Secure Indoor Positioning Against Signal Strength Attacks Via Optimized Multi-voting

Yunzhi Li
University of Delaware
liyunzhi@udel.edu

Yidan Hu
University of Delaware
yidanhu@udel.edu

Rui Zhang
University of Delaware
ruizhang@udel.edu

Yanchao Zhang
Arizona State University
yczhang@asu.edu

Terri Hedgpeth
Arizona State University
terrih@asu.edu

ABSTRACT

Indoor positioning systems (IPSes) can enable many location-based services in large indoor venues where GPS signals are unavailable or unreliable. Among the most viable types of IPSes, RSS-IPSes rely on ubiquitous smartphones and indoor WiFi infrastructures and explore distinguishable received signal strength (RSS) measurements at different indoor locations as their location fingerprints. RSS-IPSes are unfortunately vulnerable to physical-layer RSS attacks that cannot be thwarted by conventional cryptographic techniques. Existing defenses against RSS attacks are all subject to an inherent tradeoff between indoor positioning accuracy and attack resilience. This paper presents the design and evaluation of MV-IPS, a novel RSS-IPS based on weighted multi-voting, which does not suffer from this tradeoff. In MV-IPS, every WiFi access point (AP) that receives a user's RSS measurement gives a weighted vote for every reference location, and the reference location that receives the highest accumulative votes from all APs is output as the user's most likely position. Trace-driven simulation studies based on real RSS measurements demonstrate that MV-IPS can achieve much higher positioning accuracy than prior solutions no matter whether RSS attacks are present.

CCS CONCEPTS

• **Security and privacy** → **Mobile and wireless security**; • **Human-centered computing** → *Ubiquitous and mobile computing systems and tools*; • **Networks** → Mobile networks.

KEYWORDS

indoor positioning, RSS, fingerprint, signal strength attack, security

ACM Reference Format:
Yunzhi Li, Yidan Hu, Rui Zhang, Yanchao Zhang, and Terri Hedgpeth. 2019. Secure Indoor Positioning Against Signal Strength Attacks Via Optimized Multi-voting. In *IEEE/ACM International Symposium on Quality of Service (IWQoS '19), June 24–25, 2019, Phoenix, AZ, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3326285.3329068

IWQoS '19, June 24–25, 2019, Phoenix, AZ, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6778-3/19/06…$15.00
https://doi.org/10.1145/3326285.3329068

1 INTRODUCTION

Indoor Positioning Systems (IPSes) have attracted tremendous interest from the academia and industry. An IPS cannot only help mobile users obtain real-time locations in large indoor venues where G-PS signals are unavailable or unreliable, but also provide relevant location contexts to enable a wide range of exciting applications. Exemplary IPS applications include location-based proximity advertising in shopping malls, patient and visitor guidance in hospitals, personnel and asset tracking in factories, and so on.

WiFi-based RSS-IPSes [5, 6] are among the most promising types of IPSes and expected to foster a market of 2.5 billion US dollars by 2020 [1]. Relying on ubiquitous smartphones and existing indoor WiFi infrastructures, RSS-IPSes explore distinguishable received signal strength (RSS) measurements at different indoor locations as their location fingerprints. RSS-IPSes are very attractive to indoor venue owners because there is no need to perform costly infrastructure updates. A typical RSS-IPS works in two phases. In the offline training phase, the IPS operator collects RSS fingerprints at indoor positions to build a fingerprint database of sufficient spatial granularity. In the online positioning phase, on receiving a location query with an RSS measurement from the user, the IPS searches its fingerprint database for the most matching RSS fingerprint and returns the corresponding reference position to the querier.

RSS-IPS is unfortunately vulnerable to signal strength (RSS) attacks at the physical layer that cannot be thwarted by conventional cryptographic techniques. Early studies [8] demonstrate that RSS measurements can be easily manipulated by placing absorbing materials such as book, water, and foil between transmitting and receiving devices. Most recently, Li *et al.* [29] showed that, by impersonating a few WiFi APs with off-the-shelf wireless routers and fine-tuning their transmission power, the attacker can control the RSS values at a target location to either maximize the distance error or mislead the IPS server into returning an arbitrary wrong location. While several defenses against RSS attacks have proposed in the literature [9, 14, 17, 26, 29], they nevertheless all exhibit a tradeoff between attack resilience and positioning accuracy in the absence of RSS attacks. In other words, the high resilience to RSS attacks is usually achieved at the sacrifice of positioning accuracy when there is no RSS attack. This inherent tradeoff makes existing defenses less appealing to IPS operators and thus less likely to be adopted in reality.

Is it possible to design an RSS-IPS that can achieve high positioning accuracy in spite of the presence or absence of RSS attacks?

In this paper, we provide an affirmative answer to the above question by introducing MV-IPS, a novel RSS-IPS with higher positioning accuracy than prior defenses in both cases. We find that the key to achieve high positioning accuracy in both situations is to fully utilize every AP's information while limiting the impact of any individual AP on the final decision. Specifically, making full use of all available APs can improve positioning accuracy in the absence of RSS attacks, whereas limiting each individual AP's role in final decision making can provide resilience against RSS attacks.

Based on this observation, we design MV-IPS based on the Borda count [2], a family of single-winner preferential voting methods widely used in practice. In MV-IPS, the IPS server uses every AP as a voter to give a weighted vote for every reference (or candidate) location in its RSS fingerprint database. A user still submits a location query as usual, which contains the RSS measurement from each available AP. Then the IPS server produces a ranked list of reference locations per AP according to the similarity between the received RSS and the RSS fingerprint for the AP at each reference location. In addition, each reference location is assigned a point value that corresponds to the AP's weighted vote and depends on both its rank and parameters pre-trained from RSS measurements. Finally, the IPS server outputs the reference location that receives the highest total point values from all APs as the querier's most likely location.

Our contributions can be summarized as follows.

- We identify a key limitation of existing defenses against physical-layer signal strength attacks on RSS-IPSes, which forces IPS operators to choose between attack resilience and positioning accuracy.
- We propose MV-IPS, a novel RSS-IPS that explores Borda count voting mechanisms for indoor localization, in which WiFI APs cast weighted votes to jointly determine user locations.
- We formulate the location-weight assignment as an optimization problem to accommodate different APs' capabilities in differentiating user locations and present the optimal solution based on the projected gradient descent.
- We conduct trace-driven simulation studies based on prototype implementations and real RSS data to confirm the effectiveness of MV-IPS in the presence and absence of RSS attacks. Specifically, our evaluation results show that MV-IPS can achieve an average distance error of 1.68 m in the absence of RSS attacks in contrast to 1.96 m achieved by the state-of-art defense [29] while being highly resilient to RSS attacks.

The rest of the paper is structured as follows. Section 2 briefs the related work. Section 3 presents the MV-IPS design. Section 4 evaluates the performance of MV-IPS via trace-driven simulations. Section 5 concludes this paper.

2 RELATED WORK

In this section, we review some most related work.

