Recognizing whether sensors are on the same body

Cory Cornelius and David Kotz

Department of Computer Science, Dartmouth College, Hanover, NH, USA
Institute for Security, Technology, and Society, Dartmouth College, Hanover, NH, USA

Abstract. As personal health sensors become ubiquitous, we also expect them to become interoperable. That is, instead of closed, end-to-end personal health sensing systems, we envision standardized sensors wirelessly communicating their data to a device many people already carry today, the cellphone. In an open personal health sensing system, users will be able to seamlessly pair off-the-shelf sensors with their cellphone and expect the system to *just work*. However, this ubiquity of sensors creates the potential for users to accidentally wear sensors that are not necessarily paired with their own cellphone. A husband, for example, might mistakenly wear a heart-rate sensor that is actually paired with his wife's cellphone. As long as the heart-rate sensor is within communication range, the wife's cellphone will be receiving heart-rate data about her husband, data that is incorrectly entered into her own health record.
We provide a method to probabilistically detect this situation. Because accelerometers are relatively cheap and require little power, we imagine that the cellphone and each sensor will have a companion accelerometer embedded with the sensor itself. We extract standard features from these companion accelerometers, and use a pair-wise statistic – coherence, a measurement of how well two signals are related in the frequency domain – to determine how well features correlate for different locations on the body. We then use these feature coherences to train a classifier to recognize whether a pair of sensors – or a sensor and a cellphone – are on the same body. We evaluate our method over a dataset of several individuals walking around with sensors in various positions on their body and experimentally show that our method is capable of achieving an accuracies over 80%.

1 Introduction

Mobile sensing of the human body is becoming increasingly pervasive with the advent of personal devices capable of processing and storing large of amounts of data. Commercial devices like the FitBit [7] and BodyBugg [1] allow a person to collect nearly continuous data about his or her health. The FitBit, for example, allows a person to track one's own fitness and sleeping patterns by wearing an accelerometer on the waist.

Typically these devices are highly specialized, end-to-end solutions, but we imagine the sensors in these products becoming commodities and inter-operating with a device most people carry with them everyday: cellphones. A person could wear several sensors of varying types (e.g., blood pressure monitor, pulse oximeter, pedometer, blood glucose meter). Because of the physiological requirements, or comfort, these sensors will necessarily be attached at different locations on the body. We imagine these sensors wirelessly communicating with a person's cellphone, which would store and aggregate

all data coming from the sensors. In fact, this scenario is feasible today, and there are purchasable medical and fitness sensors capable of communicating to cellphones via Bluetooth.

There are many security issues, not to mention privacy issues, with this scheme. How does the cellphone authenticate valid sensors? How do sensors discover the presence of the cellphone, without exposing their own presence? How does the user pair sensors with the cellphone? What types of encryption are employed to maintain confidentiality and integrity? How do we balance privacy and usability? We focus our attention on one specific challenge: how can we verify that a suite of sensors are attached to the same person?

Suppose Alice and Fred, a health-conscious couple living together, each decide to buy a fitness-monitoring sensor. The instructions indicate that each should "pair" their respective sensor with their own cellphone. *Pairing* ensures, through cryptographic means, that a sensor is only able to communicate with a specific cellphone. One day, when Alice and Fred go for a run, Alice unknowingly wears Fred's sensor, and Fred wears Alice's sensor. As they run, thereby remaining in communication range, Fred's cellphone will be collecting data about Alice, but labeling the data as Fred's and placing it in Fred's health record, and vice versa. This problem, a result of the one-to-one pairing model, is even more likely as the number of sensors grows. The implicit assumption when pairing is that the sensors paired with a cellphone will not be used by anyone else but the user of the cellphone.

Our goal is to make life easier for people like Alice and Fred. Although Alice and Fred buy identical sensor devices, Alice should be able to strap on either device and have her cellphone recognize which device is attached *to her*, automatically creating the phone-device association without an explicit pairing step. Similarly, if Alice and Fred jointly own another sensor device, either may use the sensor at any time, and again the correct cellphone should detect which body is wearing the sensor and receive the data into the correct person's health record.

To achieve this vision requires two core problems to be solved. First, Alice's phone must be able to determine which sensors are attached to Alice's body, ignoring sensors that may be in radio range but not attached to Alice. Second, the phone and sensor devices must be able to agree on a shared encryption key, to secure their communications; ideally, this should require no user assistance and be more secure than in most "pairing" methods today. In this paper we specifically address the first challenge, leaving the second challenge to future work. There are existing solutions that address the second challenge, but it is unclear if those solutions can be applied for accelerometers that are not intentionally shaken together [13].

To address the first challenge, the sensor device must somehow attest (to the cellphone) which body is wearing the sensor at the current time. Ideally, the phone would analyze the data coming from the sensors to see whether it identifies the wearer by some biometric measure. However, not all types of sensors, or sensor locations, produce data that is suitable for biometric identity verification. Thus we propose the following compromise: every sensor device will include an accelerometer sensor in addition to its primary sensor (ECG, blood pressure, etc.). Accelerometers are cheap, so this is a relatively inexpensive addition; instead of biometric identity verification with a wide

variety of sensor data, sensor placement, and usage conditions, we only need to find correlations for the accelerometer data that answers the question: are the devices in a given set all attached to the same body?

