Fingerprint-Based Device-Free Localization Performance in Changing Environments

Brad Mager, Philip Lundrigan, and Neal Patwari

***Abstract*—Device-free localization (DFL) systems locate a person in an environment by measuring the changes in received signal on links in a wireless network. A fingerprint-based DFL method collects a training database of measurement fingerprints and uses a machine learning classifier to determine a person's location from a new fingerprint. However, as the environment changes over time due to furniture or other objects being moved, the fingerprints diverge from those in the database. This paper addresses, for DFL methods that use received signal strength as measurements, the degradation caused as a result of environmental changes. We perform experiments to quantify how changes in an environment affect accuracy, through a repetitive process of randomly moving an item in a residential home and then conducting a localization experiment, and then repeating. We quantify the degradation as well as consider ways to be more robust to environmental change. We find that the localization error rate doubles, on average, for every six random changes in the environment. We find that the random forests classifier has the lowest error rate among four tested. We present a correlation method for selecting channels which decreases the localization error rate from 4.8% to 1.6%.**

I. Introduction

Device-free localization (DFL) methods measure the changes in radio signal on links in a static wireless network and process the data to estimate a person's location within the deployment area. Researchers have demonstrated the ability of DFL systems to locate and track people with less than one meter error [1], [2], [3], [4], [5], [6], [7]. One advantage of using radio waves is their ability to penetrate walls and other objects, allowing for more flexibility in placing the sensors in an area of interest [8], [9], [10].

In fingerprint-based DFL, training data in the form of signal measurements of each link are gathered as a person walks to each of several predetermined locations in the network [11], [12], [13], [14], [15], [16], [17]. These measured "fingerprints" gathered during the training session are stored with the known (or even estimated [18]) coordinate of the person. A supervised learning algorithm then processes the test data, a new measurement on each link, to determine which fingerprint in the database to which it is closest, or to estimate a person's location as a linear combination of several fingerprints.

Fingerprint-based DFL methods are effective compared to other model-based methods because the changes in a radio channel as a function of a person's position are often difficult to model in multipath environments. The signal power on any given channel on a link typically decreases when a person obstructs the line between a transmitter and receiver (the *link line*) in a line-of-sight environment. However, in a non-line-of-sight environment, this power can increase or decrease when the link line is blocked [19]; moreover, the signal power can change even when a person is far from the link line [20]. Measuring the exact mapping between person position and received signal change for a particular environment enables position estimation from signal measurements.

Within the broad area of "device-free" environmental sensing, this paper studies DFL systems which use received signal strength (RSS) as a link signal measurement, which includes the majority of DFL research. RSS is a nearly ubiquitous measurement available from standard transceivers. Recent work has shown that use of channel state information (CSI), i.e., the signal strength and phase of the received signal at multiple discrete frequencies, provides a higher dimensional measurement per link which can then improve DFL performance [16], [17]. Although we do not measure phase, this paper also studies measuring the signal strength at multiple discrete frequencies and its effect on DFL. Related to DFL, other work has shown the capability of bistatic or multistatic radars to localize people through walls [9]. Further, radio channel measurements on static links can be used to estimate the activity in which a person engaged [21] or their gesture [22]. Although environmental changes may also impact them, this paper does not address radar measurements or recognition of activity or gesture, and is instead focused on localization using standard transceivers.

Over time, the mapping between received signal change and a person's position changes as the environment changes. Fig. 1 shows an example of the RSS function (vs. a person's location), taken from Experiment Set 2, described in Section II. In the figure, the RSS function is first measured (blue line). Then, after 19 randomized small changes are made to objects in the environment, the RSS function is remeasured (green line). In some cases, for the same position and same link, the RSS is different by 5-10 dB. The change in RSS is a phasor sum of multipath, and as the phases of the multipath change due to small changes in the positions of reflectors in the environment, so does the way that the RSS responds to a person's presence. Such changes to the RSS mean that the fingerprints measured in training are no longer accurate, and as a result, the accuracy of fingerprint-based DFL degrades.

However, degradation in localization performance over time in a changing environment have rarely been presented in the literature. In [11], the authors perform tests in an apartment one month after the training data were gathered. The results

B. Mager and N. Patwari are with the Dept. of Electrical and Computer Engineering, University of Utah, Salt Lake City, USA. P. Lundrigan is with the School of Computing, University of Utah. N. Patwari is also with Xandem Technology, Salt Lake City. This material is based upon work supported by the National Science Foundation under Grant Nos. #1315685 and #1407949.

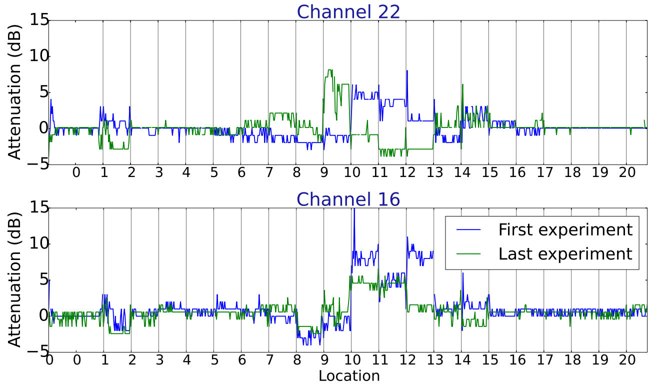

Fig. 1. Attenuation on two channels for link 19→25 vs. person location (0-20). The first attenuation function (——) is measured; then several small changes are made to the environment, and the attenuation function is re-measured (——). In some locations, the attenuation has changed significantly due to the environment changes.

degrade due to the environmental change, and the authors correct for the change by removing links that experienced a large variation from the training data due to environmental instabilities. One concern with the link removal method is that, over time, fewer and fewer links are used for localization; eventually, there may be insufficient information for localization.

Methods have been developed to track the reference or "unaffected" value of each link's signal as the environment changes. In [23], a radio tomographic imaging (RTI) system estimates where the person is currently located and updates the reference RSS for links far away from the current location. Other methods train hidden Markov models to learn the distribution of RSS during affected and unaffected states [4], [24]. Fingerprint-based DFL systems require knowing the fingerprints as a function of the known person location. In [15], a video camera is used to provide ground truth person location, at the price of having a duplicate localization method alongside the DFL system.

In this paper we examine the degradation in localization accuracy over time, and test various methods to mitigate the decrease in accuracy. We conduct the study by performing extensive measurements in a residential home in which a network of RF sensor nodes has been deployed. Environmental change is systematically introduced by moving, at random, various objects within the network. After each item is moved, a new localization experiment is performed. We then employ a fingerprint-based method using a machine learning classifier to estimate the subject's location for each measurement sample.