RSS-IPSes have been studied extensively in the past two decades, and existing solutions differ in how a user's RSS measurement is matched with RSS fingerprints. In the deterministic RSS-IPS, on receiving a location query from the user, the IPS operator evaluates the similarity between the user's RSS measurement and the stored RSS fingerprints using a proper distance metric and returns the reference location whose RSS fingerprint is most similar to the user's RSS measurement. As a representative deterministic-matching RSS-IPS, Radar[5, 6] uses Euclidean distance as the distance metric. In addition, cosine similarity [12] and Tanimato similarity [13] have been shown to yield satisfactory positioning accuracy. Moreover, Wu *et al.* [23] applied support vector machine, and Nuno *et al.* [19] adopted linear discriminant analysis for RSS fingerprint matching. Probabilistic matching has also been used in RSS-IPS. For example, Horus [28] represents the RSS fingerprint at each reference location as the probability distribution of the RSS value and determines the user's location using maximum-likelihood estimation. Other probabilistic matching algorithms have been proposed, including Bayesian network [18], expectation-maximization [11], and Gaussian process [10]. None of these works are resilient to RSS attacks.

There have been some efforts to design RSS-IPS resilient to RSS attacks. Li *et al.* [17] introduced a median-distance based defense in which the distance between the user's RSS measurement and the RSS fingerprint is calculated with respect to every AP, and the reference location with the smallest median distance is chosen as the user's location. The work [26] explored K-means cluster to distinguish good APs and attacked APs according to their geometric relationship. Kushki *et al.* [14] proposed to select a subset of reliable APs according to their confidence scores based on the covariance matrix. Fang *et al.* [9] introduced an attack-resistant localization scheme based on a probabilistic inclusive disjunction model. Yang *et al.* [25] explored Trained Mean Matching (TMM) to detect the evil twin attack in RSS-IPS. Most recently, Yuan *et al.* [29] introduced a defense against RSS attacks. However, all these solutions would force an IPS operator to choose between attack resiliency and positioning accuracy, which makes them less likely to be adopted in practice.

There are also some works loosely related to our work. For example, Li *et al.* [16] introduced several mechanisms to filter out fake RSS data in crowdsourced IPS systems. As another example, PriWFL [15] protects user's location privacy by encrypting a user's location query using Pallier cryptosystem, which is subsequently improved by Yang *et al.* [27] to further protect the fingerprint database at the IPS operator. There are also several IPSes that do not solely rely on RSS. For example, BSurroundSend [4] explores ambient information such as sound and light information to enrich the fingerprint and improve positioning accuracy. As another example, PinLoc [20] explores detailed physical layer information such as channel frequency responses to improve the position accuracy of WiFi-based IPS. Wu *et al.* [24] showed that signal fingerprints based on Channel State Information (CSI) can improve the indoor localization performance. Similarly, DeepFi [21, 22] adopts deep learning to perform indoor localization using fine-grained CSI-based fingerprints.

3 MV-IPS DESIGN

In this section, we first give an overview of MV-IPS and then detail its design.

3.1 Overview

The design of MV-IPS is inspired by the Borda count, a family of single-winner preferential voting methods widely used in both political and non-political elections. In a typical Borda count voting, every voter ranks candidates in order of preference and assigns a point value, i.e., weight vote, to every candidate based on the candidate's ranking such that higher-ranked candidates receive more point values. When all votes are cast, the candidate who receives the maximum total points is chosen as the winner. Different Borda count methods vary in how point values are assigned in accordance with rankings.

In MV-IPS, we view APs as voters and reference locations as candidates. A user still submits a location query as usual, which contains the RSS from each AP. Each AP is associated with a set of RSS fingerprints and corresponding reference locations in the IPS server's fingerprint database. So the IPS server can easily generate a ranked list of reference locations for each AP according to the difference between the received RSS and corresponding fingerprint in the database: smaller difference leads to higher ranking. The IPS server also assigns a point value to each reference location for each AP, which corresponds to the AP's weighted vote. Finally, the reference location that receives the maximum total point values is considered the user's most likely location.

A key difference between MV-IPS and the standard Borda count voting lies in how APs assign point values to their rankings. In particular, we observe that different APs could have diverse capabilities in determining user locations. For example, an AP of which the RSS exhibits large variation across different locations can provide more reliable evidence about a user's location than the one with extremely low or very similar RSS values at many reference locations. As a result, unlike traditional Borda count methods in which all voters share the same point assignment rule, the APs follow different point assignment rules under MV-IPS. In MV-IPS, we formulate the point assignment as an optimization problem based on RSS training data and then find an optimal point assignment rule via the projected gradient decent.

3.2 Detail Design

As many other RSS-IPSes, MV-IPS consists of two phases. In the offline training phase, we collect RSS fingerprints at both reference locations and training locations in the indoor venue of interest and train the system parameters. In the online positioning phase, the IPS server answers location queries from users based on the received RSS measurements and its RSS fingerprint database. In what follows, we first introduce how system parameters are trained based on RSS measurements and then explain how the IPS operator determines a user's location on receiving a location query.

3.2.1 *Data Collection.*

We first choose n reference locations $x_1, \cdots, x_n$ and m training locations $y_1, \ldots, y_m$ in the indoor venue. We then pre-compute $d[i, j]$ as the Euclidian distance between reference location x_i and training location y_j for all $1 \le i \le n$ and $1 \le j \le m$.

We then collect one RSS fingerprint at each of the n reference locations. The RSS measurement collected at reference location x_i is denoted by $\text{rss}_i = (\text{rss}_{i,1}, \ldots, \text{rss}_{i,p})$, where $\text{rss}_{i,z}$ is the zth AP's RSS at reference location x_i for all $1 \le z \le p$ and $1 \le i \le n$, and p is the number of APs in the indoor venue. These RSS measurements serve as the RSS fingerprint of the reference locations. We also collect the RSS measurements at m training locations. We denote the RSS measurement collected at training location y_i by $\text{rss}'_i = (\text{rss}'_{i,1}, \ldots, \text{rss}'_{i,p})$, where $\text{rss}'_{i,z}$ is the jth AP's RSS at training location y_i for all $1 \le z \le p$ and $1 \le i \le m$.

3.2.2 *Parameter Training.*

Let $W = [w]_{p \times n}$ be the weight matrix, where $w[z, i]$ is weight assigned by AP z to the reference location ranked ith for all $1 \le z \le p$ and $1 \le i \le n$. We use the following method to train W using the collected RSS measurements $\{\text{rss}_i | 1 \le i \le n\} \bigcup \{\text{rss}'_i | 1 \le i \le m\}$.

First, for every training location y_j, $1 \le j \le m$, with RSS measurement $\text{rss}'_j = (\text{rss}'_{j,1}, \cdots, \text{rss}'_{j,p})$, we first find its ranking under each AP. Specifically, each AP z, $1 \le z \le p$, calculates the difference between $\text{rss}'_{j,z}$ and the fingerprint of reference location x_i as

$$\triangle_z(i, j) = |\text{rss}_{i,z} - \text{rss}'_{j,z}|,$$

for all $1 \le i \le n$. Each AP z, $1 \le z \le p$, then ranks the n reference locations based on $\triangle_z(i, j)$. Let $(\phi_{z,j}(1), \phi_{z,j}(2), \ldots, \phi_{z,j}(n))$ be a permutation of $(1, 2, \ldots, n)$, such that $\triangle_z(\phi_{z,j}(1), j) < \triangle_z(\phi_{z,j}(2), j) < \cdots < \triangle_z(\phi_{z,j}(n), j)$. Reference position $x_{\phi_{z,j}(i)}$ is then ranked ith by AP z for all $1 \le z \le p$ and $1 \le i \le n$. We also define $\phi^{-1}_{z,j}(\cdot)$ as the inverse of permutation $\phi_{z,j}(\cdot)$, i.e., reference location x_i is ranked $\phi^{-1}_{z,j}(i)$th by AP z.

