RASID: A Robust WLAN Device-free Passive Motion Detection System

Ahmed E. Kosba  
Dept. of Comp. and Sys. Eng.  
Faculty of Engineering  
Alexandria University, Egypt  
ahmed.kosba@alexu.edu.eg

Ahmed Saeed  
Dept. of Comp. Sc. and Eng.  
Egypt-Japan Univ. of Sc. and Tech. (E-JUST), Egypt  
ahmed.saeed@ejust.edu.eg

Moustafa Youssef  
Dept. of Comp. Sc. and Eng.  
Egypt-Japan Univ. of Sc. and Tech. (E-JUST), Egypt  
moustafa.youssef@ejust.edu.eg

ABSTRACT

Device-free passive (DfP) indoor localization is an emerging technology enabling the localization of entities that do not carry any devices nor participate actively in the localization process using the already installed wireless infrastructure. This technology is useful for a variety of applications, where special hardware might not be applicable or affordable such as intrusion detection, smart homes and border protection. In this paper, we present the design, implementation and evaluation of RASID, a DfP system for human motion detection. RASID combines different modules for statistical anomaly detection while adapting to changes in the environment to provide accurate, robust and low-overhead detection of human activities. Evaluation of the system in two different real testbed environments shows that it can achieve an accurate detection capability of 6% miss detection rate and 9% false alarm rate in both environments. In addition, the high accuracy and low overhead performance are robust to changes in the environment as compared to the current state of the art DfP detection systems. We also relay the lessons learned during building our system and discuss future research directions.

Keywords

Anomaly detection, device-free passive localization, motion detection systems, robust device-free localization.

1. INTRODUCTION

The increasing need for context-aware information and the rapid advancements in communication networks have motivated significant research effort in the area of location-based services. This effort resulted in the development of many location determination systems, including the GPS system [1], ultrasonic-based systems [2], infrared-based (IR) systems [3], and radio frequency-based (RF) [4] systems. Moreover, motion detection systems, that aim at detecting the motion of an entity carrying a device, were also developed [5–13]. These systems require the tracked entity to carry a device that participates in the localization process. Thus, we refer to them as device-based systems.

Motivated by the wide use of wireless LANs for indoor communication, we recently introduced the concept of device-free passive DfP localization [4] which enables the localization of entities that do not carry any devices nor participate in the localization process. This concept depends on the fact that the presence and motion of entities in an RF environment affects the RF signal strength, especially when dealing with the 2.4 GHz band which is used in different IEEE standards such as 802.11b and 802.11g (WiFi).

A typical DfP system (Figure 1) consists of signal receivers or monitoring points (MPs), such as standard laptops, and an application server which collects and processes information about the received signals from each MP. The application server uses the collected information to perform the detection/localization functions and initiates actions as needed.

Different DfP algorithms were proposed for detection [4,14] and tracking [14,16,17] of entities in indoor environments. The current tracking algorithms usually assume that the presence of the entity has already been detected. This paper focuses on the detection problem. The previously proposed techniques for DfP activity detection are either based on time-series analysis like the moving average and moving variance techniques proposed in [4,14] or based on classification using the maximum likelihood (ML) estimation [15]. These techniques provide good performance under strong assumptions, which limits their application domain. For example, current techniques are not robust to changes in the environment. That is they do not adapt to changes in the environment, e.g. humidity and temperature changes and furniture layout changes. Moreover, the parameters need to be changed as the deployment area changes. In addition, using ML classification requires the construction of a human motion profile which requires high overhead when dealing with large-scale environments, as it requires access to all areas of a building which might include restricted or private areas and requires several hours of calibration which makes the cost of this technique prohibitive. Finally, these techniques were...
either evaluated in **controlled** environments [14] or in **small-scale** real environments [15].

In this paper, we address the problem of designing a low-overhead, accurate and robust DfP motion detection system. We introduce the **RASID** system, a system that provides Robust WALN device-free passive motion Detection. **RASID** uses statistical anomaly detection techniques to detect motion inside indoor environments. It only constructs a profile, for the signal strength readings received at the MPs when there is no human activity during a short training phase leading to minimal deployment overhead. This also removes the requirement of access to restricted areas, since this silence period profile does not depend on the location of the humans. **RASID** also employs techniques for detecting dynamic changes in the environment and continuously updates its profile. Combined with its feature selection and a decision refinement module, **RASID** is not sensitive to changes in the deployment area. We evaluate the system in two different large-scale environments rich in multi-path and compare **RASID** to the state of the art DfP detection techniques. Our results show that **RASID** achieves its goals of high accuracy, with a 6% miss detection rate and 9% false alarm rate in both environments with minimal deployment overhead. We also compare **RASID**’s performance to three different DfP detection systems [14, 15]. The comparison shows that, in addition to its high accuracy and low overhead performance, **RASID** is robust to changes in the environment.