We recently [6] formalized this problem as the "one-body authentication problem," which asks: how can one ensure that the wireless sensors in a wireless body area network are collecting data about one individual and not several individuals? We identified two variants of this problem. The *strong* version of this problem requires identifying which person the sensors are attached to, whereas the *weak* version of this problem simply requires determining whether the sensors are on the same body. We noted how existing solutions do not necessarily solve the problem and called for further research. Thus, we now aim to provide a solution to the weak one-body authentication problem; given such as solution, one might solve the strong-body problem for one of the sensors in a set, and be able to extrapolate the verification to all of the sensors on the body.

Our paper is organized as follows. In the next section we describe our model. In the third section we briefly describe our approach and hypothesis as to why we believe our approach will work. In the fourth section we describe, in detail, our method. In the fifth section we describe the data we collected as well as our collection method. In the sixth section we evaluate our method. In the final sections, we discuss related work and distinguish our work from earlier approaches, and provide some discussion about our method's limitations and about some potential future work.

2 Model

We imagine a world where personal health sensors are ubiquitous and wirelessly connect to a user's cellphone. Thus, there are two principle components in our system:

- One **mobile node** (e.g., the user's cellphone) per user.
- Many **sensor nodes** (e.g., blood glucose, pedometer, electrocardiography).

We assume that mobile nodes communicate wirelessly with sensor nodes. Sensor nodes are also capable of communicating wirelessly with mobile nodes but have limited computational resources relative to the mobile nodes. Additionally, sensor nodes have the ability to detect when they are attached to a user (although they will not know to whom). The sensor node might contain a circuit that is completed, for example, when the user straps a sensor node onto their body and the two ends of a necklace or wrist-strap come into contact. Finally, we also assume each sensor node, and the mobile node, has an accompanying triaxial accelerometer of the same type (so that their readings may be directly compared). Since accelerometers are tiny, cheap, and require little energy to operate, this is a reasonable assumption.[1]

2.1 Binding

"Binding" occurs when a user wishes to use a sensor node. The following happens:

[1] The Freescale MMA845xQ line of accelerometers, for example, cost $0.95 (in quantities of 100K) and consume "1.8 microamps in standby mode and as low as 6 microamps in active mode" [8].

1. The user straps the sensor node to their body, thereby turning it on.
2. The sensor node detects that it was applied, and broadcasts its presence.
3. The mobile node receives the broadcast, thereby binding it with the sensor node, and labels that sensor node as unauthenticated.

Binding is like pairing, but without the need for user intervention. In a pairing scenario, the user is usually required to enter a shared key on one of the devices. Binding does not have this requirement. When a sensor node is bound to a mobile node, the sensor node enters an unauthenticated state.

2.2 Authentication

"Authentication" is a process, initiated by the mobile node, for verifying which of the mobile node's bound sensor nodes are on the same body. Once a sensor node is authenticated, the mobile node will record sensor data from that node; until then, the data will be ignored. (As it may take some time for authentication to succeed, in some implementations the mobile node may buffer the incoming data received between the moment of binding and the moment of authentication, recording the data only once authentication is assured. This "retroactive authentication" of the early data is feasible because of our assumption that a sensor node can detect its own attachment and removal; if a sensor node is moved from one body to another before it was authenticated on the first body, the unbinding and rebinding events will clear the buffer on the first body's mobile node.)

To achieve authentication, our protocol requires an algorithm that is able to decide whether two streams of data are originating from sensor nodes on the same body. That is, given a stream of accelerometer data from a sensor node, the algorithm examines the correlation between a sensor node's data stream and the mobile node's data stream, with the requirement that the two streams should correlate well only when both the mobile node and the sensor node are on the same body. The algorithm should return true if and only if the two data streams are well correlated and false otherwise. We present the details of our algorithm in Section 4.

Procedure 1 provides an overview of the process for the mobile node to authenticate sensor nodes. Because our method depends on recognizable acceleration events, our algorithm performs authentication only when the user is walking. The mobile node records acceleration data using its internal accelerometer for t seconds. Simultaneously, it asks the other sensor node to send it acceleration data for the same duration. The duration required depends on the level of confidence desired; a shorter duration may lead to more incorrect results (false positives and false negatives), but a longer duration makes the approach less responsive after the person first puts on the sensor. It then runs our algorithm, called AreCorrelated, to determine whether its internal acceleration data correlates with the sensor node's acceleration data. Only when the accelerometer data correlates well does the mobile node begin to record that sensor node's other sensor data (e.g., electrocardiography data).