We repeat the above procedure for all m training locations to obtain m rank matrixes $\Phi^1, \ldots, \Phi^m$, where

$$\Phi^j = \begin{bmatrix} \phi_{1,j}(1) & \phi_{1,j}(2) & \ldots & \phi_{1,j}(n) \\ \phi_{2,j}(1) & \phi_{2,j}(2) & \ldots & \phi_{2,j}(n) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{p,j}(1) & \phi_{p,j}(2) & \ldots & \phi_{p,j}(n) \end{bmatrix}$$

for all $1 \le j \le m$.

Second, for every training location y_j, $1 \le j \le m$, we find the returned reference location according rank matrix Φ^j under weight matrix W. Specifically, under a given weight matrix W, each AP z gives reference location i a weighted vote $w[z, \phi^{-1}_{z,j}(i)]$ for all $1 \le z \le p$ and $1 \le i \le n$. The total weight that reference location x_i receives from all n APs is then given by

$$w[i|j] = \sum_{z=1}^{p} w[z, \phi^{-1}_{z,j}(i)], \tag{1}$$

for all $1 \le i \le n$. Given $w[1|j], \ldots, w[n|j]$, the reference location x_{j^*} with the highest total weight is estimated as the location for RSS measurement rss'_j, where

$$j^* = \underset{i \in \{1, \ldots, n\}}{\arg\max}\ w[i|j],$$

which results in a distance error $d[j^*, j]$.

We now formulate the training of weight matrix W as an optimization problem where we seek to find a weight matrix that minimizes the average error distance across m training locations. Our cost function is inspired by softargmax function widely used in multiclass classification. Consider training location y_j as an example. There are n possible reference locations $x_1, \ldots, x_n$ that rss'_j may be estimated into under MV-IPS. Since the distance between

y_j and reference location x_i is $d[i,j]$, we define the loss function with respect to training location y_j as

$$\mathcal{L}_j(W) = \frac{\sum_{i=1}^{n} \exp(\gamma \cdot w[i|j]) \cdot d[i,j]}{\sum_{i=1}^{n} \exp(\gamma \cdot w[i|j])} \tag{2}$$

where $w[i|j]$ is given in Eq. (1) and $\gamma > 0$ is a system parameter. It is easy to see that as $\gamma \longrightarrow \infty$, the term $\exp(\gamma \cdot w[j^*|j])$ dominates other terms and $\mathcal{L}(j)$ converges to $d[j^*, j]$. We further define the cost function as

$$\mathcal{J}(W) = \frac{1}{m} \sum_{j=1}^{m} \mathcal{L}_j(W). \tag{3}$$

To find the optimal weight assignment, we seek to solve the following optimization problem

$$\begin{aligned} &\textbf{Minimize} \quad \mathcal{J}(W) \\ &\textbf{Subject to} \quad \sum_{i=1}^{n} w[z,i] = 1, \quad \forall 1 \le z \le p, \end{aligned} \tag{4}$$

where the constraint indicates that every AP has a total weight of one.

We use the projected gradient method [7] to find a local minimum for $\mathcal{J}(W)$. Specifically, let us first consider loss function $\mathcal{L}_j(W)$ by rewriting it in terms of $\{w[z,i] | 1 \le z \le p, 1 \le i \le n\}$ as

$$\mathcal{L}_j(W) = \frac{\sum_{i=1}^{n} \exp(\gamma \cdot \sum_{z=1}^{p} w[z, \phi_{z,j}^{-1}(i)]) \cdot d[i,j]}{\sum_{i=1}^{n} \exp(\gamma \cdot \sum_{z=1}^{p} w[z, \phi_{z,j}^{-1}(i)]))}. \tag{5}$$

Let $i' = \phi_{z,j}^{-1}(i)$. It follows that $i = \phi_{z,j}(i')$. Substituting $\phi_{z,j}^{-1}(i)$ and i by i' and $\phi_{z,j}(i')$, respectively, we can rewrite $\mathcal{L}_j(W)$ as

$$\mathcal{L}_j(W) = \frac{\sum_{i'=1}^{n} \exp(\gamma \cdot \sum_{z=1}^{p} w[z,i']) \cdot d[\phi_{z,j}(i'), j]}{\sum_{i'=1}^{n} \exp(\gamma \cdot \sum_{z=1}^{p} w[z,i']))}. \tag{6}$$

We observe that every $w[z,i'], 1 \le z \le p, 1 \le i' \le n$ appears in both the numerator and denominator of $\mathcal{L}_j(W)$.

We now derive partial derivative of $\mathcal{L}_j(W)$ with respect to $\mathcal{L}_j(W)$. Specifically, let us define two additional functions as

$$f = \sum_{i'=1}^{n} \exp(\gamma \cdot \sum_{z=1}^{p} w[z,i']) \cdot d[\phi_{z,j}(i'), j],$$

and

$$g = \sum_{i'=1}^{n} \exp(\gamma \cdot \sum_{z=1}^{p} w[z,i']).$$

The partial derivatives of f and g with respect to $w[u,v]$ are given by

$$\frac{\partial f}{\partial w[u,v]} = \gamma \cdot \exp(\gamma \cdot \sum_{z=1}^{p} w[z,v]) \cdot d[\phi_{z,j}(v), j] \tag{7}$$

and

$$\frac{\partial g}{\partial w[u,v]} = \gamma \cdot \exp(\gamma \cdot \sum_{z=1}^{p} w[z,v]), \tag{8}$$

respectively, for all $1 \le u \le p$ and $1 \le v \le n$. We can then compute the partial derivative $\mathcal{L}_j(W)$ with respect to each $w[u,v]$ as

$$\frac{\partial \mathcal{L}_j}{\partial w[u,v]} = \frac{\frac{\partial f}{\partial w[u,v]} \cdot g - f \cdot \frac{\partial g}{\partial w[u,v]}}{g^2}, \tag{9}$$

Algorithm 1: Weight Matrix Training

```
input : Initial weight matrix $W^{(0)}$, error distances
        $\{d[i,j] | 1 \le i \le n, 1 \le j \le m\}$, rank matrices
        $\Phi^1, \ldots, \Phi^m$, learning rate $\eta$, and terminal
        parameter $\epsilon$
output: Weight matrix $W^{(t)}$
$t \leftarrow 1$;
while True do
    foreach $j \in \{1, \ldots, m\}$ do
        foreach $i \in \{1, \ldots, n\}$ do
            Compute $w[i|j]$ according to Eq. (1);
        end
        Compute $\mathcal{L}_j(W^{(t-1)})$ according to Eq. (6);
    end
    $\mathcal{J}(W^{(t-1)}) \leftarrow \frac{1}{m} \sum_{j=1}^{m} \mathcal{L}_j(W^{(t-1)})$;
    Compute $\nabla \mathcal{J}(W^{(t-1)})$ according to Eq. (10);
    $W^{(t)} \leftarrow W^{(t-1)} - \eta P \nabla \mathcal{J}(W^{(t-1)})$;
    if $|\mathcal{J}(W^{(t-1)}) - \mathcal{J}(W^{(t)})| < \epsilon$ then
        break;
    else
        $t \leftarrow t + 1$;
    end
end
return $W^{(t)}$;
```

for all $1 \le u \le p, 1 \le v \le n$, where $\frac{\partial f}{\partial w_{u,v}}$ and $\frac{\partial g}{\partial w_{u,v}}$ are given in Eqs. (7) and (8), respectively. Finally, we can derive the partial derivative of cost function $\mathcal{J}(W)$ with respect to $w[u,v]$ as

$$\frac{\partial \mathcal{J}}{\partial w[u,v]} = \frac{1}{m} \sum_{j=1}^{m} \frac{\partial \mathcal{L}_j}{\partial w[u,v]}. \tag{10}$$

To apply the projected gradient method [7], we rewrite the constraint in the optimization problem as

$$AW = b,$$

where

$$A = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{1} \end{bmatrix},$$

$\mathbf{0} = (0, \ldots, 0)$, $\mathbf{1} = (1, \ldots, 1)$, $W = [W_1, W_2, \ldots, W_p]^T$, $W_z = (w[z,1], \ldots, w[z,n])$ for all $1 \le z \le p$, and $b_{1 \times p} = (1, 1, \ldots, 1)^T$. We can then compute the orthogonal projector matrix as

$$P = I_{np} - A^T (AA^T)^{-1} A, \tag{11}$$

where I_{np} is the $np \times np$ identity matrix and T on superscript denotes matrix transposition.

