds	**0.96** (0.05)	0.82 (0.11)	**0.88** (0.08)	**0.98** (0.02)
	15 Peds	**0.84** (0.10)	0.83 (0.07)	**0.83** (0.07)	**0.98** (0.01)
Inc-iHMRF-VI	5 Peds	**0.91** (0.17)	0.88 (0.15)	**0.89** (0.15)	**0.97** (0.04)
	10 Peds	0.77 (0.13)	**0.86** (0.09)	0.81 (0.11)	0.95 (0.03)
	15 Peds	0.62 (0.10)	**0.92** (0.02)	0.73 (0.07)	0.95 (0.02)
iGMM-VI	5 Peds	0.82 (0.13)	0.32 (0.07)	0.46 (0.09)	0.83 (0.02)
	10 Peds	0.71 (0.10)	0.31 (0.05)	0.43 (0.07)	0.91 (0.01)
	15 Peds	0.62 (0.09)	0.33 (0.06)	0.43 (0.06)	0.94 (0.01)
GMM-EM	5 Peds	0.75 (0.16)	**0.90** (0.06)	0.81 (0.10)	0.90 (0.06)
	10 Peds	0.67 (0.08)	0.81 (0.07)	0.73 (0.04)	0.93 (0.02)
	15 Peds	0.71 (0.04)	0.82 (0.06)	0.76 (0.03)	0.96 (0.00)

TABLE IX: Change of Transmission Power (UdelModels - Chicago9Blk - with Only Pedestrians)

Methods	# of Devices	Precision	Recall	F-Measure	Relative Index (RI)
iHMRF-VI	5 Peds	**0.98** (0.02)	0.70 (0.07)	0.82 (0.06)	**0.94** (0.02)
	10 Peds	**0.95** (0.06)	0.77 (0.06)	**0.85** (0.06)	**0.97** (0.01)
	15 Peds	**0.93** (0.04)	0.79 (0.05)	**0.85** (0.02)	**0.98** (0.00)
Inc-iHMRF-VI	5 Peds	0.76 (0.14)	**0.98** (0.03)	**0.85** (0.09)	0.93 (0.04)
	10 Peds	0.74 (0.12)	**0.88** (0.08)	0.80 (0.09)	0.96 (0.02)
	15 Peds	0.58 (0.08)	**0.86** (0.69)	0.69 (0.06)	0.95 (0.01)
iGMM-VI	5 Peds	0.83 (0.13)	0.31 (0.05)	0.45 (0.06)	0.85 (0.02)
	10 Peds	0.72 (0.11)	0.35 (0.07)	0.47 (0.09)	0.92 (0.01)
	15 Peds	0.65 (0.07)	0.35 (0.04)	0.45 (0.05)	0.94 (0.01)
GMM-EM	5 Peds	0.74 (0.11)	0.89 (0.04)	0.81 (0.07)	0.91 (0.04)
	10 Peds	0.63 (0.13)	0.83 (0.08)	0.71 (0.11)	0.93 (0.03)
	15 Peds	0.69 (0.04)	0.85 (0.04)	0.76 (0.02)	0.96 (0.00)

duration (seconds). We can observe that our proposed incremental inference algorithm Inc-iHMRF-VI is much more efficient than the offline inference algorithm iHMRF-VI. Our algorithm Inc-iHMRF-VI is even faster than iGMM. This indicates an significant improvement on the computational efficiency. The savings on time cost by Inc-iHMRF-VI will become greater when the data size increases. Note that, GMM has the lowest time cost. However, since GMM does not need to automatically estimate the number of clusters. Its time complexity should be much smaller than iGMM and iHMRF.