2.3 Unbinding

Unbinding occurs when a user removes a sensor node. In the ideal case, the following happens:

Procedure 1 Authenticating sensor nodes, from the mobile node's perspective

Notation:
B: set of bound sensor nodes, initially empty
A_i: acceleration data from sensor node i, where $i = 0$ is the mobile node's acceleration data.
$\mathsf{Record}(t)$: read mobile node's accelerometer for t seconds
$\mathsf{Recv}(b, t)$: read sensor node b's accelerometer for t seconds
$\mathsf{AreCorrelated}(x, y)$: determine whether acceleration data x and y

---

while { true } **do**
  if $b :=$ NewSensorNodeDetected() **then**
    $B := B \cup b$
    { Mark sensor node b as unauthenticated }
  end if
  for $b \in B$ **do**
    if $\mathsf{Disconnected}(b)$ **or** $\mathsf{Timeout}(b)$ **then**
      $B := B \setminus b$
    else if $d :=$ $\mathsf{RecvData}(b)$ **and** $\mathsf{IsAuthenticated}(b)$ **then**
      $\mathsf{RecordData}(b, d)$ { Save b's data d in our health record }
    end if
  end for
  if UserIsWalking() **then**
    for $b \mid b \in B$ **and not** $\mathsf{IsAuthenticated}(b)$ **do**
      { The next two lines are accomplished in parallel }
      $A_0 := \mathsf{Record}(t)$
      $A_b := \mathsf{Recv}(b, t)$
      if $\mathsf{AreCorrelated}(A_0, A_b) =$ true **then**
        { Mark sensor node b as authenticated }
        { Tell sensor node b to send sensor data }
      end if
    end for
  end if
end while

1. The user unstraps the sensor node from their body.
2. The sensor node detects that it was removed and notifies the bound mobile node of this fact.
3. The mobile node acknowledges this notification, thereby unbinding it with the sensor node.
4. Upon receipt of this acknowledgement (or upon timeout), the sensor node turns off.

A sensor node may lose power or go out of range of the mobile node, during this process or prior to the user unstrapping the sensor node. Thus, the mobile node periodically pings each sensor node (not shown in Procedure 1); if the sensor node does not reply (after some timeout period), the sensor node is likely not on the same body, and the mobile node treats it as unauthenticated and unbound.

3 Approach

Our goal is to determine whether a sensor node is on the same body as a mobile node receiving the sensor node's data. That is, we provide a solution for the weak one-body authentication problem. Our solution could be used as the first step in a strong one-body authentication solution by first verifying that all the sensors are on the same body, then using some subset of the sensors to provide strong one-body authentication (i.e., via some biometric one of the sensors could determine) to all the sensors on the body. To maximize the generality of our solution, we require each sensor to have an accompanying accelerometer.

Our intuition is that if sensors are on the same body, then (at a coarse level) all of the sensors' accelerometers experience similar accelerations. If a user is seated, or lying down, then there is not much information we can extract from the accelerometer data to make the determination that a suite of sensors are on the same body. There are a variety of activities that cause bodily acceleration, but we focus on walking. When walking, a human body is largely rigid in the vertical direction. Although our limbs do bend, we hypothesize that the vertical acceleration (i.e., the acceleration relative to gravity) experienced by sensors placed anywhere on a walking body should correlate well. As one foot falls, that side of the body experiences a downward acceleration due to gravity, followed by an abrupt deceleration when the foot contacts the ground. Sensors on one side of the body should experience a similar vertical acceleration, while sensors on the other side of the body will experience the opposite. We should expect positive correlation for one side of the body, and an inverse correlation on the other side. Of course, this observation is complicated by the fact that it is difficult to extract the vertical acceleration component without knowing the orientation of the sensor. Furthermore, although the signal can be very noisy, the accelerations due to walking are likely to dominate the accelerations due to intra-body motion (such as arm swings or head turns) and we should be able to reliably make a determination that the supposed suite of sensors are on the same body.

Fortunately, there is already an existing body of work that shows how to do activity recognition given user-annotated data [2], and even on a mobile phone class device [4]; these techniques are particularly good at detecting when a user is walking. Our approach, therefore, is to detect periods when a user is walking by monitoring the accelerometer data periodically; when the data indicates the user is walking, we use Procedure 1 to collect accelerometer data from the sensors. (In Section 8 we discuss users who cannot walk.)

Lester et al. [11] provide a solution the one-body authentication problem, but only for sensors that are carried in the same location on the body. They also propose using accelerometers attached to each sensor and measure the *coherence* of the accelerometer data. "Coherence measures the extent to which two signals are linearly related at each frequency, with 1 indicating that two signals are highly correlated at a given frequency and 0 indicating that two signals are uncorrelated at that frequency" [11]. By looking at the coherence at the 1-10Hz frequencies (the frequency range of human motion), they can experimentally determine a threshold (e.g., coherence > 0.9) at which it is appropriate to deem two sensors as located on the same body.