Let $W^{(0)}$ be the initial weight matrix, where we set $w[z,i] = \frac{2(n-i+1)}{n(n+1)}$ for all $1 \le i \le n$ and $1 \le z \le p$ as in the standard Borda count voting. We repeatedly compute

$$W^{(t)} = W^{(t-1)} - \eta P \nabla \mathcal{J}(W^{(t-1)}) \tag{12}$$

for $t = 1, 2, 3, \ldots$, where η is the learning rate and $\nabla\mathcal{J}(W^{(t-1)})$ is the gradient of $\mathcal{J}$ with respect to $W^{(t-1)}$ given by Eq. (10). The learning rate η is usually set dynamically via backtracking line search [3]. The process terminates if

$$|\mathcal{J}(W^{(t)}) - \mathcal{J}(W^{(t+1)})| < \epsilon,$$

where ϵ is a small constant.

We summarize the training process in Algorithm 1. Line 1 initializes the iteration index $t = 1$. In Lines 2-17, we iteratively update the weight matrix until the terminal condition is met. Specifically, in Line 4-6, for every training location y_j, we compute the total weight each reference location x_i receives according to Eq. (1). Next, based on $\{w[i|j]|1 \le i \le n\}$, the loss function of each training location y_j is calculated according to Eq. (6) in Line 7. We then calculate the cost function as the average loss across all m training locations in Line 9. In Lines 10-11, we compute the gradient of the cost function using Eq. (10) and then update the weight matrix updating according to Eq. (12). If the difference between the cost function of the new weight matrix and the cost function of the previous weight matrix is less than ϵ, we terminate the process and output the current weight matrix. Otherwise, we repeat the same procedure in the next iteration.

3.2.3 Online Positioning. In the online positioning phase, the IPS operator processes location queries from the user. Assume that the user issues a location query with RSS measurement $\mathrm{rss}_u = (\mathrm{rss}_{u,1}, \ldots, \mathrm{rss}_{u,p})$. The IPS operator first computes

$$\triangle_z(i, u) = |\mathrm{rss}_{i,z} - \mathrm{rss}_{u,z}|,$$

for all $1 \le z \le p$ and $1 \le i \le n$. Each AP z $(1 \le z \le p)$ then ranks the n reference locations based on $\triangle_z(i, u)$. The IPS operator then computes

$$w[i|u] = \sum_{z=1}^{p} w[z, \phi_{z,u}^{-1}(i)], \tag{13}$$

for all $1 \le i \le n$, where $\phi_{z,u}^{-1}(i)$ is the rank of reference location x_i under AP z given rss_u. Given $w[1|u], \ldots, w[n|u]$, the reference location x_{u^*} with the highest total weight is estimated as the user's location, where

$$u^* = \underset{i\in\{1,\ldots,n\}}{\arg\max}\ w[i|u].$$

4 PERFORMANCE EVALUATION

In this section, we report the simulation results for MV-IPS.

4.1 Simulation Settings

We have implemented a prototype of MV-IPS. The prototype system is based on Android studio/Java on a Huawei Honor8 smartphone, which has a 2.3 GHz octa-core CPU and 4 GB RAM. The sampling frequency of the WiFi module is 0.67 Hz. We deploy the prototype on a square zone of 17.8×17.8m^2 inside an office building with $m = 35$ WiFi APs. Fig. 1 shows the floor plan of the indoor venue.

We collect RSS measurements at $n = 72$ reference locations and $m = 360$ training locations as shown in Fig. 1, where every reference location is surrounded by five reference locations. We use the RSS measurements collected at the 72 reference locations as the RSS-fingerprint database and the ones collected at the 360 training locations to train the weight matrix and evaluate the performance of the MV-IPS. In particular, we divide the 360 training measurements into five groups of equal size, in which every group contains one training measurement close to each reference location.

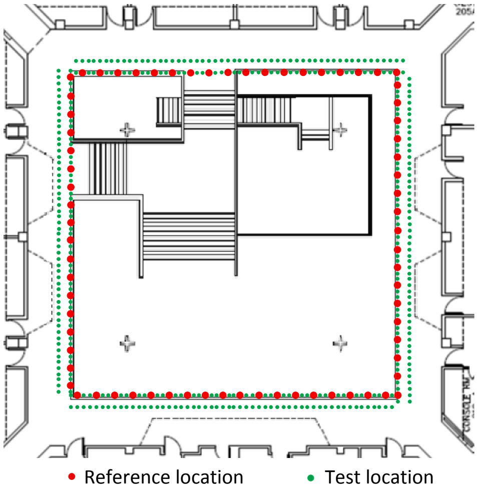


Figure 1: The floor plan of the indoor venue.

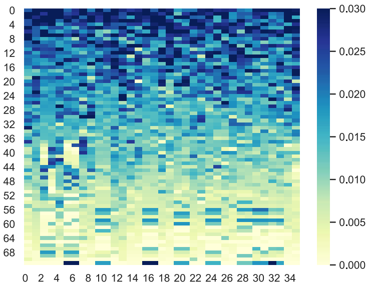


Figure 2: An example of weighted matrix under MV-IPS.

We use 5-fold cross-validation to evaluate the performance of MV-IPS. Specifically, we select 4 groups of training measurements to train the weight matrix and use the remaining group as the testing RSS measurements. We repeat this process for five times such that every group is used as the testing set once. The results we report below are the average across the 5 runs.

We mainly use mean distance error (MDE) to evaluate the performance of MV-IPS. For every rss'_j in a testing group G, let y_j be the training location at which it was was taken and x_{j^*} the reference location returned by MV-IPS. We define the MDE as

$$\mathrm{MDE} = \frac{\sum_{\mathrm{rss}'_j \in G} d[j^*, j]}{|G|}.$$

We consider an attacker model similar to the one in [29]. In particular, we assume that the attacker is able to impersonate a subset of k APs of his choice with fake ones under his control. He can also fine-tune the transmission powers of the fake APs to manipulate the RSS values experienced by a target user. We consider the following two attack strategies.

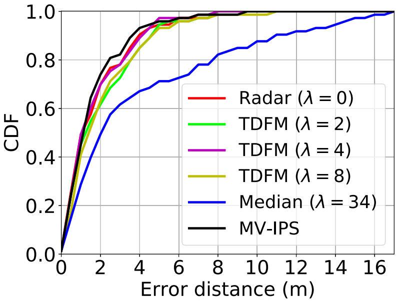


Figure 3: Comparison of the CDFs of distance error under Radar, TDFM, Median, and MV-IPS.