**RASID** is a software only solution on top of the already installed wireless networks and have many applications. These include intrusion detection, sensorless sensing, low cost surveillance, and smart buildings. Also, it can be used as a low-cost low-power first level security system that detects intrusions and then trigger more sophisticated security systems. This can be useful if the second level security system has limited power resources.

The rest of this paper is organized as follows: Section 2 presents the **RASID** system architecture and operation. Section 3 presents the experimental evaluation of the **RASID** system and comparison with other techniques. We discuss our experience with **RASID** and present some open research issues in Section 4. Section 5 reviews related work in the area of motion sensing and anomaly detection. Finally, Section 6 concludes the paper.

2. THE RASID SYSTEM

In this section, we give the details of the **RASID** system. We start by an overview of the system architecture followed by the details of the basic detection technique. We then discuss how the system handles temporal changes in the environment and updates its profile.

![Figure 1: RASID system architecture.](image)

After that, we describe the decision refinement module. We end the section by describing our region tracking user interface that provides a tool for the system user to visualize the detection results.

2.1 System Overview

The modules of the proposed system are implemented in the application server that collects samples from the monitoring points and processes them. Figure 1 gives an overview of the system architecture. The **RASID** system works in two phases:

1. A **short**, typically two minutes, **offline** phase: During the offline phase, the system studies the signal strength values when no human is present inside the area of interest to construct what we call a **“normal or silence profile”**. Note that this is a very low-overhead operation as the samples need not be collected as specific locations in the area of interest. So, the human participation in the calibration process is minimal. Previous techniques, e.g. [15], that use a ML classifier require building a motion fingerprint, that depends on the human positions, and therefore is a very high-overhead process.

2. A **monitoring** phase: In the monitoring phase, the system collects readings from the monitoring points and decides whether there is human activity (anomalous behavior) or not based on the information gathered in the offline phase. It also updates the stored normal profile so that it can adapt to environment changes. Finally, a decision refinement module is applied to further enhance the accuracy.

The **Normal Profile Construction Module** constructs the initial silence profile based on a short, typically two-
2.2 Mathematical Notations

Let $k$ be the number of streams, which is equal to the number of APs times the number of MPs. Let $s_{j,t}$ denote the signal strength reading for a stream $j$ that is received at a time instant $t$. The system studies the behavior of a sliding window $W_{j,t}$ of size $l$ that ends at time $t$, i.e., $W_{j,t} = [s_{j,t-l+1}, s_{j,t-l+2}, ..., s_{j,t}]$.

In order to study the behavior of the sliding windows, each sliding window $W_{j,t}$ is mapped to a single feature or value $x_{j,t}$ through a function $g$. For example, if the mean is the selected feature, then $g(W_{j,t}) = \frac{1}{l} \sum_{i=1}^{l} s_{j,t-l+i}$. Two types of features can be considered: measures of central tendency, such as the mean, and measures of dispersion or variation, such as the variance.

2.3 Normal Profile Construction

The purpose of this module is to construct a normal profile, capturing the RSS characteristics when there is no human in the area of interest. This is used later by other modules to detect anomalies, i.e. when a human moves in the area of interest.

This module is part of the offline phase. It extracts the feature values from the sliding windows over the collected data and estimates its distribution. The density function of the feature values observed is estimated using kernel density estimation. This is done for each stream independently. Figure 2 illustrates the operation.

Formally, for a stream $j$, given a set of $n$ sliding windows, each of length $l$ samples, each window $W_{j,i}$ is mapped to a value $x_{j,i}$, where $x_{j,i} = g(W_{j,i})$. Assume $f_j$ is the density function representing the distribution of the observed $x_{j,i}$’s. Then, given a random sample $x_{j,1}, x_{j,2}, ..., x_{j,n}$, the estimated density function $\hat{f}_j$ is given by [18]:

$$\hat{f}_j(x) = \frac{1}{nh_j} \sum_{i=1}^{n} V \left( \frac{x - x_{j,i}}{h_j} \right) \quad (1)$$

where $h_j$ is the smoothing parameter or the bandwidth and $V$ is a kernel function that satisfies:

$$V(u) \geq 0 \text{ and } \int_{-\infty}^{\infty} V(u) du = 1$$

The choice of the kernel function is not significant for the results of the approximation [19]. Hence, we choose the Epanechnikov kernel as it is bounded and efficient to integrate:

$$V(u) = \begin{cases} \frac{3}{4}(1-u^2), & \text{if } |u| \leq 1 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Also, we used Scott’s rule to estimate the optimal bandwidth [19]:

$$h_j^* = 2.345 \sigma_j n^{-0.2} \quad (3)$$

where $\sigma_j$ is an estimate for the standard deviation for the $x_{j,i}$’s.

After estimating the density function for the feature values extracted from the sliding windows, critical bounds minutes, training sample taken when there is no human motion present in the area of interest (Sections 2.3 and 2.4).

The Basic Detection Module examines each stream readings in the monitoring phase and decide whether there is an anomalous behavior or not. This operation is applied to each stream independently. It also assigns an anomaly score to each stream to express the intensity of the anomalous behavior (Section 2.5).