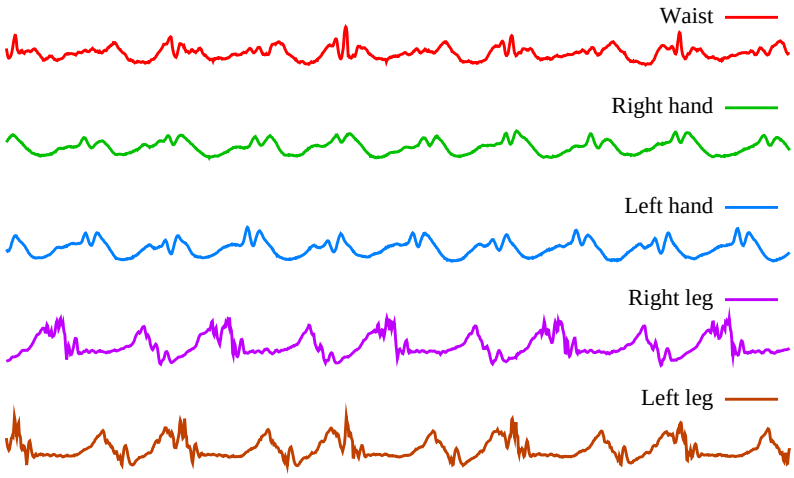


Fig. 1. Five seconds of magnitude data for each position on the body for one user

We extend Lester et al. [11] to sensors carried at different locations on the body – wrist, ankle, and waist – by using features often used for activity recognition. We then extract the pairwise *coherence* of features for the sensors on the same body. Given these coherences, we can train a classifier and use it to determine whether the alleged set of sensors are on the same body. We train our classifier to be as general as possible by using data collected from several individuals; the same model can then be used by all users for all sensor devices. We describe our method in more detail in the following section.

4 Method

As stated previously, we assume each sensor node has an accompanying accelerometer; our method uses only the accelerometer data. Specifically, consider a signal s sampled at some frequency such that:

$$s = \{(x_0, y_0, z_0), (x_1, y_1, z_1), \ldots\}$$

where x_i, y_i, and z_i are the three axes of the instantaneous acceleration, relative to gravity, at time i.

Because sensors might be mounted in different orientations, or might be worn in different orientations each time they are worn, we discount orientation by using the *magnitude* of the acceleration. Figure 3 shows that the magnitude exposes the overall walking motion well. Thus, we compute the magnitude of all three axes for all samples in s:

$$m_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$$

This gives us the rate of change of speed over time for that particular sensor node.

4.1 Feature Computation

We partition this orientation-ignored signal $\{m_0, \ldots, \}$ into non-overlapping windows of length w. For each window j, comprising $\{m_{jw}, \ldots, m_{jw+w}\}$, we extract seven common features (mean, standard deviation, variance, mean absolute deviation, inter-quartile range, power, energy); collectively, these seven values form the *feature vector* $F_j = (f_j^1, f_j^2, \ldots, f_j^7)$.

We chose these features primarily because others [12, 14] have used these features successfully to detect physical activities, and we hypothesize they would similarly be useful for our problem. If they can capture the physical activity of walking and we examine the correlation of these features, we should expect them to correlate if and only if they are attached the same body.

4.2 Coherence

Coherence is a measure of how well two signals correlate in the frequency domain. More precisely, it is the cross-spectral density of two signals divided by the auto-spectral density of each individual signal. Like Lester et al. [11], we approximate coherence by using the magnitude-squared coherence:

$$C_{xy}(\phi) = \frac{|S_{xy}(\phi)|^2}{S_{xx}(\phi)S_{yy}(\phi)}$$

In the above, x and y are the signals, S_{xy} is the cross-spectral density between signals x and y, S_{xx} is the auto-spectral density of signal x, and ϕ is the desired frequency. Cross-spectral density is calculated by the Fourier transform of the cross-correlation function. If x and y are well correlated at some frequency ϕ, then $C_{xy}(\phi)$ should be close to 1.

To get a final measure, we compute the normalized magnitude-squared coherence up to some frequency $\phi_{\max}$:

$$N(x, y) = \frac{1}{\phi_{\max}} \int_0^{\phi_{\max}} C_{xy}(\phi) d\phi$$

We chose $\phi_{\max} = 10$ because, as Lester et al. notes, "human motion rests below the 10Hz range" [11].

In addition, to compute the cross-spectral density over different frequencies, it is necessary to window the signals x and y. We choose a Hamming window of length equal to one-half of the size of the signals with no overlap.

4.3 Feature Coherence

Given two sets of feature matrices $A = (F_1, F_2, \ldots)$ and $B = (F_1, F_2, \ldots)$ with entries F_j as described above, we want to determine how well A and B are correlated. Here, A and B represent the *feature matrices* extracted from the accelerometer data of the mobile node and sensor node respectively.