Four machine learning classifiers are tested to determine their relative performance over the course of the experiments. We further investigate channel selection as a means for robustness. We find that localization performance varies significantly by classifier and fingerprint link features used. For example, after 18 changes in the environment have been made for one experiment set, classification error ranges from 12.8% to 47.9% for the four different classifiers. The random forests classifier consistently outperforms others tested. We then introduce a novel correlation metric for selecting the channel to be used as a link's feature, and show that it achieves consistently high classification accuracy as the environment changes, reducing average (over all experiments) classification error from 3.5% to 1.4%.

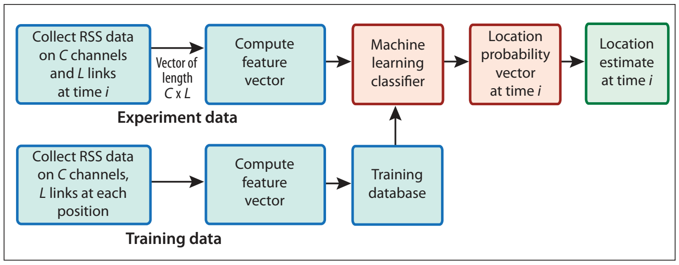
Collect RSS data on C channels and L links at time i
Vector of length C x L
Compute feature vector
Machine learning classifier
Location probability vector at time i
Location estimate at time i
Experiment data
Collect RSS data on C channels, L links at each position
Compute feature vector
Training database
Training data

Fig. 2. Algorithm framework for localizing a subject.

A. DFL Problem Statement

A DFL system is composed of N sensor nodes deployed around the area of interest. In this paper, we consider measurements of received signal strength (RSS) on many links in the network. We assume a protocol exists to allow many links to be measured, and that each transmitter/receiver pair measures RSS on multiple frequency channels. Let link l be the directional link between transmitter t_l and receiver r_l, where $t_l \neq r_l$ and $t_l, r_l \in \{1, \ldots, N\}$. Next, let $s_{l,c}$ be the RSS measured on link l on frequency channel $c \in \{1, \ldots, C\}$ where C is the total number of possible channels. A complete measurement of the network's L transmitter/receiver pairs and C channels is given as,

$$\mathbf{s} = [s_{1,1}, \ldots, s_{1,C}, \ldots, s_{L,1}, \ldots, s_{L,C}]^T. \quad (1)$$

Training data are first gathered by having a person stand at several predetermined locations throughout the environment, so as to cover as much of the area as possible. Let $\mathcal{I}$ be the set of training locations. We collect several vectors $\mathbf{s}^i$ as given in (1) for each location $i \in \mathcal{I}$.

A fingerprint may be, in general, a subset or a function of the data in $\mathbf{s}^i$. For example, in order to decrease the dimension of the fingerprint, the fingerprint may extract RSS measurements made on particular channels. We assume some function $\mathbf{u}^i = f(\mathbf{s}^i)$ computes a fingerprint vector $\mathbf{u}^i$ which is then stored in a database.

During system operation, when a person is standing in the environment, a new vector $\mathbf{s}$ is measured as in (1), with corresponding feature vector $\mathbf{u} = f(\mathbf{s})$. The fingerprint-based DFL problem is to determine in which location $i \in \mathcal{I}$ the person is standing, as shown in Fig. 2.

In this paper, we first present an experimental method to evaluate the performance of fingerprint-based DFL methods after environmental changes have made RSS feature vectors diverge from those stored in the training database.

II. Methods

In this section, we describe our method for providing a repeatable experimental evaluation of the performance of a fingerprint-based localization system in an environment with a quantifiable degree of change. One method is to test DFL performance after a long period of time after training [11]; however, the reader will not be aware of what has changed in that time, or even how much the environment was used in the interim. As an alternative, we use a procedure in which we repeatedly make small changes to objects in the environment and collect a new set of RSS data after each change. First, we describe the network, the sensors, the environment, and the data collection methods. Then, we describe the method we use to introduce random changes into the environment.

A. Environment

The experiments are conducted on the main floor of a brick house, in an area approximately 84 m^2. The main floor area includes living room, dining room, kitchen, bedroom, den, and two bathrooms. Thirty-two locations are identified and marked by numbers on the floor, as shown in Fig. 3. Neighboring locations are separated by about 50 cm, although furniture and walls prevented a complete grid of testing locations.

B. Network

A network of $N = 30$ transceivers is deployed in vertical pairs at fifteen locations, with the lower nodes approximately 28 cm from the floor and the upper nodes 132 cm above the floor. Placing nodes at two levels provides an opportunity to test different topologies of nodes based on height. The nodes are Texas Instruments CC2531 dongles, which operate in the 2.4 GHz range.

A multichannel system with a TDMA protocol is employed, where each node takes a turn transmitting and all the nodes switch to the next channel of a predefined set of C channels after each transmission cycle. The nodes can transmit and receive on any of 16 channels numbered 11–26, where the center frequencies f_c are given by,

$$\mathbf{f_c} = 2405 + 5(k - 11)MHz, k \in [11, 26] \qquad (2)$$

For this research, data are gathered on the $C = 8$ even-numbered channels (i.e., $12, 14, \ldots, 26$), allowing us to test a set of evenly-spaced center frequencies over the available range. In the deployed network, we have full connectivity, such that data are measured on $L = N(N - 1) = 870$ links. An additional receiver listens to the traffic on the network and collects the data for post-processing.

C. Training and Test Sets

a) Training: In our experimental procedure, we first gather training data. The subject stands at testing location i for 50 s while RSS vectors $\mathbf{s}^i$ are collected. To ensure the training data are not specific to one orientation of the person, the subject slowly spins around in place, making one rotation over the course of the 50 s. Given that we collect one sample on all eight channels every 0.817 s, we have about 61 samples of training data. Note that 50 s of calibration data is gathered with the subject standing outside the house. We call this location $i = 0$, which provides a baseline measurement of RSS on all the links with no one in the network.

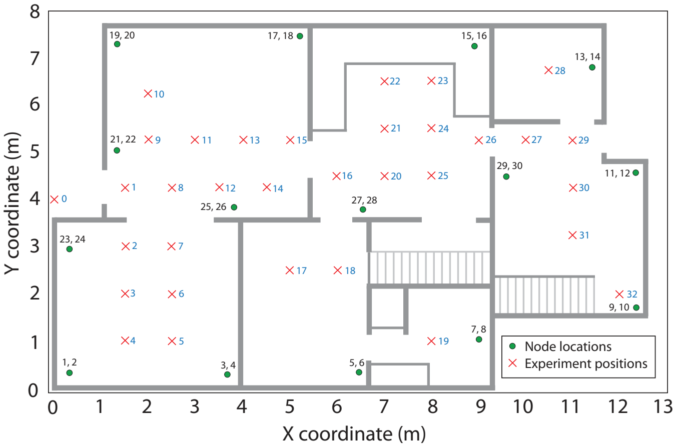

Fig. 3. House layout with node locations and test locations. The nodes are deployed in vertical pairs; lower node numbers are odd, upper node numbers are even.

b) Test Experiments: The next phase consists of several experiments in which the subject moves from one location to the next, standing still for a minimum of 20 s in each position. The order and timing of the movements are predetermined, and the subject uses a stopwatch to know when to move to each spot. This period of time provides about 24 samples of RSS vector s for each position. Data recorded during movement from one location to the next are not used since the exact position of the person is not known. Note that location $i = 0$ is also tested, for a minimum of 50 s.