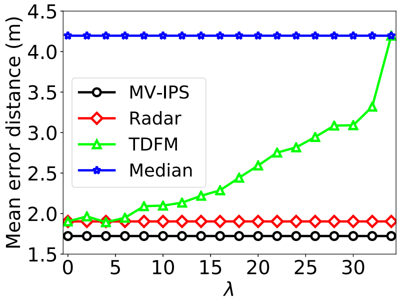


Figure 4: MDE vs. λ in the absence of RSS attack.

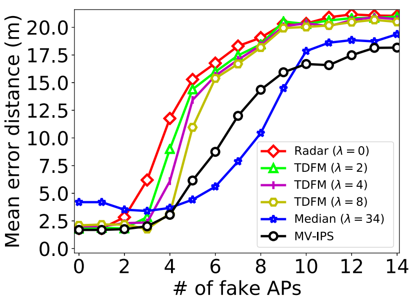


Figure 5: MDE vs. # of fake APs under DEM attack.

- *Distance error maximization (DEM) attack* [29]: The attacker aims to maximize the distance error experience by a target user. In this attack, the attackers first learns the RSS fingerprints stored at the IPS operator by acting as a normal user repeatedly issuing location queries and then finds the furthest reference location from the target user's location that he can mislead the IPS operator into returning through impersonating and manipulating the transmission powers of the k fake APs. With respect to MV-IPS, we assume that the attacker knows the weight matrix used by the IPS operator and is able to perfectly control the fake APs' RSS values experienced by the target user.
- *Random RSS attack*: In this attack, the attacker controls the RSSes of k APs randomly chosen from all the APs. The user's RSS measurement under this attack is assumed to be uniform at random in the range from -30 dB to -95 dB.

We compare the MV-IPS with the following RSS-IPSes.

- Radar [5, 6]: As the most representative RSS-IPS, Radar returns the reference location whose RSS fingerprint is the closest to the user's RSS measurement under the Euclidean distance.
- Median [17]: As a defense against RSS attack, the median-based defense uses the median among the p element-wise distances as the metric to measure the similarity between the user's RSS measurement and the stored RSS fingerprint. The reference location with the smallest median element-wise distance is selected as the user's location.
- TDFM [29]: As the state-of-art defense against RSS attacks, TDFM generalizes Median [17] and Radar. Specifically, the IPS operator calculates p element-wise distances between the user's RSS measurement and each RSS fingerprint with each corresponding to one AP. The similarity between the user's RSS measurement and an RSS fingerprint is measured by the λ-truncated distance [29], which is the sum of $p - \lambda$ element-wise distances after dropping the $\lambda/2$ largest and $\lambda/2$ smallest element-wise distances. When $\lambda = 0$, TDFM is equivalent to Radar. When $\lambda = (p-1)/2$, TDFM is equivalent to the median-based defense.

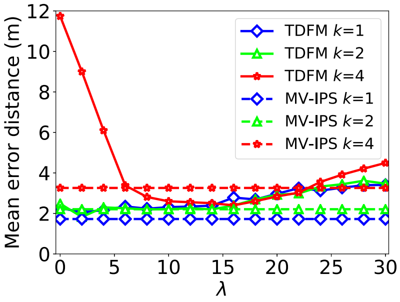


Figure 6: MDE vs. λ under DEM attack.

Table 1 summarizes the default parameters in our simulation unless stated otherwise.

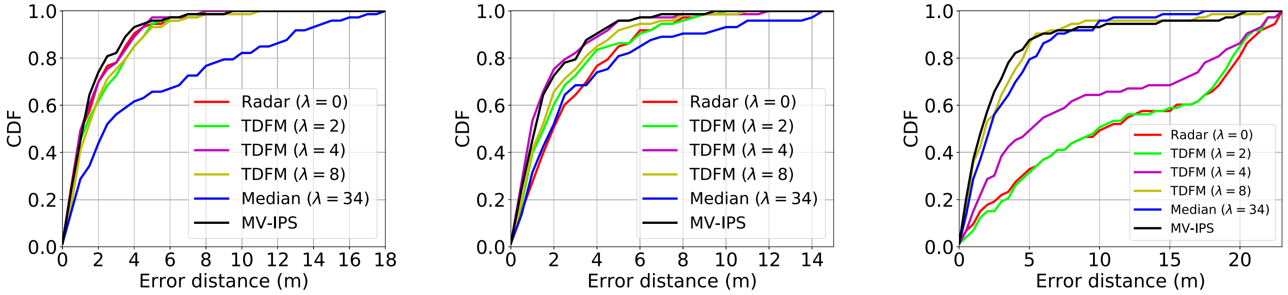
Table 1: Default Settings

Para.	Value	Description
n	72	# of reference locations
m	360	# of training locations
p	35	# of APs
γ	200	The exponential parameter in Eq. (2)
η	0.1	Learning rate
ϵ	0.9	The terminating condition

4.2 Simulation Results

We now report our simulation results.

4.2.1 An Example Of Weight Matrix In MV-IPS. Fig. 2 shows an example of the weighted matrix trained in MV-IPS, where the x-axis represents the IDs of APs and y-axis represents the indexes of reference locations. As we can see, different APs have very different weight assignments over their rankings. This clears highlights the key difference between MV-IPS and the standard Boada count voting where all the voters use the same weight assignment.



(a) CDF of MDE where $k = 1$

(b) CDF of MDE where $k = 2$

(c) CDF of MDE where $k = 4$

Figure 7: CDF of MDEs under DEM attack.

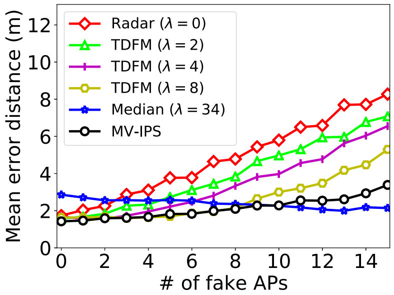


Figure 8: MDE vs. # of fake APs under random RSS attack.

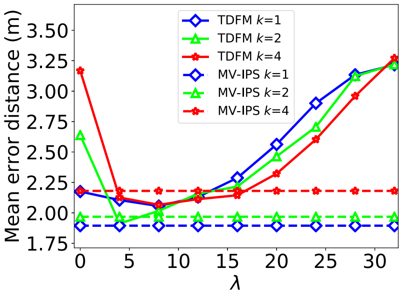


Figure 9: MDE vs. λ under random RSS attack.

4.2.2 *Performance In The Absence Of RSS Attack.*

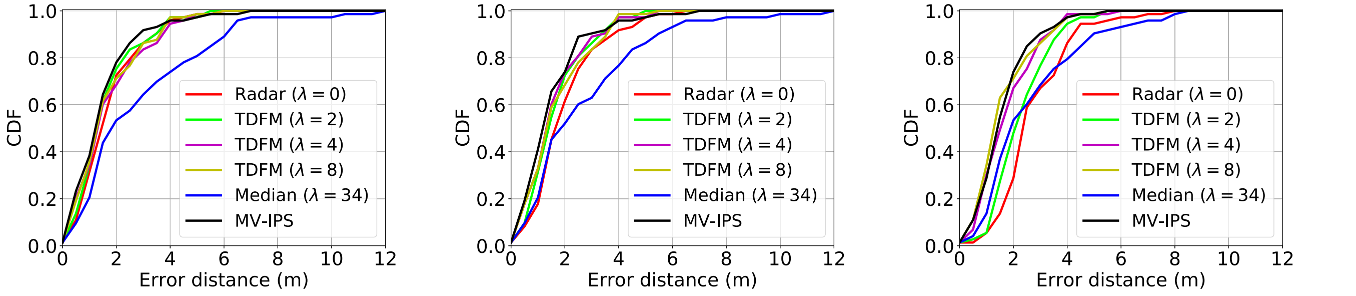
Fig. 3 compares the CDFs of the error distance under Radar, Median, TDFM, and MV-IPS in the absence of RSS attack. As we can see, MV-IPS not only outperforms Median and TDFM, but also achieves smaller MAE than Radar that is designed for benign environment. For example, 83% of distance errors are smaller than 2.5m under MV-IPS, whereas 78 % and 71 % of distance errors are smaller than 2.5m under TDFM and Median, respectively. These results demonstrate that MV-IPS achieves higher positioning accuracy than prior defenses and Radar in the absence of RSS attack and is thus more appealing to IPS operators in reality.