The Normal Profile Update Module updates the normal profiles constructed in the offline phase in order to adapt to changes in the environment (Section 2.6).

The Decision Refinement Module applies heuristic methods to refine the decision generated by the basic detection module to reduce the false alarm rates (Section 2.7).

The Region Tracking Interface provides an interface that visualizes the output of the above modules. This interface enables the user to identify detection events and provides the regions of the moving entities (Section 2.8).

We start by giving the mathematical notations followed by the details of the different modules.
are selected so that if the feature values observed in the monitoring state exceed those bounds, the observed values are considered anomalous. Given a significance parameter $\alpha$ and assuming $\hat{F}_j$ is the CDF of distribution shown in Equation 1 if the feature is a measure of central tendency, which can deviate to the left or the right, then lower and upper bounds will be calculated such that the lower bound is $\hat{F}_{j_{-\alpha/2}}$ and the upper bound is $\hat{F}_{j_{1-\alpha/2}}$. However, if the feature is a measure of dispersion, which can only deviate in the positive (or right) direction, then an upper bound is only needed and is equal to $\hat{F}_{j_{1-\alpha}}$.

In the next subsection, we study different features that can be selected and the one we choose.

2.4 Feature Selection

As the system requires an offline phase before operation, to learn the behavior of the signal readings in the normal state, the selected feature for system operation should be resistant to possible environmental changes that may affect the stored data, e.g. temporal variations. In addition, the selected feature should also be sensitive to the human motion so as to enhance the detection accuracy.

In this section, we compare two categories of features: central tendency measures and dispersion measures. The goal of this study is to identify the category that will be more promising for the system operation. For this study, we consider the mean as a central tendency measure, and the standard deviation as a measure of dispersion. We use the standard deviation, rather than the variance, as the variance is a squared measure, while the mean is not.

2.4.1 Sensitivity to Human Activity

The selected feature should be sensitive to human activity. To compare the two features, we use the Euclidean distance between the normalized histograms representing the silence and motion states. The Euclidean distance is defined as the square root of the sum of the squared distance between each corresponding histogram bin. The histograms are constructed over a two minutes period for each state using Testbed 1, which is discussed in Section 3. Figure 3 shows the comparison versus different window sizes. The figure shows that the distance between the histograms of the standard deviation is larger than the distance between the histograms of the mean. This indicates that the standard deviation feature is more discriminant of the human motion than the mean feature.

2.4.2 Resistivity to Temporal Variations

As the proposed system requires a learning phase before operation, it is necessary to reduce the temporal variation effect on the stored profiles. To compare the two features, we use two different silence data sets collected two weeks apart. Figure 4 shows the results. The more similar the histograms, the more resistive the feature is to the introduced variations. The figure shows that the standard deviation feature is less affected by temporal variations.

From this study, we conclude that the measures of dispersion, e.g. the standard deviation or variance, are more suitable for our proposed system. For the rest of the paper, we use the variance as the selected feature.

2.5 Basic Detection Procedures

The basic detection module runs during the monitoring phase. The purpose of this module is to detect signal strength anomalies, i.e. human presence, based on the normal profile constructed during the offline phase.
In particular, for a window of samples \( W_{j,t} \) for stream \( j \) at a given time instant \( t \), the module calculates the corresponding feature value \( x_{j,t} \), i.e., the sample variance. A stream \( j \) is considered anomalous if \( x_{j,t} \) is above the bound \( \hat{F}_{j_{-\alpha}} \) as discussed in Section 2.3.

The module also calculates an anomaly score, \( a_{j,t} \), that is used by the Decision Refinement Module (Section 3.3.3) to further enhance accuracy. For a given window, \( W_{j,t} \), the anomaly score, \( a_{j,t} \), can be calculated as:

\[
a_{j,t} = \frac{x_{j,t}}{\hat{F}_{j_{-\alpha}}}
\]  

(4)

where \( x_{j,t} \) is the sample variance of the window and \( \hat{F}_{j_{-\alpha}} \) is the critical value. Based on the chosen anomaly score calculation method above, we can see that a detected anomaly will have a score greater than one and a silence window will have a score of less than one.

In summary, the basic detection module requires two parameters: the window size \( l \) and the significance \( \alpha \) and uses the sample variance as a feature for anomaly detection. Finally, it calculates an anomaly score based on Equation (4).

2.6 Capturing Changes in the Environment: The Normal Profile Update Module

Due to the dynamic changes in the environments, such as changes in temperature and/or addition or removal of furniture, the stored normal profiles in the system may not capture the real normal state. This means that if the normal profile the system uses is fixed, this will lead to significant accuracy degradation. This highlights the need to update the normal profiles during the monitoring phase. This is the function of the Normal Profile Update Module.