Thus, the term “training” refers to the procedure of gathering the RSS signature with a subject standing in every location; “experiment” refers to a localization test; data are collected for the purpose of having the system estimate where a person is; and “calibration” refers to the collection of data when the network is empty.

D. Simulating the Passage of Time

We use a procedure, described here, to randomly change the environment in a way that might be done by residents of a home, but in a manner that allows repeatability of the experiments.

First, we identify several moveable objects and manner of movements (e.g., direction and distance). Objects include a coat rack, couch, box of books, bag of groceries, house plant, ironing board, dining set, and two interior doors.

Next, each object is given a list of possible placements, up to four possible locations or states. For example, the bag of groceries could be placed in four possible locations on the kitchen counter; the stack of boxes is assigned four possible positions, one in each of the four main rooms (living room,

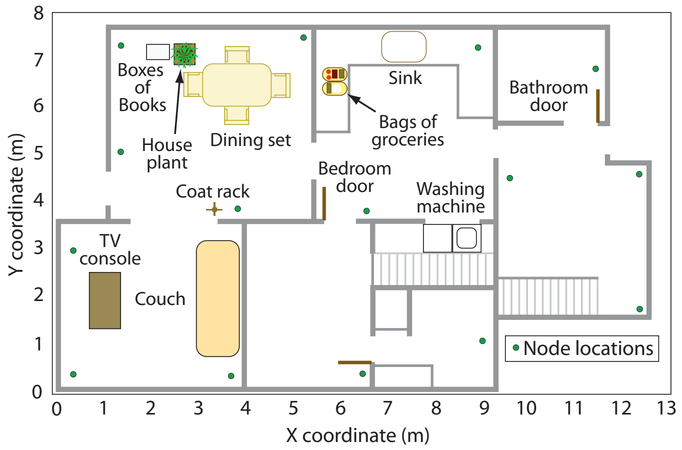

Fig. 4. House layout with node locations and default placement for objects.

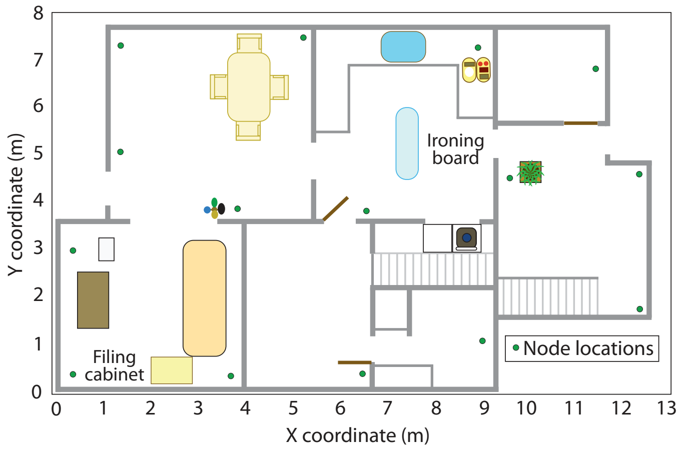

Fig. 5. House layout with node locations and final placement of objects for Sets 4 and 5.

dining room, kitchen, and den); each chair in the dining set could be placed facing one of four orthogonal directions. One of these positions is assigned as the default position, in which the object is placed during training.

Finally, to reduce the chance of experimenter bias, when something is to be moved, the object and its new placement are randomly generated by computer. A complete list of the chosen objects and placements are listed in the appendix of [25].

Each set of experiments consists of several separate data-gathering experiments. Each set begins with a training and an experiment. Next, an object is moved or changed, and a new experiment is conducted. This process is repeated, with the next object being moved and new experiment conducted. Changes are cumulative over the set of experiments.

Five sets of experiments are performed. Sets 1 and 2 include minor movements, such as slightly rotating a chair; however, later experiment sets include only more significant movements that would represent a greater change in the environment and thus potentially lead more quickly to a measurable degradation in localization performance. For example, in Set 3 each experiment involves moving two items at a time, while in Sets 4 and 5 the dining table and chairs form a group that is rotated or moved together, rather than as individual pieces. Figures 4 and 5 show the "before and after" of objects in the environment for Sets 4 and 5. A complete sequence of object movements in each set is listed in [25].

1) Classifiers Tested: Four machine learning classifiers are tested in this paper: random forests with 100 estimators, k-nearest neighbors (KNN) using three neighbors, support vector machine (SVM), and linear discriminant analysis (LDA).

SVM is a supervised learning algorithm that uses labeled training data to create a model that can then predict which classes the new test data belong to, given a set of features from the test data [26], [27]. SVMs treat the data to be classified as points in a multidimensional space, in which the different classes can be separated by a hyperplane. SVMs have been used in fingerprint-based DFL [12].

The SVM used here employs a radial basis function (RBF) kernel to project data to a higher-dimensional space, where the RBF kernel on two samples x and x' is defined as

$$K(x, x') = \exp(-\gamma |x - x'|^2), \tag{3}$$

where γ is a kernel size parameter. The SVM also allows for a few anomalous data points falling on the incorrect side of a hyperplane by the inclusion of a user-defined penalty parameter, C, which determines how much the outliers are taken into account. Finding the optimal values of γ and C is accomplished through a grid search that involves testing several pairs of each parameter and choosing the pair that results in the lowest error rate. From our data, we conclude that the value of $\mathbf{C} = 0.95$ and $\gamma = 5 \times 10^{-4}$ provide the best performance.

The KNN algorithm is an instance-based learning method, where learning simply involves storing the training data for later retrieval, and each instance (i.e., new test sample) is assumed to correspond to a point in n-dimensional space [28]. When a new test sample arrives, the algorithm examines the new sample's Euclidean distance to the stored samples and assigns a value for this sample. In the case of k-nearest neighbors, the algorithm compares the test sample to a user-defined number of training samples, k, that are closest in distance to the test sample. The value of the test sample is then assigned through a simple majority vote of its k nearest neighbors. For our experiments, we use a value of k = 3.