Fig. 4 compares the MDEs under Radar, Median, TDFM, and MV-IPS as λ varying from 0 to 34, where the MDEs under Radar, Median, and MV-IPS are not affected by the change in λ and are plotted for reference only. We can see that MV-IPS has the smallest MDE among the four. In addition, the MDE under TDFM increases as λ increases. This is expected, because the more element-wise distances dropped, the lower the positioning accuracy for TDFM in the absence of RSS attack, and vice versa.

4.2.3 *Performance Under DEM Attack.*

Figure. 5 compares the MDE under TDFM and MV-IPS under the DEM attack with the number of fake APs varying from 0 to 14. We can see that the MDE increases as the number of fake APs increases under both TDFM and MV-IPS. In particular, when the number of fake APs is small, e.g., 2, MV-IPS outperforms TDFM with smaller MDE. As the number of fake APs increases to 3, MV-IPS still outperforms other mechanisms except for TDFM with $\lambda = 8$. This is anticipated as when λ is set to be slightly larger than twice of the number of fake APs, all the element-wise distances involving fake APs are likely dropped and the remaining good RSS values can ensure sufficiently high positioning accuracy. However, properly setting λ would require the IPS operator to know the number of fake APs in advance, which is usually unavailable in practice. On the other hand, when λ is set too small or too large, either some fake RSS values will be used for determining user's location or too many good RSS values are dropped, leading to the increase in MDE and thus lower positioning accuracy. In contrast, MV-IPS does not require the IPS operator to tune any parameter and can always maintain high positioning accuracy.

Fig. 6 compares the MDEs under TDFM and MV-IPS with λ varying from 0 to 30, where the MDE under MV-IPS is not affected by the change in λ and plotted for reference only. As we can see that for any given number of fake APs, the MDE under TDFM first decreases and then increases as λ increases. The reason is that when λ is set too small, some fake RSS values are included for location determination, which results in lower positioning accuracy. When the λ is large enough, all fake RSS values are likely dropped, leading to higher positioning accuracy of TDFM. Even in this scenario, we can see that the MDE under MV-IPS is still very close to that under TDFM. Furthermore, as λ further increases, MV-IPS outperforms



(a) CDF of distance error where $k = 1$

(b) CDF of distance error where $k = 2$

(c) CDF of distance error where $k = 4$

Figure 10: CDFs of distance error under random RSS attack.

TDFM again. Since λ is normally difficult to set without knowing the number of fake APs in advance, these results highlight the significant advantage of MV-IPS over the state-of-art defense TDFM.

Figs. 7(a) to 7(c) show the CDFs of distance error under Radar, TDFM, Median, and MV-IPS where the number of fake APs is 1, 2, and 4, respectively. Generally speaking, MV-IPS outperforms other three mechanisms in most cases. In particular, as shown in Figs. 7(a) to 7(c), the MDE under MV-IPS is always smaller than those under Radar, Median, and TDFM except for TDFM with $\lambda = 2$. We can also see from Fig. 7(c) that when $k = 4$, the MDE under Median is acceptable and is close to that under MV-IPS, but the MDE under Radar increases drastically. The MDE under TDFM is always between that under Radar and that under Median but still larger than that under MV-IPS. These results show that MV-IPS is highly resilient to DEM attack when the number of fake APs is small.

4.2.4 Performance Under Random RSS Attack. Fig. 8 shows the MDEs under Radar, TDFM, Median, and MV-IPS under random RSS attack with the number of fake APs varying from 0 to 15. We can see that the MDEs under Radar, TDFM, and MV-IPS all increase as the number of fake APs increases. This anticipated as the more fake APs, the more fake RSS values being used for determining the user's location. In contrast, the MDE under Median decreases as the number of fake APs increases. We can also see that the MDE under MV-IPS grows much slower than under TDFM and Radar. While the MDE under MV-IPS is not as low as that under Median when there are more than ten fake APs, it outperforms Median by a large margin when there are fewer than ten fake APs.

Fig. 9 compares the MDEs under TDFM and MV-IPS under random RSS attack with λ varying from 0 to 32, where the MDEs of MV-IPS are not affected by the change in λ and are plotted for reference only. We can see that the MDE under TDFM first declines and then increases as λ increases, which once gain highlights the importance of properly setting λ for TDFM. Furthermore, while the MDE under TDFM is acceptable when λ is in the range of $(5, 15)$, the MDE under MV-IPS is either very close to or smaller than that under TDFM.

Figs. 10(a) to 10(c) show the CDFs of distance error under Radar, TDFM, Median, and MV-IPS under random RSS attack. Once again, we can see that MV-IPS outperforms the other three mechanisms in most cases. While we can see from Fig. 10(b) that TDFM achieves a MDE comparable to MV-IPS when $\lambda = 2$, properly setting λ is difficult without knowing the number of fake APs in advance. Finally, the MDEs under all four mechanisms are smaller than the corresponding cases under the DEM attack, which is anticipated as random RSS attack is less effective than DEM attack. It is thus not surprising that Median always has the highest MDE even when $k = 4$.

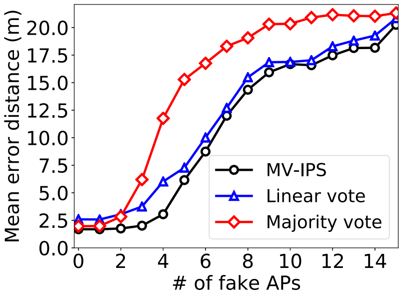
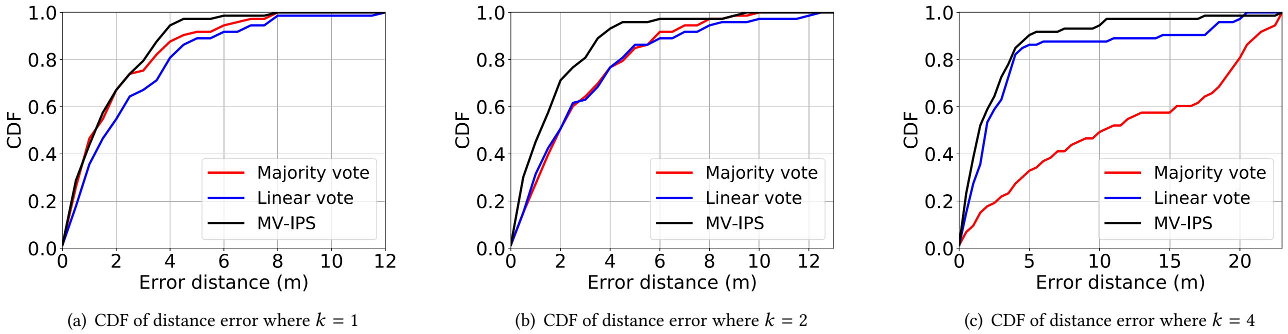


Figure 11: Impact of weight matrix on MDE where the number of fake APs varying from 0 to 14.