We apply coherence to the feature matrices in the following manner. For some window length c (the feature coherence window), we compute the normalized coherence of A and B as such:

$$N_k^{AB} = \left\{N(A^1_{k\ldots k+c}, B^1_{k\ldots k+c}), N(A^2_{k\ldots k+c}, B^2_{k\ldots k+c}), \ldots, N(A^7_{k\ldots k+c}, B^7_{k\ldots k+c})\right\}$$

where $A^1_{k\ldots k+c} = \left\{f^1_n \in A : k \leq n < k+c\right\}$, the window of a specific feature of A. That is, we take each feature (i.e., a column of the matrix) of A and the corresponding feature of B, and compute the normalized coherence using c samples (i.e., the rows of the matrix). At this stage, we are left with a matrix of normalized coherences for each feature and window k.

Because we want to capture how the two signals are related over time, the coherence window c should be sufficiently large to capture periodicities in the features. Because the typical walk cycle is on the order of seconds, it is advisable to chose a coherence window on the order of several seconds.

4.4 Supervised Learning and Classification

To account for the many positions a sensor node might be placed on the body, we collect data from several locations. In our method, we compare the mobile node's accelerometer data to each other sensor node's accelerometer data. That is, the mobile node acts as a reference accelerometer, to which every other sensor node must correlate using the method described above. For a given set of locations and one reference location, we compute the feature coherence of each location (i.e., A in the above) relative to the reference location (i.e., B in the above). In our experiments, we compute the coherence of the right wrist and waist; left wrist and waist; left ankle and waist; and right ankle and waist. When we do this for one user, this yields feature coherences of the sensor on the same body, and we can label them as such. To yield feature coherences of sensors on different bodies, we take pairs of users and mix their locations. For example, at the waist and left hand there are two possible ways to mix up the sensors: Alice's waist and Fred's left hand, Fred's waist and Alice's left hand. By mixing locations for any pair of users, it is possible to compute an equal number of feature coherences that are and are not on the same body, labeling them as such.

Given a set of feature coherences and their respective labels, we can train a classifier to learn a model that is the coherence threshold for each feature. We employ support vector machines (SVMs) for this task since, once trained, they are good at predicting which label a given feature coherence is associated with. An SVM accomplishes this task by finding the hyperplane with the largest separation between the set of training feature coherences that are on the same body, and those that are not on the same body. In our experiments, we trained a support vector machine with a radial basis kernel using LIBSVM [5].

Given a trained SVM, we can use it to classify whether a given feature coherence is on the same body. That is, at the window the feature coherence was computed, the support vector machine can determine if the sensor node is on the same body as the mobile node. The SVM does so by determining on which side of the hyperplane the test feature coherence lies.

User	Walking Time (minutes:seconds)	Magnitude Samples	Feature Vectors
1	18:45	288017	9000
2	29:57	460047	14375
3	21:02	322962	10092
4	19:30	299553	9361
5	20:24	313215	9787
6	28:33	438484	13701
7	19:01	291974	9123

Fig. 2. Time spent walking, total acceleration samples, and number of features extracted for each user.

4.5 Classification Smoothing

The classification method described above makes an instantaneous classification of a feature coherence for that particular coherence window. It is, however, possible to boost the classification rates by examining a window of classifications over time. For example, if over the course of three classifications, two classifications positive and the third classification is negative, we can use a simple voting scheme to smooth over these misclassifications. In the example, because the majority of the classifications are classified as on the same body, we assume the sensor node is on the same body for that classification window. We can empirically determine the best window by varying the window and choosing the one that yields the best classification rates.

5 Dataset

We collected a set of accelerometer data, from several test subjects wearing sensors in several locations on their body, to use as training data (for the model) and to use as test data for (for our evaluation). We used WiTilt (version 2.5) accelerometers [15]. We followed the user with a laptop as they walked around a flat, predetermined course. The laptop was used to synchronize the accelerometer readings sent via Bluetooth by the WiTilt nodes.

We collected 2.5 hours of acceleration from 5 accelerometers sampled at 255Hz from seven users for a total of 13 hours of acceleration data. The average user walked for 22 minutes while wearing 5 accelerometers (waist, left wrist, right wrist, left ankle, right ankle). We chose the waist (specifically, the right pocket), because it represents a common location for the mobile node (cellphone). Of the likely locations for medical sensors (arms, legs, chest, head) we chose the wrists and ankles for our experiments because (as extremities) we expect they would raise the most difficult challenge for our method. Figure 2 gives more detailed information about how much data was collected for each user.

6 Evaluation

We evaluate how well our method performed for each location, at the wrists only, at the ankles only, on the left side of the body, on the right side of the body, and at all locations.

For each experiment we used only the data from that location, or type of location, for training and for evaluation; for example, in the "left leg" case we train on (and test on) the accelerometer data from the left ankle in comparison to the data from the waist. In neither the learning process nor in the operation of our system was the data labeled as to which location produced the acceleration data. We varied the coherence window size from 2 to 16 seconds.