The random forests classifier consists of a collection of single classification trees, in which each tree is grown by randomly drawing samples, with replacement, from the training set [29]. To our knowledge, random forests have not been used for DFL, and we include them because they are known to achieve state-of-the-art results in several other applications [30]. A small number of features is selected at random out of the input features, and the best split on these is used to construct each tree. Trees are then combined by averaging their probabilistic prediction [27]. For these experiments, the number of trees is set to 100, while the number of random subsets of features to consider when looking for the best split on each node is set to the square root of the number of features. Although performance improves for increasing number of trees, we see diminishing returns after 100.

Use of LDA has been tested for DFL in [11], [14], and [18], and is closely related to a Bayesian classifier with Gaussian-distributed observations such as used in [17]. It is a type of classifier that finds linear decision boundaries between classification regions that divide the input space [31]. Each class density, $P(\mathbf{u}|i)$, i.e., the probability density of RSS fingerprint vector $\mathbf{u}$ given a person at location i, is modeled as multivariate Gaussian, and classes are assumed to share a common covariance matrix. Given $P(\mathbf{u}|i)$ for each class (location) i, LDA finds the probability that a particular RSS fingerprint $\mathbf{u}$ belongs to a class by applying Bayes' rule:

$$P\left[i|\mathbf{u}\right] = \frac{P\left[\mathbf{u}|i\right] P\left[i\right]}{P\left[\mathbf{u}\right]}. \tag{4}$$

For our implementation, we assume locations i are equally likely. Location classification is performed by finding the i that maximizes (4), or equivalently, $P\left[\mathbf{u}|i\right]$.

III. Results

In this section, we address particular design questions in RSS-based DFL, in particular, how performance degrades over time as the environment changes. We do not focus on the value of the error percentage or RMSE — we are primarily interested in the *relative* performance among a set of possible design choices, or over time.

A. Selecting a Machine Learning Classifier

We first explore machine learning classifier performance. We use the data from experiment Set 2, since it is the most representative of the data sets. In Section III-H, we test the accuracy of the other experiment sets once a classifier and feature set have been selected.

For this evaluation, we set the feature vector $\mathbf{u}$ to contain the link attenuation values measured on only one channel c, for some fixed channel c; we refer to this as the One Fixed Channel method. Specifically,

$$\mathbf{u} = \left[s_{1,c}, s_{2,c}, \ldots, s_{L,c}\right]^T. \tag{5}$$

Measuring on one fixed channel on all links has been reported in the fingerprint-based DFL literature [11], and this setting for fingerprint vector $\mathbf{u}$ is reasonable for comparing different classifiers. Since our network records data on $C = 8$ channels, we run the test with eight different settings for c and show the average performance.

For each setting of channel c, we build each classifier using the fingerprint vectors $\{\mathbf{u}^i\}_i$ recorded in the training period. Skewing of the class distribution is avoided by ensuring that we gather the same number of samples per class. Then, we test each classifier using the attenuation data vectors collected during each experiment in Set 2.

Fig. 6 shows the error rates, averaged over eight possible channels, for each method over time, i.e., over the course of the experiments, which are labeled from 1 to 19 in the graph.

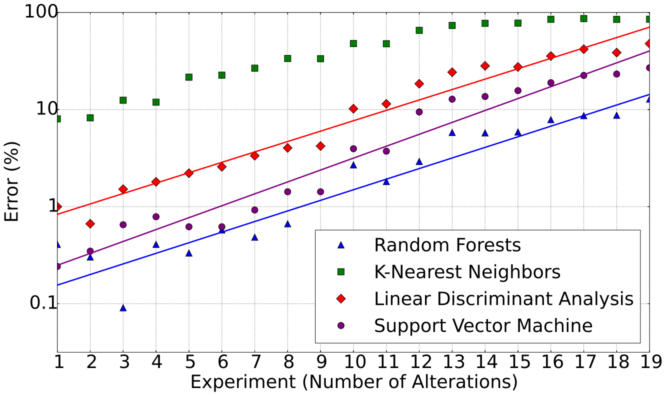

Fig. 6. Error percentage for classifier methods vs. Set 2 experiment number, with log-linear best fit for random forests, KNN, LDA, and SVM.

Classifier	Overall Average Error
Random Forests	3.5%
K-Nearest Neighbors	47.9%
Linear Discriminant Analysis	16.1%
Support Vector Machine	8.3%

TABLE I. Error rates from four classifiers

Note that one experiment was conducted after each of 19 environmental changes as described in Section II-D, so plotting vs. experiment number effectively plots performance vs. the quantity of change in the environment. Table I summarizes the classifiers' error rates, also averaged over the 19 experiments.

The k-nearest neighbors classifier performs poorly compared to the other methods. Even before any items have been moved in the house, at experiment 1, KNN's error rate of 8.0% is nearly an order of magnitude worse than the next highest error rate of 1.0% for LDA. Increasing k does not improve the error rate of KNN; for example, k = 5 results in an error rate of 9.7% for experiment 1.

The KNN classifier suffers from the curse-of-dimensionality problem in which as the dimension increases, the distance to the closest sample approaches the distance to the furthest data point [32]. The poor performance with a standard KNN classifier may be the result of having noisy data with a dimension of $L = 435$.

LDA also performs relatively poorly, perhaps because LDA assumes that 1) data is multivariate normal within each class and 2) all classes share the same covariance matrix. Although quadratic discriminant analysis removes the second assumption, it performs poorly when the number of training samples in each class is too small for accurate estimation of the conditional covariance matrices [11].

One advantage of random forests is that using a random split selection when building the tree classifiers makes the algorithm more robust to noise, since it does not put weight on any particular subset of the instances [33]. If we were to retrain the system after a change in the environment, we might find the feature vectors associated with some locations to have some altered values. We could thus think of the original training data as containing noise, i.e., some of the data will be

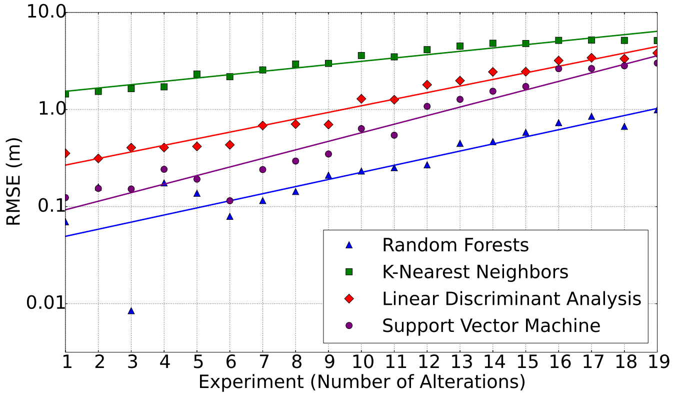

Fig. 7. Penalized root-mean-square error with a penalty of 6 m, for classifier methods vs. Set 2 experiment number, with log-linear best fit.

misclassified since some pairs of feature vectors and labels no longer correspond to what the new training data would show.