4.2.5 Impact of Different Weight Matrices. While the weight matrix is trained from RSS data under MV-IPS, we also evaluate the impact of different weight matrices. Specifically, we compare MV-IPS with the mechanisms based on the following two weight matrices.

- *Linear vote*: the weight matrix is the same as the initial weight matrix $W^{(0)}$, where $w[z, i] = \frac{2(n-i+1)}{n(n+1)}$ for all $1 \le i \le n$ and $1 \le z \le p$.
- *Majority vote*: the weight matrix is defined by $w[z, 1] = 1$ and $[z, j] = 0$ for all $1 \le z \le p$ and $2 \le j \le n$.

Fig. 11 compares the MDEs under MV-IPS, Linear note, and Majority vote. As we can see, as the number of fake APs increases, the MDEs under all three mechanisms increase, which is expected. Moreover, MV-IPS that uses the trained weight matrix always



(a) CDF of distance error where $k = 1$

(b) CDF of distance error where $k = 2$

(c) CDF of distance error where $k = 4$

Figure 12: CDF of distance error under DEM attack under different weight matrices.

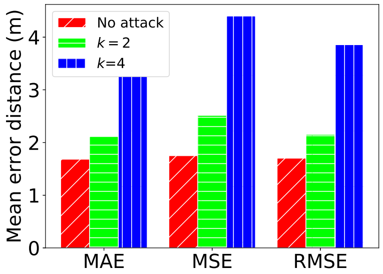


Figure 13: Comparison of different cost functions.

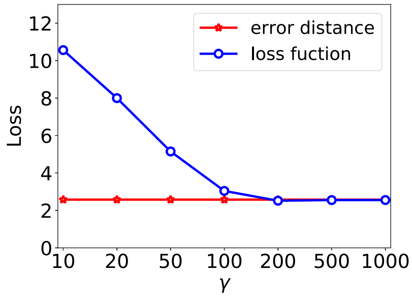


Figure 14: Impact of γ on loss function $\mathcal{L}_j(W)$.

has the lowest MDE among the three, which demonstrates the effectiveness of training in finding a good weight assignment.

Figs. 12(a) to 12(c) show the CDFs of MDEs under MV-IPS, Linear vote, and Majority vote where the number of fake APs is 1,2, and 4, respectively. Not surprisingly, MV-IPS outperforms both Linear Vote and Majority Vote by large margins, especially for the case where $k = 4$. These results confirm the advantages of MV-IPS based on the weight matrix trained by RSS data.

4.2.6 Impact of Different Cost Functions. We also evaluate the impact of different cost functions on the positioning accuracy of MV-IPS. In particular, we evaluate MV-IPS under the following three cost functions.

- Mean Absolute Error (MAE): the cost function used by MV-IPS and given in Eq. (3).
- Mean Square Error (MSE): the cost function is given by

$$\mathcal{J}(W) = \frac{1}{m}\sum_{j=1}^{m}(\mathcal{L}_j(W))^2.$$

- Root Mean Square Error (RMSE): the cost function is given by

$$\mathcal{J}(W) = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(\mathcal{L}_j(W))^2}.$$

Fig. 13 compares the distance errors under three cost functions where the number of fake APs is 0,2, and 4. We can see that the MDEs under MAE and RMSE cost functions are approximately 3.2m when the number of fake APs is 4, whereas that that under MSE is above 4.2m. In addition, the MDE under MSE is always the lowest among the three cost functions. These results indicate that the cost function chosen by MV-IPS outperforms the other two options and leads to high positioning accuracy.

4.2.7 Impact of Parameter γ. The loss function given in Eq. (2) involves the parameter γ. Intuitively, as γ approaches ∞, loss function $\mathcal{L}_j(W)$ approaches to $d[j^*, j]$. We also evaluate the impact of γ. Fig. 14 plots the values of $\mathcal{L}_j(W)$ and $d[j^*, j]$ as γ increases from 10 to 1000. We can see that as γ increases, the difference between $\mathcal{L}_j(W)$ and $d[j^*, j]$ decreases. When γ exceeds 200, the difference between $\mathcal{L}_j(W)$ and $d[j^*, j]$ becomes negligible. By choosing $\mathcal{L}_j(W)$ as the loss function, we are able to derive the close form of the gradient of the cost function $\mathcal{J}(w)$.

4.3 Summary

We summarize the simulation result as follows.

- MV-IPS achieves higher positioning accuracy than Radar, Median, and TDFM in the absence of RSS attacks.
- In the presence of RSS attacks, MV-IPS achieves higher positioning accuracy than Radar and Median. It also either

outperforms TDFM or achieves a positioning accuracy closer to TDFM when the parameter λ is set to be approximately twice of the number of fake APs.

- Unlike TDFM whose performance is highly dependent on properly setting of parameter λ that requires the knowledge of the number of APs, MV-IPS is oblivious to the number of fake APs and can always achieve satisfactory positioning accuracy no matter whether RSS attacks are present.
- MV-IPS relies on a weight matrix properly trained from the RSS data and significantly outperforms other weight matrices used in the standard Borda count and majority vote.

5 CONCLUSION

In this paper, we have introduced the design and evaluation of MV-IPS, a novel RSS-IPS based on weighted multi-voting. Inspired by the Borda count voting, MV-IPS treats every AP as a voter to cast a weighted vote for every reference location, and the reference location that receives the highest accumulative vote is considered as the user's location. Unlike existing RSS-IPSes that suffer from the inherent tradeoff between indoor positioning accuracy and attack resilience, MV-IPS can achieve high indoor positioning accuracy no matter whether RSS attacks are present. Trace-driven simulation studies based on real RSS measurements have confirmed the significant advantages of MV-IPS over prior RSS-IPSes.

ACKNOWLEDGMENTS

We would like to thank anonymous reviewers for their insightful comments that have helped improve the quality of this work. This work was supported in part by the US National Science Foundation under grants CNS-1700039, CNS-1651954 (CAREER), CNS-1718078, CNS-1514381, and CNS-1619251.

REFERENCES

[1] [n.d.]. Wi-Fi Indoor Location in Retail Worth $2.5 Billion by 2020. https://www.abiresearch.com/press/wi-fi-indoor-location-retail-worth-25-billion-2020

[2] Rony M Adelsman and Andrew B Whinston. 1977. Sophisticated voting with information for two voting functions. *Journal of Economic Theory* 15, 1 (1977), 145 – 159.

[3] Larry Armijo. 1966. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific J. Math.* 16, 1 (November. 1966), 1–4.

[4] Martin Azizyan, Ionut Constandache, and Romit Roy Choudhury. 2009. SurroundSense: Mobile Phone Localization via Ambience Fingerprinting. In *Annual International Conference on Mobile Computing and Networking (Mobicom'09)*. Beijing, China, 261–272.

[5] Paramvir Bahl and Venkata N. Padmanabhan. 2000. RADAR: an in-building RF-based user location and tracking system. In *IEEE International Conference on Computer Communications (INFOCOM'00)*, Vol. 2. Tel Aviv, Israel, 775–784. https://doi.org/10.1109/INFCOM.2000.832252

[6] Paramvir Bahl, Venkata N Padmanabhan, and Anand Balachandran. 2000. Enhancements to the RADAR user location and tracking system. *Microsoft Research* 2, MSR-TR-2000-12 (Feb. 2000), 775–784.