The technique we employ for handling the update process is based on continuously updating the profile, i.e., the estimated kernel in Equation (1) by adding \( x_{j,t} \)'s, that do not have high anomaly scores in average to it. In particular, during the monitoring phase, the system groups the consecutive \( x_{j,t} \)'s in disjoint groups of size \( l_{\text{update}} \). The group that has an average anomaly score, calculated by Equation (4), less than one is added to the normal profile. The parameter \( l_{\text{update}} \) can be tuned to provide the desired performance. If \( l_{\text{update}} \) is small, this will lead to excessive sensitivity that implies a high false alarm rate. This is in addition to the overhead of updating the profiles continuously. On the other hand, if \( l_{\text{update}} \) is set to a large value, the system sensitivity will degrade and the miss detection rate will increase. We quantify the effect of the \( l_{\text{update}} \) parameter in more details in Section 3.3.2.

Adding new data to the normal profiles implies the need to give more weight to the recent data. Thus, instead of giving equal weights to the samples used for the probability calculation in the kernel equation (Equation 1), more weight will be given to the recent data. Therefore, Equation (1) is modified to:

\[
f_j(x) = \frac{1}{h_j} \sum_{i=1}^{n} w_i V \left( \frac{x - x_{j,i}}{h_j} \right)
\]  

(5)

where \( \sum_{i=1}^{n} w_i = 1 \). We choose linear weights such that \( w_i = \frac{n}{i(n+1)/2} \). We found that exponential weights do not provide good performance due to the high discrimination introduced between older and newer data.

2.7 The Decision Refinement Module

Wireless environments are usually noisy and continuously changing. This fact leads to the observation that there can be many false alarms if the system generates alarms just because a single stream is anomalous. As false alarms can lead to unnecessary actions, a mechanism is needed to reduce those false alarms so that an alarm is only generated when it is highly likely that someone is inside the area of interest. The decision refinement module reduces the high false alarm rate by fusing the states of all streams. Two methods can be considered for that fusion: the number of concurrent alarms and the sum of the anomaly scores.

2.7.1 Number of Concurrent Alarms

In this method, an alarm is generated in case there are at least \( m \) concurrent alarms from \( m \) streams (i.e. at least \( m \) streams have anomaly scores > 1). A similar method was already proposed and used with a small time buffer in [14]. However, in Section 3.3.3, we show that this approach can lead to ignoring significant alarms, i.e., significantly increasing the false negative rate. This is due to the fact that many parts in the area of interest may be covered by only a few number of streams, especially with a low density of APs and MPs. To avoid this case, we need to select a method that considers both the state of all streams and the anomaly score of each of them, which leads to the following method.

2.7.2 Sum of Anomaly Scores

Since the Basic Detection Module assigns an anomaly score to each detected event that expresses its significance, this can be leveraged to enhance the detection performance. In the sum of anomaly scores method, a global anomaly score \( a_t \) is calculated by summing the individual anomaly scores for each stream:

\[
a_t = \sum_{j=1}^{k} a_{j,t}
\]  

(6)

If a noticeable change in \( a_t \) occurs while at least one stream is anomalous, this implies the start of an anomalous behavior. The module uses exponential smoothing
to monitor the \( a_i \) in order to avoid the noisy samples, hence reduces the false alarm rate (Figure 11).

Both of the above methods will be compared quantitatively in Section 3.3.3.

2.8 Region Tracking User Interface Module

The system provides an interface that provides information about the probable regions of the detected event. This is based on visualizing the anomaly degree of each stream enabling the user to identify the regions that probably have moving entities inside. This is done by coloring each pixel on the map according to its distance from each stream endpoints and according to the anomaly score of each stream. Figure 5 displays the output of this interface when two persons are moving inside a typical site, showing the true locations of the two persons.

3. EXPERIMENTAL EVALUATION

In this section, we present the experiments we conducted to evaluate the RASID system. First, we describe the testbeds of the experiments and the metrics used for the evaluation. Then, we present the results of the evaluation and provide an analysis of the main system parameters. Finally, we compare the performance of the RASID system with the other techniques proposed for WLAN device-free passive detection [14, 15].

3.1 Experimental Testbeds and Data Collection

We collected two sets of data to evaluate the system performance, each in a different testbed. The first testbed was collected in an office of approximately 2000 ft\(^2\). The second experiment was conducted in a two-floor home building where each floor was approximately 1500 ft\(^2\). Both testbeds were covered with typical furniture. For both testbeds, we used four Cisco Aironet 1130AG series access points and three DELL laptops equipped with D-Link AirPlus G+ DWL-650+ Wireless NICs. The experiments were conducted in typical IEEE 802.11b environments. Figures 6 and 7 show the layouts of both experiments.

For the data collection, sets of normal (silence) state readings and continuous motion readings were collected for each testbed. A total of about one hour and 15 minutes of data was collected for each testbed. For Testbed 1 this includes three motion sets, while for Testbed 2, this includes two motion sets. A motion set covers the entire area of the testbed, as shown in figures 6 and 7 and represents the movement of a single person moving around the site continuously without any stops.