B. Distance Error

If we present classifier performance in terms of distance, as is standard in the literature, we find that the random forests classifier still performs the best. Fig. 7 shows the results for all classifiers for Set 2 using penalized RMSE. A penalty is included for cases where the system decides that no person is present, when in fact a person is at a location inside the house. In these cases, we introduce a penalty of 6 m, which is approximately half the length of the house. Comparing this figure with Fig. 6 informs us that while the number of errors increases for all four classifiers over time, those for random forests are due to misclassifications nearer to the subject's actual location.

Since the random forests classifier provides the best overall performance, we will use that for the remainder of the discussions.

C. Error % vs. Environmental Change

The data from the classifier tests plotted in Figs. 6 and 7 show that when system performance is good ($\leq$ 10% or RMSE $<$ 1 m), the rise in percentage error and RMSE is approximately exponential with experiment number. That is, the error metric increases multiplicatively with linear increases in the number of environmental changes. Further, the rate of increase is nearly the same for random forests, LDA, and SVM. For low values, error percentage is approximately proportional to $\exp(0.113x)$ where x is the experiment number, and RMSE is proportional to $\exp(0.176x)$. This corresponds to error percentage and RMSE doubling every 6.1 and 3.9 experiments, respectively.

This experimental result may help predict, when an RMSE result is reported with no environmental changes between training and test, how the error will increase over time.

Further, the good fit between the exponential fit and the experimental results shows that it is not just a few specific changes in the environment that result in the observed performance degradations. For example, rotating a chair in Set 1 resulted in a small decrease in error, while rotating the same chair for an experiment in Set 2 caused a large increase in error. Closing the bedroom door in Set 1 increased the error by nearly 4%, yet in Set 2 the same change increased the error by only about 1%. Thus, the same change can degrade performance strongly in one instance and make little or no difference in another instance. We find that performance degradations are unpredictable individually, but predictable on their collective effect over time.

Link Set	Overall Average Error
All links	3.5%
Upper links	5.9%
Lower links	26.8%
Diagonal cross links	7.8%
All w/o lower links	3.6%

TABLE II. ERROR RATES FOR DIFFERENT LINK SETS

D. Link Sets Based on Network Topology

At this point we can investigate whether it is possible to find a feature set that will help improve localization performance even though the environment changes. Including links that have been too greatly affected could lead to a misclassification by the machine learning algorithm. Thus, we want to only include data from the links and channels that remain more constant throughout the course of any changes in the environment.

We begin by looking at the results of different link sets based on topology. Typical experimental setups consist of nodes placed at a height of 1.0–1.5m above the floor. While we employ this arrangement (referred to as the upper level here), we also include a lower level of nodes near the floor. This setup can potentially be used in applications such as fall detection [34]. Thus, we can select from the upper links (created by nodes on the upper level of the network), the lower links, the diagonal cross links (all the links traversing diagonally from the lower level to the upper level or vice versa), or all links minus the lower links.

Primarily, changes in the environment are due to moving objects and furniture, which typically sit on the floor. Thus we expect the results generated by the lower links to degrade the most. Indeed, this is what we find. However, we also want to determine if using one of the other link sets will provide localization performance that equals or surpasses that of using all the links.

Using the same fingerprint vector $\mathbf{u}$ as used in Section III-A and the random forests classifier, we compute the error rate for each of the link sets. The average over all experiments and channels is given in Table II.

We see that we cannot improve performance by selecting a link set based only on topology, though it is possible to remove the lower links and achieve results nearly equivalent to using all the links. For the remainder of the paper, we use all links when processing the data.

E. Selecting a Channel Set

The results in Sections III-A and III-D use only data from channel c in the fingerprint $\mathbf{u}$, as given in (5). However, for any particular link, two different channels may behave differently both before and after the environment changes. Our choice of channel or channels to use in fingerprint vector for a link will impact the performance of the DFL system. In this section, we explore the selection of one or more channels for each link.

At one extreme of fingerprint vector dimension, the system could use the data from all eight channels at once in the fingerprint vector, i.e., $\mathbf{s} = \mathbf{u}$. Using the random forests classifier for Experiment Set 2, this setting achieves an overall average error rate of just 0.185%, which is far better than the 3.5% error we get by processing the channels separately and averaging their results. We could conclude that, to minimize the degradation in localization performance over time, we simply need to gather data on multiple channels and process them together. Although the measurement now requires significantly more bandwidth, the result is motivation for use of devices that can measure CSI, which includes signal strength at each of multiple frequency channels (in addition to phase), such as reported in [17], [16]. Note that the eight channels we measure span 80 MHz of bandwidth, while WiFi channels at 2.4 GHz span only 20 MHz.

If we use devices which report signal strength on a single frequency, measuring eight channels requires eight times the energy and latency. Sending data on C channels would multiply the data rate sent from each transceiver to the sink by C, and would increase the computational requirements at the processor. For a real-time embedded DFL system, such an increase is undesirable. In the following sections, we determine if we can choose a subset of the channels to process, while still achieving acceptable error rates.

1) Subsets of Channels: We test processing the data using a subset of the eight available channels. That is, for a subset $\mathcal{A} \subset \{1, \ldots, 8\}$ with elements $c_1, \ldots, c_{|\mathcal{A}|}$, we set the fingerprint vector,

$$\mathbf{u} = [s_{1,c_1}, \ldots, s_{1,c_{|\mathcal{A}|}}, \ldots, s_{L,c_1}, \ldots, s_{L,c_{|\mathcal{A}|}}]^T.$$

Testing every possible subset $\mathcal{A}$ and averaging the results of the sets with the same number of channels $|\mathcal{A}|$ indicates that, as expected, the average error is higher with fewer channels (Table III).

Note that for any particular number of channels, some subsets of channels provide excellent results. Fig. 8 shows an example in which the performance of channels $\{22, 20, 16\}$ is as good, on average, as using six channels. The difficulty lies in determining beforehand which are the "right" channels to use.