Using these datasets, we performed two types of cross-validations to evaluate the accuracy of our method. The first cross-validation we performed was a simple 10-fold cross-validation. A *k-fold cross-validation* partitions the dataset into k partitions, trains the classifier over $k - 1$ of the partitions (the training set) and classifies the remaining partition (the testing set), repeating this procedure for each partition. This type of cross-validation will tell us how well our classifier generally performs since it will classify every sample in the dataset. The second cross-validation we performed is a variant of leave-one-out cross-validation we call leave-one-user-out cross-validation. A *leave-one-user-out cross-validation* leaves an entire user's data out as the testing set and trains the classifier using the remaining data. We then test the classifier using the left-out user's data, repeating this procedure for each user. This type of cross-validation will tell us how general our classifier is. Ideally our classifier would not be user-specific, and would perform well in the case of a never-before-seen user.

We define a *true feature coherence* as a feature coherence computed from a sensor node and mobile node on the same body, and a *false feature coherence* as a feature coherence computed from a sensor node and mobile node *not* on the same body. A *positive classification* means the classifier determined that the given feature coherence indicates the sensor node and mobile node were on the same body, while a *negative classification* means the classifier determined that the given feature coherence indicates the sensor node and mobile node were *not* be on the same body. It follows, then, that a *true positive* occurs when a true feature coherence is classified as positive, and a *true negative* occurs when a false feature coherence is classified as a negative. A *false positive* occurs when a false feature coherence is classified as positive, and a *false negative* occurs when a true feature coherence is classified as negative.

We present the accuracy, precision and recall for each possible scenario. *Accuracy* is the sum of true positives and true negatives over the total number of classifications. Accuracy tells us how well our classifier is doing at classifying feature coherences. *Precision* is the number of true positives over the total number of positive classifications. Precision tells us how well our classifier is able to discriminate between true and false positives. *Recall* is the number of true positives over the sum of true positives and false negatives. Recall tells us how well our classifier classifies true features coherences.

In all of our experiments, we chose a feature window size of 17 acceleration magnitudes with no overlap so that each second may be divided evenly and thus yield 15 features per second. We present results using our dataset for both our method and the method used in Lester et al. [11] for sake of comparison.

6.1 Our Method

We ran a 10-fold cross-validation using the data from all users and for each specified location, resulting in Figures 3(a), 3(b), and 3(c). The results show how the choice of

coherence window size affects the accuracy, precision and recall. A smaller window is more desirable because the coherence window size is directly proportional to the window of accelerometer data that needs to be transmitted to the mobile node, and wireless communication is typically expensive. However, a smaller window will not capture the periodicity of walking. According to Figure 3(a), a 4–6 second coherence window, or about 60–90 feature values, performed the best and minimized the communication overhead. In such cases our method was about 70–85% accurate.

In general, as the coherence window length increases the accuracy briefly climbs then settles down, precision increases steadily, and recall drops significantly. Given a longer coherence window length, this means the classifier is more likely to make negative classifications rather than positive ones. Since a longer coherence window means more walking cycles are taken into account, it also means there is more opportunity for the signals to differ due to accumulated noise and/or a change in walking style in accordance with the environment.

These plots show that the method was more accurate for the legs than for the hands, which is not surprising because the legs have more consistent motion behavior during walking than do the hands, particularly across users. The right leg (or left hand) seemed to do better than the left leg (or right hand, respectively), perhaps because the waist accelerometer was always carried in the right pocket, and most people swing their hands in opposition to their legs. When the hands and legs were combined, as in the left-body and right-body cases, this effect was cancelled out and the results of both were fairly similar to the all-body case.

In Figure 3(d), we ran a leave-one-user-out cross-validation for each user with a fixed coherence window of 6 seconds. The accuracy, precision, and recall for all users are nearly identical, thus providing some evidence that our trained model is not specific to any user, and can in fact be used to predict a never-before-seen user.

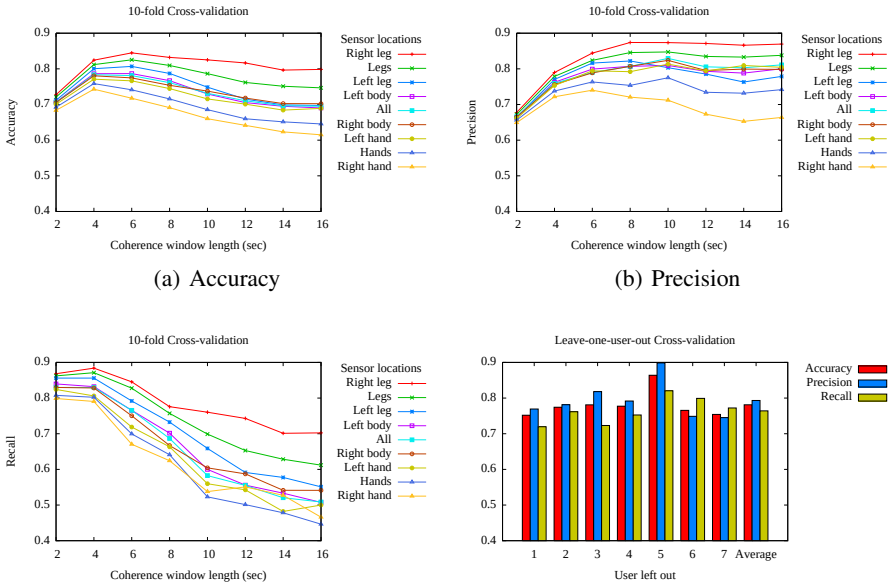
6.2 Lester et al. Method

For comparison's sake, we implemented the method described in Lester et al. [11], after extending it to use a support vector machine for determining the threshold instead of choosing an arbitrary threshold. Figure 4 shows that for any of the given locations, their method has poor classification rates, little better than random guess (0.50).