For system evaluation, extreme conditions were employed. The training period is chosen to be the first two minutes of the entire data collected with the absence of human motion. In addition, only one person moved in the area of interest.\(^1\)

\(^1\)Note that more persons moving in the area of interest leads to an easier detection problem due to the higher changes in the environment.
3.2 Evaluation Metrics

We used two metrics to analyze the detection performance: the false positive (FP) rate and the false negative (FN) rate. The false positive rate refers to the probability that the system generates an alarm while there is no human motion in the area of interest. Whereas the false negative rate refers to the probability that the system fails to detect the human motion in any place inside the area. We also use the F-measure, which provides a single value to measure the effectiveness of the detection system [20].

3.3 System Performance

In this section, we present the results of the system evaluation and show how each module contributes to these results. We also analyze the effect of the different system parameters on the performance of the system. Table 1 summarizes the system performance for both testbeds using the same parameters. The table also shows the enhancement introduced by each module. It should be noted that the best attainable system performance (i.e. given the optimal parameters) is provided in Section 3.4 when we compare the performance of the system to other techniques.

3.3.1 Basic Detection Module

As mentioned earlier, the basic detection module requires the selection of the sliding window size $l$ and the significance $\alpha$. Figure 8 illustrates the effect of these parameters applied to Testbed 1. Similar performance has been observed for Testbed 2. Table 1 summarizes the results for both experiments. The figure shows that choosing a very short window size will make the system less sensitive to human motion. On the other hand, choosing a very large window size will introduce detection latency and a high FP rate.

For the significance parameter, as $\alpha$ decreases, the FP rate decreases and the FN rate slightly increases. This means that increasing the significance will result in less system sensitivity. Therefore, to balance the different performance metrics, we choose $l = 10$ and $\alpha = 0.01$.

Table also shows that Testbed 2 has a much higher FN rate than Testbed 1. This is due to the lower device density in Experiment 2 (same number of devices are used to cover a much larger area). However, using the basic detection technique, we can say that approximately 94% of the motion is detected in Testbed 1 and 76% of the motion is detected in Testbed 2. It can be noted also that the FP rates in both experiments are high because of the environment changes. The two-minute training period is not enough to sustain accurate detection for one hour of operation. This highlights the need for the Normal Profile Update Module that can reduce this high FP rate as we show in the next subsection.

3.3.2 Normal Profile Update Module

The normal profile update module requires the selection of the update window size $l_{\text{update}}$. Choosing a small $l_{\text{update}}$ will make the system very sensitive to noisy readings causing a high FP rate. On the other hand, a large $l_{\text{update}}$ will make the system less sensitive to human motion causing a higher FN rate. Figure 9 illustrates these effects of the update window size on the system performance for Testbed 1 when $l = 10$ and
An update window size between 6 and 10 is sufficient to reduce the high FP rate. Figure 9 also shows that although update window sizes that are more than 10 achieve better F-measure we did not find that to hold for other values of the parameters $l$ and $\alpha$. Therefore, we should be conservative about increasing the update window size. Thus, we choose $l_{update} = 8$. The results are shown in Table 1. The table shows that there is at least 40% reduction in the FP rate in both testbeds. However, this resulted in a slight growth in the FN rate. Overall, the F-measure was enhanced by 3 to 4%.

### 3.3.3 Decision Refinement Module

The normal profile update module reduced the high FP rate by updating the stored profiles. However, the FP rates still need to be addressed. The decision refinement module tries to limit the FP rate by generating alarms only when it is highly likely that there is someone inside the area of interest.

Figure 10 shows the result of the concurrent alarms method for Testbed 1. While the increase of the number of concurrent alarms reduces the high FP rate, it resulted in a severe degradation in the system sensitivity to human motion, i.e. FN rate. This is mainly due to the fact that there are some areas that are covered by a single stream. Therefore, the case when a person walks through these regions cannot be detected when the number of the required concurrent alarms is high. Using a voting mechanism between streams while
giving them equal weights cannot be used efficiently to enhance the accuracy. On the other hand, Figure 11 displays the sum of anomaly scores curve for the data collected for Testbed 1. The figure shows that the motion periods are clearly distinguishable from the silence state.

In order to reduce the FP rate, we smooth the curve in Figure 11 using exponential smoothing. A large increment in the value of the smoothed curve implies the start of a period of human motion. This technique makes use of the locality of human motion, meaning that the human will continue to affect the same stream and/or other streams near it, causing the sum of the anomaly scores curve to have higher values.

This method requires the selection of a suitable smoothing coefficient and a threshold for the increment in the smoothed curve. Choosing a smoothing coefficient of 0.05 and a threshold of 35% of the normal level leads to a significant improvement for both testbeds. Deviations from these parameters will not lead to significant degradation in the results. Table 1 illustrates the results and shows that the decision refinement module can lead to up to 10% enhancement in the F-measure for both testbeds.

It is important to note that this module also reduces the FN rate. This can be explained by noting that some of the previously non-detected events are now detected because this technique makes use of the history of the state of the activity inside the area of interest.