F. Best Channels Per Link

In this section, we explore selecting, based on information available to us from the training and calibration, a single channel for each link that we predict will be best for fingerprint-based DFL. Denoting b_l as our channel selection for link l, we set the fingerprint vector to,

$$\mathbf{u} = [s_{1,b_1}, s_{2,b_2}, \ldots, s_{L,b_L}]^T. \tag{6}$$

Number of Channels	Overall Average Error
1	3.50%
2	1.80%
3	1.10%
4	0.750%
5	0.547%
6	0.413%
7	0.329%
8	0.185%

TABLE III. ERROR RATES USING VARIOUS SUBSETS OF CHANNELS

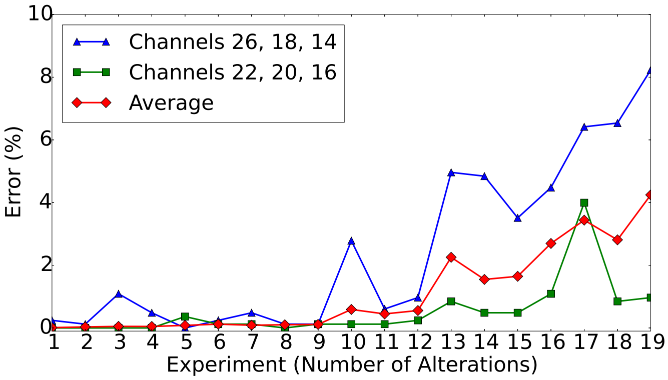

Fig. 8. Different combinations of three channels. Depending on the channels selected, the results can vary dramatically.

In contrast to (5), this also uses one channel's data per link, but allows each link to use an arbitrary channel.

1) Best Channels – Highest Average RSS Methods: First, we explore using, for link l, the channel b_l with highest average RSS. This is motivated by the fact that in a multipath channel, as a function of center frequency, a link may experience low RSS because of destructive interference (said to be in "deep fade") or high RSS because of constructive interference (called "antifade" links in [19]). A link in an antifade generally experiences a sharp drop in RSS due to a change directly on the link line, and little variation otherwise. In contrast, a link in a deep fade experiences high variance when there is change in a wide area near the link line [19]. Deep fade links are, in short, very sensitive to environmental change, and thus should be avoided.

We test two methods to using highest average RSS to select a channel for each link:

1) *Channel with Highest RSS During Training* method: From the training data collected when no person is present, select the channel on each link that has the highest average RSS, then use that channel for all experiments.
2) *Channel with Highest RSS Adaptive* method: Adaptively select the channel on each link that has the highest average RSS during each calibration period. In our experiments, we have a calibration period before each experiment during which we know the area to be empty. In a real-time system, one would need to use a method such as used in [35], [23], [4], [24] to adaptively track the calibration RSS.

In the Channel with Highest RSS Adaptive method, the channel b_l used during training to generate fingerprint vectors $\{\mathbf{u}^i\}$ may later be changed. Denote $b_l(m)$ as the channel with highest calibration RSS at time m on link l. A new calibration is collected (and m is incremented) whenever the link is determined to be not affected by any person's presence. The fingerprint vector sent to the classifier at time m will be,

$$\mathbf{u} = [s_{1,b_1(m)}, s_{2,b_2(m)}, \ldots, s_{L,b_L(m)}]^T, \tag{7}$$

even though b_l used during training may be not equal to $b_l(m)$. In effect, we replace the data from channel b_l with the data from a new channel $b_l(m)$ which we believe is now closer in attenuation characteristics to the training data for link l.

Using the Channel with Highest RSS Adaptive method results in an error rate of 1.25%, compared with a 3.50% error rate using the One Fixed Channel method, and a 6.89% error rate using the Channel with Highest RSS During Training method. The rationale for this is that links with channels with the highest average RSS when the network is empty are less likely to be in deep fade, and thus more likely to be affected only when someone is standing on the link. In terms of complexity, determining b_l from training is preferable compared to adapting b_l; however, the results show it is not robust to environmental change.

2) Best Channels – Correlation Method: Next, we test the idea that if two channels have highly correlated RSS during training then they are also likely to be robust to environmental change. To motivate this, consider that a link l's RSS $s_{l,c}$ is the squared amplitude of a phasor sum of multipath, in dB:

$$s_{l,c} = 10 \log_{10} \left| \sum_i \alpha_i e^{j 2\pi f_c \tau_i} \right|^2, \tag{8}$$

where f_c is the center frequency of channel c, α_i is the complex amplitude and τ_i is the propagation delay of path i, and $j = \sqrt{-1}$. We suspect that many small changes in the environment alter the time delays $\{\tau_i\}$ because the changes make a path from transmitter to receiver slightly longer or shorter. These changes in τ_i alter the phases $e^{j2\pi f_c \tau_i}$. Similarly, changing the center frequency alters the phases $e^{j2\pi f_c \tau_i}$. If we find a frequency f_c near which the amplitude of the phasor sum (and thus $s_{l,c}$) changes very little, then we hypothesize that $s_{l,c}$ will change very little due to changes in $\{\tau_i\}$.

In this correlation method, we first find for link l the Pearson correlation coefficient between the RSS measured during training on any pair of channels. We denote the two channels with highest correlation coefficient as $\gamma_{l,1}$ and $\gamma_{l,2}$. We set b_l to be the channel $\gamma_{l,1}$ or $\gamma_{l,2}$ with the highest average RSS over the course of the training experiment.

We set b_l in (6) in this manner and run the random forests classifier and plot results for Experiment Set 2 in Fig. 9. Using this correlation method provides better results than the One Fixed Channel method, reducing the overall average error rate from 3.5% to 1.4%.

The correlation method for channel selection has similar performance to the Channel with Highest RSS Adaptive method, but yet does not require adaptive tracking of the calibration RSS. Further, it is significantly better than the Channel with Highest RSS During Training method, which also relied only on the training period RSS data.

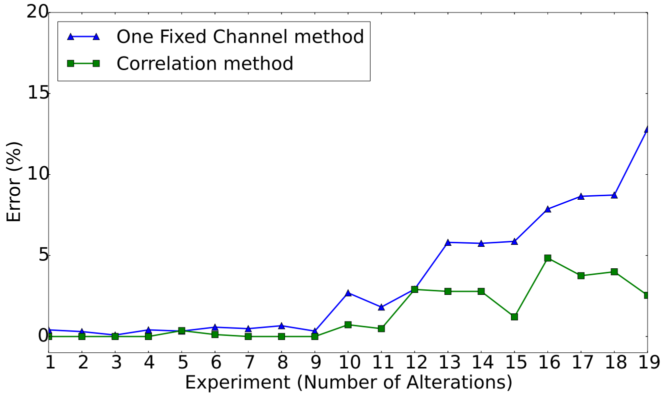

Fig. 9. Error rates for Set 2, comparing the One Fixed Channel method with the Correlation method of selecting the best channels.

G. *Analysis of the Correlation Method*

Analysis of the two most correlated channels $\gamma_{l,1}$ and $\gamma_{l,2}$ over all the experiment sets indicates that 96.0% of these pairs consist of adjacent channels (e.g., channel pairs such as 14 and 12, 22 and 20, 26 and 24, since we are only using the even-numbered channels). One might expect to find that channels near each other would behave similarly. However, it does not appear that lower frequencies perform better than higher frequencies, or vice versa. The specific channels chosen are evenly distributed among the eight possible channels, so there is no bias towards a particular center frequency.