Lester et al. [11] do present results for "devices at other locations on the body, including accelerometers on the wrist, placed in one or both pockets, in a backpack, and in a fanny pack." These placements, however, are in the same relative location and therefore not comparable. Furthermore, we evaluated the scheme over longer time intervals, and averaged the results for a specified window.

6.3 Classification Smoothing

We now return to the leave-one-user-out experiments, as they most closely model how the method would be used in practice. In these experiments, for each user left out (the testing set), we used the model trained on all other users' data to predict the testing set. Now, instead of instantaneous prediction, we use a simple majority vote to smooth over



(a) Accuracy (b) Precision

(c) Recall (d) Leave-one-user-out

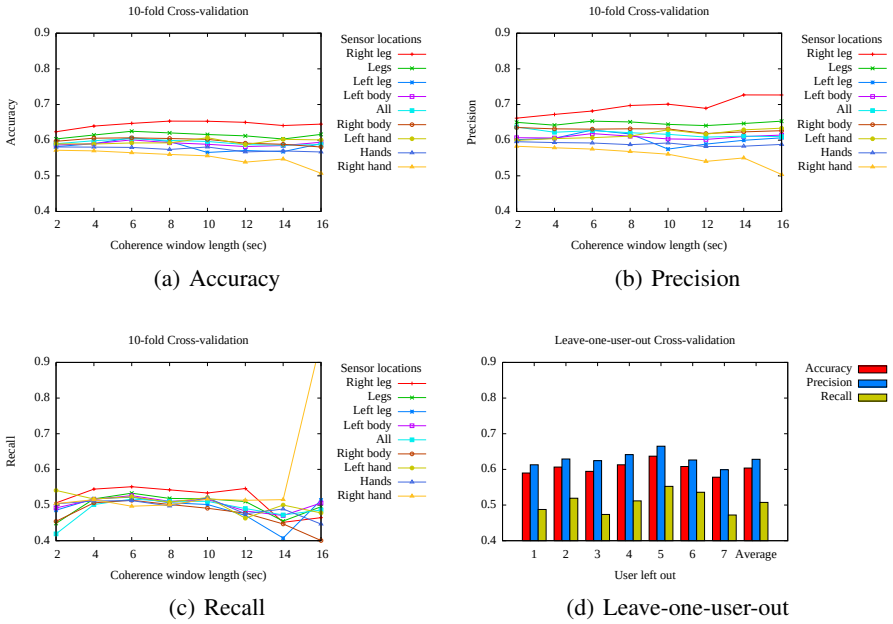
Fig. 3. Evaluation of our method. Subfigures (a), (b), and (c) were computed from a 10-fold cross-validation of all users at the specified locations and coherence windows. Subfigure (d) was computed from a leave-one-user-out cross-validation for each user with a coherence window of 6 seconds.

classifications and plot how well this smoothing performed for a given window size of classifications.

Figure 5 shows the average accuracy, precision, and recall over all users for varying classification windows with a fixed coherence window of 6 seconds. Our method, Figure 5(a), benefits slightly from classification smoothing as does Lester et al.'s method, Figure 5(b). This result tells us that our method makes sporadic mis-classifications that can be reduced with smoothing. Like any smoothing scheme, one must strike a balance between the size of a smoothing window and the desired classification rates. For our method, a 42 second smoothing window, or 7 feature coherences, modestly boosts our instantaneous classification rates by 8%.

7 Related Work

Mayrhofer et al. [13] provide a solution to exchange a cryptographic key between two devices by manually shaking the two devices together. They use the method described in Lester et al. [11] to determine whether two devices are being shaken together. But, as they notice, coherence "does not lend itself to directly creating cryptographic key material out of its results" [13]. To extract key material they extract quantized FFT coefficients from the accelerometer data to use as entropy for generating a key. Our



(a) Accuracy (b) Precision

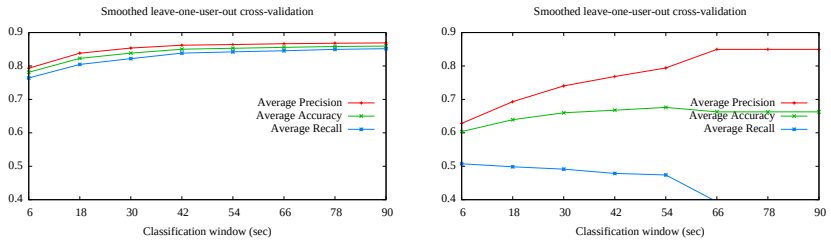
(c) Recall (d) Leave-one-user-out

Fig. 4. Evaluation of Lester et al. method. Subfigures (a), (b), and (c) were computed from a 10-fold cross-validation over all users at the specified locations and coherence window lengths. Subfigure (d) was computed from a leave-one-user-out cross-validation for each user with a coherence window of 6 seconds.

problem is more difficult because the accelerometers are not being shaken together but are attached to a body and will therefore experience less-correlated accelerations.