3.4 Comparison with Previous Techniques

In this section, we compare the performance of RASID to the previous techniques devised for WLAN device-free passive detection. We start by a brief description of the techniques, followed by the different aspects we evaluate the techniques on. Finally, we present the results of the comparison.

3.4.1 Comparison Techniques

Three techniques are considered for the comparison:

1. The moving average technique [14]: The moving average technique uses a central tendency feature, i.e. the average. It uses two sliding window averages: a short window average representing the current system condition and a long window average representing history. The idea is to compare the two averages and if the difference is above a threshold, a detection is announced. It is important to note that the moving average technique does not require a training phase.

2. The moving variance technique [14]: The moving variance technique uses a dispersion feature, i.e. the variance. Similar to the moving average technique, it compares the variance of the current system state, based on a sliding window, to the variance of the silence period, obtained through a training phase. If the difference is above a threshold, a detection is announced.

3. The maximum likelihood (ML) classification technique [15]: The ML technique constructs profiles for the silence period as well as for the motions period for different locations in the area of interest. The profiles represent the signal strength distribution for each stream at each location. Therefore, it involves significant training data. During the detection phase, the system finds the profile that has the maximum likelihood given a signal strength vector, one entry for each stream. If the
estimated profile corresponds to a motion profile, an event is detected.

3.4.2 Comparison Aspects

The comparison considers three aspects:

- Static accuracy: in terms of the F-measure when the system is evaluated with the same profiles it was trained on.
- Profiles’ robustness: that is how consistent the performance of the system is when the tested profiles are different from the trained ones, for example due to temporal changes in the environment.
- Overhead: That is the effort needed to deploy the system.

3.4.3 Comparison Results

Table 2 shows the results of the four techniques in two cases. The first case is when the optimal parameters of each algorithm are adjusted for each experiment. In addition, the systems are tested with the same profiles (i.e. data sets) that were used to train them. This is to test the best attainable accuracy by tuning the parameters to obtain the maximum possible F-measure. The second case is when the testing data set is collected two weeks after the data set used for training. This is to test the robustness of each technique to changes in the environment.

The results show that:

1. In terms of static accuracy: The results show that, in terms of the F-measure, the RASID system outperforms all other systems in Testbed 1 and is slightly lower than the maximum likelihood classifier in Testbed 2. Compared to the moving average and the moving variance techniques, the RASID system provides high accuracy due to the techniques it uses to adapt to the environment changes and the decision refinement procedure. On the other hand, the maximum likelihood technique achieves slightly higher accuracy in Testbed 2 as it stores a motion profile, which requires a significantly higher overhead than the RASID system.

2. In terms of profiles’ robustness: the moving average technique does not store any profiles, therefore, its overall performance is low but almost the same as the profiles change. On the other hand, the robustness of the ML technique is the least because it uses the mean signal strength values as the features used for classification. Therefore, after two weeks, it is not guaranteed that the distribution of the testing signal strength values follows the learned one. This is why the FP rate for the ML technique is too high due to the shift that occurred in the signal strength distributions. It can also be observed that RASID’s performance in the two cases was the best because of two main reasons. First, RASID uses a dispersion feature (variance) for its operation, and we showed in Section 2.4 that measures of dispersion are more suitable for the system operation. The second is due to the techniques RASID employs for adapting to changes in the environment, and for enhancing the performance. This is why RASID’s performance is better than the moving variance in general, although the moving variance uses the same feature as RASID.

3. In terms of overhead: the moving average technique has the minimum overhead as it does not need any learning phase. The moving variance and RASID deployment need to construct the normal/silence profile by collecting samples for two minutes when the human is not present. On the other hand, the maximum likelihood classifier has the worst overhead as it needs to construct a motion profile at each location in the area of interest in addition to the normal profile. This limits its applicability to large-scale environments.

In summary, although RASID’s static detection accuracy is as accurate as the maximum likelihood technique (and in Testbed 1 better), the maximum likelihood technique has significantly higher overhead than RASID because of its motion profile requirements. In addition, RASID is robust to changes in the training profiles and significantly outperforms the ML technique, by up to 40% enhancement, when the environment conditions change.

Compared to the moving average and moving variance techniques, the RASID system is significantly more accurate, more robust, and has a minimal deployment overhead.

4. DISCUSSION

In this section, we discuss some points related to the configuration and the performance of the RASID system. We also highlight some research issues and some challenges that can be addressed in future work.

4.1 Univariate VS Multivariate Density Estimation

As mentioned before, the basic detection module studies each stream independently by estimating the univariate density for the selected feature of the sliding windows extracted from the training data. Another possibility was to construct a multivariate density estimate for the data of all streams. This implies a modification to the anomaly detection criteria. Different
algorithms can be applied in this case, e.g. [21]. Our experience with this algorithm shows that this leads to a degradation of the system accuracy. The main reason for this degradation is that the system sensitivity is significantly reduced, especially when the number of streams is large. In that case, the system may not be able to detect an anomaly in one stream only, as its effect may not be much sensed.