As an example, Fig. 10 compares the attenuation on the link between nodes 24 and 28 for the initial and final experiments. Channel 14 is selected by the Channel with Highest RSS During Training method, and channel 24 is selected by the correlation method. Another example is in Fig. 1, showing a different link (19 to 25) for which the Channel with Highest RSS During Training method selects channel 22 and the correlation method selects channel 16. Both links are typical of links in which the two methods do not select the same channel. We note that the correlation method often selects a channel that experiences more signal, i.e., change in RSS as a function of position. This is seen in Fig. 10 at locations 2, 5, and 17, compared with the lack of such a change in RSS in the upper graph. Second, when the channel selected by the correlation method does show a high signal, it is often more consistent over time, compared to locations for which the highest average RSS method shows a high signal.

H. *Testing the Repeatability of the Results*

Having identified, for one set of the experiments, a machine learning classifier that offers the best localization performance for the system under investigation, as well as feature sets that are relatively robust to environmental change time, we now need to determine if the results are repeatable with other experiment sets. Five sets of experiments are conducted in the same house over the course of two weeks. Each set consists of several experiments in which one item is moved before each

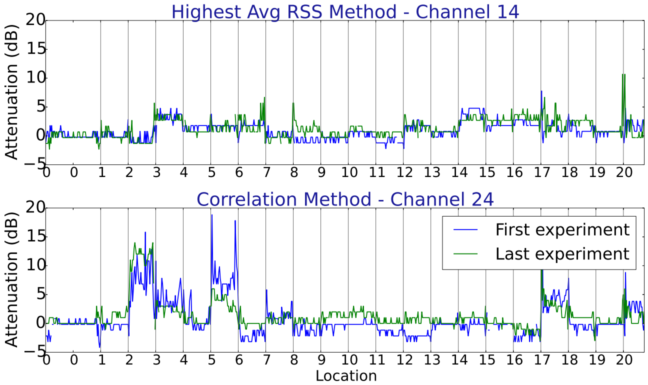

Fig. 10. Attenuation for the two channels selected by different methods, for the link between nodes 24 and 28.

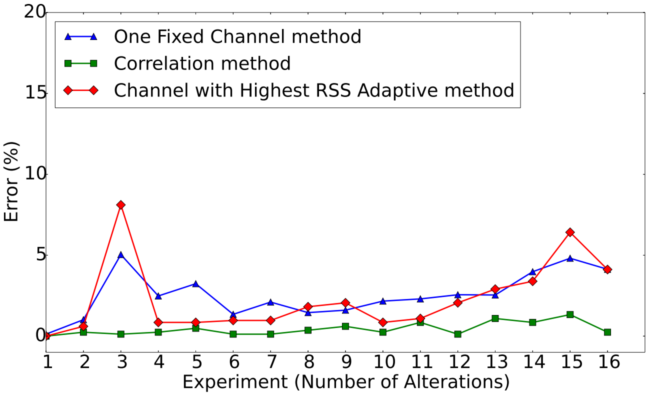

Fig. 11. Error rates for Set 1, comparing all channels with two methods of selecting the best channels.

experiment, and all experiments in a set are conducted on the same day.

Fig. 11 shows the results of experiment Set 1, using the random forests classifier. We compare:

1) One Fixed Channel method,
2) Correlation method, and
3) Channel with Highest RSS Adaptive method.

Table IV summarizes the error rates for each experiment set using the three methods. The correlation method performs the best across the five experiment sets, with a 1.6% overall average classification error, 2.6 to 3.0 times lower than the other methods. We note that the highest RSS adaptive method was better than the correlation method only in Set 2. As the correlation method is easier to implement, we can conclude that, of the methods to select one channel per link, the correlation method is recommended.

IV. Conclusion

Experiment Set	Correlation	One Fixed Channel	Highest RSS Adaptive
Set 1	0.44%	2.56%	2.32%
Set 2	1.40%	3.50%	1.25%
Set 3	2.05%	8.68%	8.68%
Set 4	0.35%	3.30%	2.21%
Set 5	3.85%	5.79%	6.50%
Overall	**1.62%**	**4.76%**	**4.19%**

TABLE IV. Error rates for different sets and methods

This paper investigates the degradation of fingerprint-based DFL accuracy as a function of the degree of change in an environment. A network of sensor nodes on the main floor of a residential home is used to gather link RSS data on multiple channels on all the links in the network. We use a procedure in which we perform a sequence of randomly-selected environmental changes and conduct new localization test experiments after each change. We conduct five sets of such change-over-time experiments, where each set consists of 10 to 19 changes and re-testing.

This paper uses the extensive data to test how fingerprint-based DFL methods degrade over time, and how they may be made more robust. When the classification error percentage is less than 10%, it increases exponentially, doubling, on average, after 6.1 random changes in the environment. The random forests classifier is shown to be more reliable than three other classifiers tested. We next test different methods to select the channel, as a function of the link, to be used in the fingerprint vector. We propose a correlation method for channel selection which, in combination with the random forests classifier, helps achieve a much lower localization error rate even as the environment changes. The correlation method decreases localization classification error from an average of 4.8% over all the experiment sets to 1.6%. This demonstrates that we can develop a more robust localization system that requires less frequent retraining.

Brad Mager recently graduated with an M.S. in Computer Engineering at the University of Utah. His master's thesis research is in the area of device-free localization systems. He has a B.S. in Computer Engineering, also from the University of Utah. His undergraduate thesis involved research on using wireless sensor networks for fall detection.

Philip Lundrigan received the B.S. degree (with honors) in Computer Engineering from Brigham Young University in 2012. He is pursing the Ph.D. degree in Computer Science from the University of Utah. His current research interests include mobile networks, mobile computing, and wireless network management.

Neal Patwari received the B.S. (1997) and M.S. (1999) degrees from Virginia Tech, and the Ph.D. from the University of Michigan, Ann Arbor (2005), all in Electrical Engineering. He was a research engineer at Motorola Labs, Florida, between 1999 and 2001. He is an Associate Professor in the Department of Electrical and Computer Engineering at the University of Utah. He directs the Sensing and Processing Across Networks (SPAN) Lab, which performs research at the intersection of statistical signal processing and wireless networking. Neal is also the Director of Research at Xandem Technology. He has received best paper awards from the IEEE Signal Processing Magazine (2009), SenseApp (2012), and IPSN (2014), and the 2011 University of Utah Early Career Teaching Award.