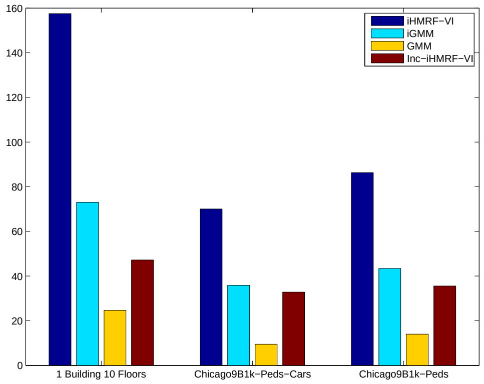


Fig. 5: Comparison on Time Costs (Seconds)

F. A Case Study on Detecting Masquerade Attacks

This section presents a case study on masquerade attacks detection, which is one of the most dangerous attack types. A masquerade attack refers to the attacking behavior where an attacker impersonates an authorized user of a system by using a faked identity (e.g., MAC address) in order to gain access to unauthorized personal resources. In order to simulate this attack behavior, we used the ChicagoBlk pedestrians data set and the 1-Building-10-Floors data set, and randomly selected k clusters and set their cluster identities into an identical cluster identity. By using fingerprinting techniques, this type of attackers can be identified if we discover that multiple clusters share the same identity information. Here k refers to the number of masquerade devices. We considered different settings of k, from 3 to 6, and evaluated the related detection rates based on different detection methods. The results are summarized in Table **??**. The results indicate that our framework (by either iHMRF-VI or Inc-iHMRF-VI) achieved the highest detection rate in most cases. The GMM method performed slightly than Inc-iHMRF-VI and iHMRF-VI. However, here we used the true number of clusters as the initial setting for the GMM method. In real applications, where the actual number is unknown, the GMM method will perform much worse.

IX. Conclusion and Future Works

Device fingerprinting is a fundamental problem for wireless network security. Passive fingerprinting techniques are effective since they are designed based on device-dependent features (e.g., RSS, AOD, and TOA) that attackers can not manipulate. However, existing solutions can only support either time-dependent or time-independent features, but no methods can handle both. This paper presents the first

Peds	Cars	Att.	iHMRF	Inc-iHMRF	iGMM	GMM
#	#	#	-VI	-VI		
10	10	3	**0.98**	0.81	0.46	0.76
10	10	4	**0.97**	0.84	0.55	0.81
10	10	5	**0.97**	0.87	0.62	0.84
10	10	6	**0.97**	0.88	0.67	0.86
15	15	3	**0.97**	0.86	0.62	0.85
15	15	4	**0.97**	0.85	0.62	0.85
15	15	5	**0.97**	0.86	0.64	0.86
15	15	6	**0.97**	0.87	0.66	0.87

TABLE X: Detection Rates for Masquerade Attacks Based on UdelModels - Chicago9B1k - Pedestrains

Peds	Cars	Att.	iHMRF	Inc-iHMRF	iGMM	GMM
#	#	#	-VI	-VI		
10	0	3	0.71	**0.86**	0.44	0.76
10	0	4	0.85	0.89	0.72	**0.94**
10	0	5	0.89	**0.91**	0.73	0.88
10	0	6	0.93	**0.98**	0.87	0.95
15	0	3	**0.80**	0.60	0.36	0.68
15	0	4	0.85	0.86	0.56	**0.87**
15	0	5	**0.88**	0.86	0.65	**0.88**
15	0	6	**0.95**	0.91	0.78	0.93

TABLE XI: Detection Rates for Masquerade Attacks on UdelModels - Chicago9B1k - 1 Building 10 Floors

unified fingerprinting approach based on infinite hidden Markov random field (iHMRF). It is able to model both time-independent and time-dependent features concurrently and is able to automatically detect the number of devices. We present a novel incremental classification algorithm that is suitable for a streaming environment with limited memory and computational resources. Extensive numerical analysis further validated the effectiveness and efficiency of our proposed approach. For our future work, we are planning to evaluate the performance of our proposed approach in real life devices. We will also extend our approach to handle other related wireless security problems, such as the identification of primary and secondary users to prevent dynamic spectrum access and malicious behavior attacks in cognitive radio networks.

References

[1] T. M. Gil and M. Poletto, "Multops: a data-structure for bandwidth attack detection," in *Proceedings of the 10th conference on USENIX Security Symposium - Volume 10*, ser. SSYM'01. Berkeley, CA, USA: USENIX Association, 2001, pp. 3–3.