### 4.2 Detection and Identification of Independent Events

The above experiments showed that the system is capable of detecting a single person moving inside the area of interest. Obviously, the detection performance will be enhanced if there were more than one entity in the area of interest. We verified that the system will be able to declare that there is anomalous behavior inside the area more clearly in this case.

It would be useful to identify the number of moving entities in some applications. Figure 5 shows that we can detect that there are multiple entities in the area of interest. However, as our system uses limited data to satisfy the feasibility design goal (normal profiles only), the system cannot provide full information about the number of entities in all cases. For example, if two entities are affecting a single stream only, the system will detect them as one entity. This is because there is no enough information that enables the system to differentiate between the two cases. On the other hand, in some cases, the system can tell with high probability that some events are due to independent entities. Here, we briefly describe the constraints through which the system can provide information about the number of independent entities.

First, let $T_{\min_{ij}}$, a $k \times k$ square matrix denote what we call a minimum time reachability matrix. Each entry in this matrix stores the minimum time needed for an entity to affect two streams $i$ and $j$, such that

$$T_{\min_{ij}} = \frac{D_{\min_{ij}}}{v_{\max}}$$

, where $D_{\min_{ij}}$ represents the minimum distance between the nearest two points on the $i$ and $j$ streams lines of sight and $v_{\max}$ represents the maximum movement velocity inside the area. The distance $D_{\min_{ij}}$ can be calculated from the site map, and $v_{\max}$ can be estimated based on empirical observations.

Two events $E_1$ and $E_2$ are considered independent (i.e. not generated by the same entity), if they satisfy the following conditions. First, they should be affecting two different streams $i$ and $j$ and second, the time difference between $E_1$ and $E_2$ is less than the value $T_{\min_{ij}}$. The time difference between the two events are calculated based on the time difference between the times when the anomaly scores for the two events reach the peaks as they express the moments when the entities are affecting the streams the most. To tell that $n$ events are independent, each pair of those events should satisfy the conditions described above. The above conditions imply that the system cannot detect more than $k$ moving entities, where $k$ is the number of streams as stated earlier.

To conclude, despite the limited information the system uses, the system can provide information about the number of independent events inside the monitored area given some conditions. The significance of this point can be clear when applied inside large scale environments.

Another possibility is to use the level of the change in variance as an indication of the number of entities. The hypothesis is that the more human affecting a sin-

<table>
<thead>
<tr>
<th>Results with static profiles</th>
<th>Moving Average</th>
<th>Moving Variance</th>
<th>ML.</th>
<th>RASID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testbed 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FN Rate</td>
<td>0.1446</td>
<td>0.1426</td>
<td>0.0363</td>
<td>0.0222</td>
</tr>
<tr>
<td>FP Rate</td>
<td>0.1385</td>
<td>0.104</td>
<td>0.1547</td>
<td>0.0896</td>
</tr>
<tr>
<td>F-measure</td>
<td>0.858</td>
<td>0.8743</td>
<td>0.9099</td>
<td>0.9423</td>
</tr>
<tr>
<td>Testbed 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FN Rate</td>
<td>0.0759</td>
<td>0.398</td>
<td>0.0372</td>
<td>0.054</td>
</tr>
<tr>
<td>FP Rate</td>
<td>0.7412</td>
<td>0.1478</td>
<td>0.0774</td>
<td>0.0713</td>
</tr>
<tr>
<td>F-measure</td>
<td>0.6935</td>
<td>0.7522</td>
<td>0.9438</td>
<td>0.9368</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Results with testing profiles separated two weeks from the training profiles.</th>
<th>Moving Average</th>
<th>Moving Variance</th>
<th>ML.</th>
<th>RASID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Testbed 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FN Rate</td>
<td>0.2165</td>
<td>0.319</td>
<td>0.1563</td>
<td>0.1058</td>
</tr>
<tr>
<td>FP Rate</td>
<td>0.0711</td>
<td>0.1561</td>
<td>0.952</td>
<td>0.0782</td>
</tr>
<tr>
<td>F-measure</td>
<td>0.8449</td>
<td>0.7414</td>
<td>0.5991</td>
<td>0.9385</td>
</tr>
<tr>
<td>Testbed 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FN Rate</td>
<td>0.2461</td>
<td>0.4152</td>
<td>0.1293</td>
<td>0.0931</td>
</tr>
<tr>
<td>FP Rate</td>
<td>0.3602</td>
<td>0.0513</td>
<td>0.831</td>
<td>0.0722</td>
</tr>
<tr>
<td>F-measure</td>
<td>0.7022</td>
<td>0.7149</td>
<td>0.6491</td>
<td>0.9165</td>
</tr>
</tbody>
</table>

Table 2: Performance comparison with previous DfP detection techniques. Highlighted entries are the techniques that give the best F-measure.
ggle stream, the higher the variance should be. This hypothesis still needs to be verified though.