References

[1] D. Zhang, J. Ma, Q. Chen, and L. M. Ni, "An RF-based system for tracking transceiver-free objects," in *IEEE PerCom*, 2007, pp. 135–144.

[2] X. Chen, A. Edelstein, Y. Li, M. Coates, M. Rabbat, and M. Aidong, "Sequential Monte Carlo for simultaneous passive device-free tracking and sensor localization using received signal strength measurements," in *ACM/IEEE IPSN*, April 2011.

[3] O. Kaltiokallio and M. Bocca, "Real-time intrusion detection and tracking in indoor environment through distributed rssi processing," in *2011 IEEE 17th Intl. Conf. Embedded and Real-Time Computing Systems and Applications (RTCSA)*, vol. 1, Aug. 2011, pp. 61 –70.

[4] Y. Zheng and A. Men, "Through-wall tracking with radio tomography networks using foreground detection," in *IEEE WCNC*, April 2012, pp. 3278–3283.

[5] O. Kaltiokallio, M. Bocca, and N. Patwari, "Enhancing the accuracy of radio tomographic imaging using channel diversity," in *9th IEEE International Conference on Mobile Ad hoc and Sensor Systems (MASS 2012)*, October 2012.

[6] M. Moussa and M. Youssef, "Smart services for smart environments: Device-free passive detection in real environments," in *IEEE PerCom-09*, 2009, pp. 1–6.

[7] C. Xu, B. Firner, R. S. Moore, Y. Zhang, W. Trappe, R. Howard, F. Zhang, and N. An, "Scpl: indoor device-free multi-subject counting and localization using radio signal strength," in *Proceedings of the 12th International Conference on Information Processing in Sensor Networks*, April 2013, pp. 79–90.

[8] J. Wilson and N. Patwari, "See through walls: motion tracking using variance-based radio tomography networks," *IEEE Trans. Mobile Computing*, vol. 10, no. 5, pp. 612–621, May 2011, appeared online 23 September 2010.

[9] F. Adib and D. Katabi, "See through walls with Wi-Fi!" in *ACM SIGCOMM'13*, Aug. 2013.

[10] D. Maas, J. Wilson, and N. Patwari, "Toward a rapidly deployable rti system for tactical operations," in *8th IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp 2013)*, Oct. 2013.

[11] C. Xu, B. Firner, Y. Zhang, R. Howard, J. Li, and X. Lin, "Improving RF-based device-free passive localization in cluttered indoor environments through probabilistic classification methods," in *Proc. Information Processing in Sensor Networks (IPSN-2012)*, April 2012, pp. 209–220.

[12] F. Viani, P. Rocca, M. Benedetti, G. Oliveri, and A. Massa, "Electromagnetic passive localization and tracking of moving targets in a WSN-infrastructured environment," *Inverse Problems*, vol. 26, pp. 1–15, March 2010.

[13] M. Seifeldin and M. Youssef, "Nuzzer: A large-scale device-free passive localization system for wireless environments," Arxiv.org, Tech. Rep. arXiv:0908.0893, Aug. 2009.

[14] C. Xu, B. Firner, Y. Zhang, R. Howard, and J. Li, "Trajectory-based indoor device-free passive tracking," in *2nd Intl. Workshop on Mobile Sensing (IWMS)*, vol. 12, June 2012.

[15] C. Xu, M. Gao, B. Firner, Y. Zhang, R. Howard, and J. Li, "Towards robust device-free passive localization through automatic camera-assisted recalibration," in *ACM SenSys*, 2012, pp. 339–340.

[16] J. Xiao, K. Wu, Y. Yi, L. Wang, and L. M. Ni, "Pilot: Passive device-free indoor localization using channel state information," in *IEEE DCOSS*, 2013, pp. 236–245.

[17] H. Abdel-Nasser, R. Samir, I. Sabek, and M. Youssef, "Monophy: Mono-stream-based device-free wlan localization via physical layer information," in *IEEE WCNC*, 2013, pp. 4546–4551.

[18] A. Saeed, A. E. Kosba, and M. Youssef, "Ichnaea: A low-overhead robust wlan device-free passive localization system," *IEEE J. Sel. Topics Signal Processing*, vol. 8, no. 1, pp. 5–15, Oct. 2013.

[19] J. Wilson and N. Patwari, "A fade level skew-Laplace signal strength model for device-free localization with wireless networks," *IEEE Trans. Mobile Computing*, vol. 11, no. 6, pp. 947–958, June 2012, appeared online 12 May 2011.

[20] O. Kaltiokallio, M. Bocca, and N. Patwari, "A fade level-based spatial model for radio tomographic imaging," *IEEE Trans. Mobile Computing*, vol. 13, no. 6, pp. 1159–1172, Dec. 2013.

[21] S. Sigg, M. Scholz, S. Shi, Y. Ji, and M. Beigl, "RF-sensing of activities from non-cooperative subjects in device-free recognition systems using ambient and local signals," *IEEE Trans. Mobile Computing*, 2013, appeared online 21 Feb. 2013.

[22] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *19th Intl. Conf. on Mobile Computing & Networking*, Sept. 2013, pp. 27–38.

[23] M. Bocca, O. Kaltiokallio, and N. Patwari, *Radio tomographic imaging for ambient assisted living*. Springer, 2013, pp. 108–130.

[24] A. Edelstein and M. Rabbat, "Background subtraction for online calibration of baseline RSS in RF sensing networks," *IEEE Trans. Mobile Computing*, 2012, appeared online 3 Oct. 2012.

[25] B. Mager, "Maintaining accuracy of device-free localization systems in changing environments," Master's thesis, University of Utah, 2014.

[26] W. Noble, "What is a support vector machine?" *Nature Biotechnology*, vol. 24(12), p. 15641567, 2006.

[27] F. Pedregosa, "Scikit-learn: Machine learning in python," *JMLR 12*, p. 28252830, 2011.

[28] T. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[29] L. Breiman, "Random forests," *Machine Learning*, vol. 45 (1), p. 532, 2001.

[30] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *IEEE Computer Vision Workshops*, 2009, pp. 1393–1400.

[31] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[32] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is nearest neighbor meaningful?" in *Database TheoryICDT99*. Springer, 1999, pp. 217–235.

[33] T. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting and randomization," *Machine Learning*, pp. 1–22, 1998.

[34] B. Mager, N. Patwari, and M. Bocca, "Fall detection using RF sensor networks," in *IEEE Personal, Indoor and Mobile Radio Communications Conference (PIMRC 2013)*, Sept. 2013.

[35] O. Kaltiokallio, M. Bocca, and N. Patwari, "Follow @grandma: long-term device-free localization for residential monitoring," in *7th IEEE International Workshop on Practical Issues in Building Sensor Network Applications (SenseApp 2012)*, October 2012.