[2] J. R. Douceur, "The sybil attack," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS '01. London, UK: Springer-Verlag, 2002, pp. 251–260.

[3] D. B. Faria and D. R. Cheriton, "Detecting identity-based attacks in wireless network using signalprints," in *Proceedings of the 2006 ACM Workshop on Wireless Security (WiSe '06)*. ACM Press, September 2006, pp. 43–52.

[4] S. Bratus, C. Cornelius, D. Kotz, and D. Peebles, "Active behavioral fingerprinting of wireless devices," in *Proceedings of the first ACM conference on Wireless network security*, ser. WiSec '08. New York, NY, USA: ACM, 2008, pp. 56–61.

[5] N. T. Nguyen, G. Zheng, Z. Han, and R. Zheng, "Device fingerprinting to enhance wireless security using nonparametric bayesian method." in *INFOCOM*. IEEE, 2011, pp. 1404–1412.

[6] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Mobicom*, 2008.

[7] Y. Sheng, K. Tan, G. Chen, D. Kotz, and A. Campbell, "Detecting 802.11 mac layer spoofing using received signal strength." in *INFOCOM*. IEEE, 2008, pp. 1768–1776.

[8] Y. Chen, W. Trappe, and R. P. Martin, "Detecting and localizing wireless spoofing attacks," in *Proceedings of the Fourth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*,2007, pp. 193–202.

[9] J. Yang, Y. Chen, W. Trappe, and J. Cheng, "Determining the number of attackers and localizing multiple adversaries in wireless spoofing attacks." in *INFOCOM*. IEEE, 2009, pp. 666–674.

[10] R. Chen and J. Park, "Ensuring trustworthy spectrum sensing in cognitive radio networks," in *1st IEEE Workshop on Networking Technologies for Software Defined Radio Networks*, Sept. 2006, pp. 110–119.

[11] S. M. Y. Zhao, J. H. Reed and K. K. Bae, "Overhead analysis for radio environment map-enabled cognitive radio networks," in *1st IEEE Workshop on Networking Technologies for Software Defined Radio Networks*, Sept. 2006, pp. 18–25.

[12] X. J. L. H. Caidan Zhao, Liang Xie and Y. Yao, "A phy-layer authentication approach for transmitter identification in cognitive radio networks," in *International Conference on Communications and Mobile Computing*, vol. 2, Apr. 2010, pp. 154–158.

[13] J. Yang, Y. Chen, and W. Trappe, "Detecting spoofing attacks in mobile wireless environments," in *SECON*, 2009, pp. 1–9.

[14] K. Zeng, K. Govindan, D. Wu, and P. Mohapatra, "Identity-based attack detection in mobile wireless networks." in *INFOCOM*. IEEE, 2011, pp. 1880–1888.

[15] S. Venkatesh, "The design and modeling of ultra-wideband position-location networks," Ph.D. dissertation, Bradley Dept. of Electrical and Computer Engineering, Virginia Tech, VA, USA, 2007.

[16] N. T. Nguyen, R. Zheng, and Z. Han, "On identifying primary user emulation attacks in cognitive radio systems using nonparametric bayesian classification," *IEEE Transactions on Signal Processing*, vol. 60, no. 3, pp. 1432–1445, 2012.

[17] S. P. Chatzis and G. Tsechpenakis, "The infinite hidden markov random field model," *Trans. Neur. Netw.*, vol. 21, pp. 1004–1014, June 2010.

[18] K. Kurihara, M. Welling, and N. A. Vlassis, "Accelerated variational dirichlet process mixtures." in *NIPS'06*, 2006, pp. 761–768.

[19] R. Gomes, M. Welling, and P. Perona, "Incremental learning of nonparametric bayesian mixture models." in *CVPR*. IEEE Computer Society, 2008.

[20] J. Kim, V. Sridhara, and S. Bohacek, "Realistic mobility simulation of urban mesh networks," *Ad Hoc Netw.*, vol. 7, pp. 411–430, March 2009.

[21] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Elsevier: Ad Hoc Networks*, vol. 1, pp. 293–315, 2003.