4.3 Integration with DfP Tracking Systems

Our system can provide useful information to DfP tracking systems like the ones proposed in [16,17]. First, a DfP tracking system can use our system to decide whether to start the tracking process or not. Also, the system can enhance the tracking accuracy by limiting the probable locations to a certain area (e.g., as in Figure 5). In addition, given the conditions described earlier, our system can help the tracking system identify the number of intruders and the area of each one, so that it can apply the tracking algorithms to each area independently. This will need further investigation and experimentation.

4.4 Combining Features

Although we showed in this paper that using the variance as a feature is better than using the mean, both features can be used concurrently to achieve better performance. Our initial results show that combining both features and using a simple voting scheme can enhance the results in some cases. This is a subject for future investigations.

4.5 Signal Strength Readings Synchronization

The synchronization of the signal strength readings received at the monitoring point can be necessary in some cases. For example, the technique described before for checking the independence of the detected events requires synchronization of the readings across the streams. In addition, the decision refinement module requires the different streams to be synchronized. In this paper, we took a centralized approach for synchronization, where the application server requests the MPs to initiate a reading. Other approaches, such as time synchronization of the MPs can be employed. The advantages and disadvantages of each technique in terms of accuracy and overhead can also be investigated.

4.6 Effect of Different Hardware

The hardware used to capture the signal strength values can affect system performance. Through our experiments, we studied how the WLAN NIC type affects the quality of the collected readings. We found that NICs differ in two main aspects: sensitivity to human activity and noise readings. For example, some cards cannot sense the human shadowing effect unless it is sustained for a sufficient period of time. The readings of some other cards are noisy and require extensive filtering. These experiments considered the NICs only. However this can hold from the APs perspective too. Therefore, we believe a study is needed to identify which hardware will be more suitable for the system operation and how to account for these variation between cards and allow the system to operate with different cards.

5. RELATED WORK

Motion Detection in device-based systems has been an active field of research. Several works have been proposed to detect the motion of an entity carrying a device either with the use of special hardware like accelerometers or motion sensors [5,8], or by using the existing network infrastructures like wireless networks [9,11] and GSM [12,13].

From the device-free perspective, multiple technologies can be used to provide the desired capabilities including: ultra-wide band radar [22], computer vision [23], physical contact based systems [24] and radio tomographic imaging [25]. Other technologies include the usage of wireless sensors for tracking transceiver-free objects [20] as well as the usage of RFID tags [27]. Those technologies share the requirement of installing special hardware to handle the device-free different functionalities. In addition, cameras and IR sensors are limited to line-of-sight vision and thus they require a high cost deployment to cover all site regions. Moreover, regular cameras can fail to work in the dark or in the presence of smoke, and they can cause privacy concerns. Ultra-wide band radar based techniques also suffer from high complexity. Moreover, some techniques can require high density to provide full coverage like radio tomographic imaging and physical contact based systems using pressure sensors.

WLAN device-free passive systems try to avoid the above drawbacks by using the already available wireless infrastructure. The concept of device-free passive detection and tracking using WLANs was first proposed in [14]. Techniques for DfP detection [14,15] and tracking [14,16,17] were introduced. The proposed techniques for the detection capability are either based on time-series analysis like the moving average and moving variance techniques proposed in [14] or based on classification using the maximum likelihood estimation [15].

In comparison, RASID uses anomaly detection techniques to identify the deviations from the normal (silence) state. RASID system uses a semi-supervised statistical technique that models the learned normal behavior using a kernel-function based non-parametric estimation. The kernel-function based anomaly detection has been used in several applications where the distribution of the normal behavior is not known. For example, non-parametric estimation using Gaussian kernels was used in network intrusion detection [21] and novelty detection applied to oil flow data [28]. Also, density estimation using Epanechnikov kernels was used in online outlier detection in sensor data [29] and to achieve continuous adaptive outlier detection on distributed data streams [30].
Compared to the previously proposed WLAN DfP detection techniques, the usage of the statistical anomaly detection technique, along with the other techniques devised for adapting to environment changes and refining the decision, enable RASID to achieve low deployment overhead, high accuracy and high robustness.

6. CONCLUSIONS

In this paper, we presented the RASID system, a system that enables device-free passive motion detection using already installed wireless networks. RASID uses statistical anomaly detection techniques to provide the detection capability. It constructs a profile for the normal state through a short training phase, then applies different procedures to check for human activity. The RASID system also employs profile update techniques to capture changes in the environment and to enhance the detection accuracy.

RASID was evaluated in two different real environments. Given the optimal parameters configurations, the system provided an accurate detection capability reaching 6% false negative rate and 9% false positive rate in both environments. The system performance was also analyzed when the same parameters were used for both environments achieving high accuracy in both environments. To show its significance, the performance of the RASID system was also compared to the previously introduced techniques for WLAN DfP detection in terms of overhead, accuracy, profiles' robustness. The results showed that RASID provides a low cost, high accuracy, and high robustness solution for WLAN DfP motion detection achieving its design goals.

7. REFERENCES