[7] Paul H. Calamai and Jorge J. Moré. 1987. Projected gradient methods for linearly constrained problems. *Mathematical Programming* 39, 1 (01 Sepember 1987), 93–116.

[8] Yingying Chen, Konstantinos Kleisouris, Xiaoyan Li, Wade Trappe, and Richard P. Martin. 2009. A Security and Robustness Performance Analysis of Localization Algorithms to Signal Strength Attacks. *ACM Trans. Sen. Netw.* 5, 1 (Feb. 2009), 2:1–2:37. https://doi.org/10.1145/1464420.1464422

[9] Shihhau Fang, Chungchih Chuang, and Chiapin Wang. 2012. Attack-Resistant Wireless Localization Using an Inclusive Disjunction Model. *IEEE Transactions on Communications* 60, 5 (May 2012), 1209–1214. https://doi.org/10.1109/TCOMM.2012.040212.100291

[10] Brian Ferris, Dieter Fox, and Neil Lawrence. 2007. WiFi-SLAM Using Gaussian Process Latent Variable Models. In *International Joint Conference on Artifical Intelligence (IJCAI'07)*. Hyderabad, India, 2480–2485.

[11] Abhishek Goswami, Luis E. Ortiz, and Samir R. Das. 2011. WiGEM: A Learning-based Approach for Indoor Localization. In *International Conference on emerging Networking EXperiments and Technologies (CoNEXT'11)*. 3:1–3:12.

[12] Suining He and S.-H. Gary Chan. 2014. Sect junction: Wi-Fi indoor localization based on junction of signal sectors. In *IEEE International Conference on Communications (ICC'14)*. Sydney, NSW, 2605–2610.

[13] Yifei Jiang, Xin Pan, Kun Li, Qin Lv, Robert P. Dick, Michael Hannigan, and Li Shang. 2012. ARIEL: automatic wi-fi based room fingerprinting for indoor localization. In *ACM Conference on Ubiquitous Computing (UbiComp'12)*. Pittsburgh, PA, 441–450. https://doi.org/10.1145/2370216.2370282

[14] Azadeh Kushki, Konstantinos N. Plataniotis, and Anastasios N. Venetsanopoulos. 2008. Sensor selection for mitigation of RSS-based attacks in wireless local area network positioning. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'08)*. Las Vegas, NV, 2065–2068. https://doi.org/10.1109/ICASSP.2008.4518047

[15] Hong Li, Limin Sun, Haojin Zhu, Xiang Lu, and Xiuzhen Cheng. 2014. Achieving privacy preservation in WiFi fingerprint-based localization. In *IEEE International Conference on Computer Communications (INFOCOM'14)*. Toronto, Canada, 2337–2345. https://doi.org/10.1109/INFOCOM.2014.6848178

[16] Tao Li, Yimin Chen, Rui Zhang, Yanchao Zhang, and Terri Hedgpeth. 2018. Secure crowdsourced indoor positioning systems. In *IEEE International Conference on Computer Communications (INFOCOM'18)*. Honolulu, HI, 1034–1042.

[17] Zang Li, Wade Trappe, Yanyong Zhang, and Badri Nath. 2005. Robust Statistical Methods for Securing Wireless Localization in Sensor Networks. In *International Symposium on Information Processing in Sensor Networks (IPSN'05)*. Los Angeles, CA, 1–12. https://doi.org/10.1109/IPSN.2005.1440903

[18] David Madigan, Eiman Elnahrawy, Richard P. Martin, Wenhua Ju, Prajindra Sankar A/l Krishnanm, and A. S. Krishnakumar. 2005. Bayesian indoor positioning systems. In *IEEE International Conference on Computer Communications (INFOCOM'05)*, Vol. 2. Miami, FL, 1217–1227 vol. 2. https://doi.org/10.1109/INFCOM.2005.1498348

[19] Galo Nuno and Jose Paez Borrallo. 2006. A New Location Estimation System for Wireless Networks Based on Linear Discriminant Functions and Hidden Markov Models. *EURASIP Journal on Applied Signal Processing* 2006 (01 2006), 159–159. https://doi.org/10.1155/ASP/2006/68154

[20] Souvik Sen, Božidar Radunovic, Romit Roy Choudhury, and Tom Minka. 2012. You Are Facing the Mona Lisa: Spot Localization Using PHY Layer Information. In *International Conference on Mobile Systems, Applications, and Services (MobiSys'12)*. Low Wood Bay, Lake District, UK, 183–196.

[21] Xuyu Wang, Lingjun Gao, Shiwen Mao, and Santosh Pandey. 2015. DeepFi: Deep learning for indoor fingerprinting using channel state information. In *IEEE Wireless Communications and Networking Conference (WCNC'15)*. New Orleans, LA, 1666–1671. https://doi.org/10.1109/WCNC.2015.7127718

[22] Xuyu Wang, Lingjun Gao, Shiwen Mao, and Santosh Pandey. 2017. CSI-Based Fingerprinting for Indoor Localization: A Deep Learning Approach. *IEEE Transactions on Vehicular Technology* 66, 1 (January 2017), 763–776. https://doi.org/10.1109/TVT.2016.2545523

[23] Chaolin Wu, Lichen Fu, and Fengli Lian. 2004. WLAN location determination in e-home via support vector classification. In *IEEE International Conference on Networking, Sensing and Control (ICNSC'04)*, Vol. 2. Taipei, Taiwan, 1026–1031, Vol.2. https://doi.org/10.1109/ICNSC.2004.1297088

[24] Kaishun Wu, Jiang Xiao, Youwen Yi, Dihu Chen, Xiaonan Luo, and Lionel M. Ni. 2013. CSI-Based Indoor Localization. *IEEE Transactions on Parallel and Distributed Systems* 24, 7 (July 2013), 1300–1309.

[25] Chao Yang, Yimin Song, and Guofei Gu. 2012. Active User-Side Evil Twin Access Point Detection Using Statistical Techniques. *IEEE Transactions on Information Forensics and Security* 7, 5 (October 2012), 1638–1651. https://doi.org/10.1109/TIFS.2012.2207383

[26] Jie Yang, Yingying Chen, Victor B. Lawrence, and Venkataraman Swaminathan. 2009. Robust wireless localization to attacks on access points. In *IEEE Sarnoff Symposium*. Princeton, NJ, 1–5. https://doi.org/10.1109/SARNOF.2009.4850372

[27] Zheng Yang and Kimmo Javinen. 2018. The Death and Rebirth of Privacy-Preserving WiFi Fingerprint Localization with Paillier Encryption. In *IEEE International Conference on Computer Communications (INFOCOM'18)*. Honolulu, HI, 1223–1231. https://doi.org/10.1109/INFOCOM.2018.8486221

[28] Moustafa Youssef and Ashok Agrawala. 2005. The Horus WLAN Location Determination System. In *International Conference on Mobile Systems, Applications, and Services (MobiSys'05)*. Seattle, WA, 205–218. https://doi.org/10.1145/1067170.1067193

[29] Lizhou Yuan, Yidan Hu, Yunzhi Li, Rui Zhang, Yanchao Zhang, and Terri Hedgpeth. 2018. Secure RSS-Fingerprint-Based Indoor Positioning: Attacks and Countermeasures. In *IEEE Conference on Communications and Network Security (CNS'18)*. Beijing, 1–9. https://doi.org/10.1109/CNS.2018.8433131