Kunze et al. [10] provide a method for determining where on a body a particular sensor is located. They detect when a user is walking regardless of the location of a sensor, and by training a classifiers on a variety of features (RMS, frequency range power, frequency entropy, and the sum of the power of detail signals at different levels) on different positions on the body they can use the classifier to determine where on the body the sensor is located. We seek to provide a method that determines whether a suite of sensors is located on the same body without having to use multiple classifiers for different body locations. It might be the case that knowing the location of a sensor node could boost our classification rates, but we leave that for future work.

Kunze et al. [9] provide similar methods to account for sensor displacement on a particular body part. This problem is difficult primarily because "acceleration due to rotation is sensitive to sensor displacement within a single body part" [9]. To alleviate this problem, the authors observe that "combining a gyroscope with an accelerometer and having the accelerometer ignore all signal frames dominated by rotation can remove placement sensitivity while retaining most of the relevant information" [9]. We choose to limit our approach to accelerometers; although the inclusion of a gyroscope might



(a) Our method (b) Lester et al. method

Fig. 5. Average accuracy, precision and recall over all users for different classification windows with a fixed coherence window of 6 seconds.

increase accuracy, it would also increase the size, cost, and energy consumption on each sensor device.

Sriram et al. [16] provide a method to authenticate patients using electrocardiography and acceleration data for remote health monitoring. While electrocardiography has proven to be useful for authentication, they observe that these methods do not perform well in the real world because physical activity perturbs the electrocardiography data. By employing an accelerometer to differentiate physical activities, they can use electrocardiography data from those physical activities to authenticate patients. We both make the observation that "the monitoring system needs to make sure that the data is coming from the *right person* before any medical or financial decisions are made based on the data" [16] (emphasis ours). Our work is complementary since it is necessary to establish that accelerometer is on the same body as the sensor used to collect electrocardiography data. Their method extracts 50 features from the electrocardiography and accelerometer data and uses these features to train two types of classifiers, k-Nearest Neighbor and a Bayesian Network, whose output can be used for identification and verification. We follow a similar procedure except that we work exclusively with accelerometer data, again, to reduce the complexity and cost of the solution. We also look at the correlation between sensors, whereas they assume there is a prior profile of the patient's combined electrocardiography and accelerometer data.

8 Discussion and Future Work

There are a variety of existing technologies one could imagine using to solve the weak one-body authentication problem. For example, one could employ a wireless localization technique to ensure the sensors nodes are within some bodily distance. The body, however, might block all or some of the wireless signal thereby limiting localization, nor is it clear how these kind of techniques would provide confidence to a physician that the data is coming from one body. Similarly, one can trivially use a form of body-coupled communication [3], but the security properties these type of communication mediums provide are not well understood. If two users were to hold hands, for example, would they be considered one body?

When two people are walking together, it is a common natural phenomenon for two walkers to synchronize their walking patterns. It is unclear whether our method will be fooled by such a situation, mis-classifying Alice's and Fred's sensor devices as being on the wrong body. The first dataset we captured to test this method actually employed one user trying to mimic the gait of another user, and our first results showed our algorithm not being fooled by this. This case, however, requires exploration in a larger dataset.

Our method relies on the assumption that a user is capable of walking, which may not be true for some users. It remains as future work to determine whether we can extend the method for a person who is confined to a wheelchair, for example. Even for a user who is able to walk, there may be an extended period of time after binding a sensor node and before the user walks. It may be necessary for the mobile node to alert the user that they should walk around so that authentication can be performed. As future work, we may explore other acceleration events; for example, to ask the user for clap their hands, or perform some unique movement.

Ideally the algorithm should be tuned to produce more false negatives (i.e., the algorithm determined the sensor nodes to be on different bodies when they really were on the same body) than false positives (i.e., the algorithm determined the sensor nodes to be on the same body when they were actually not) because the consequences of a false positive (recording the wrong person's data in someone's health record) are more severe than the consequences of a false negative (losing data). It may be possible to 'bias' the SVM toward false negatives by adding a margin to its hyperplane-testing function.

Although we do not discuss encryption mechanisms, ensuring data confidentiality is paramount in any health-related scenario. If one were to optimize the authentication phase by simultaneously authenticating all bound sensor nodes, it might be necessary to encrypt the acceleration data to avoid replay attacks (in which the adversary replays one node's acceleration data in hopes that its rogue sensor node will be authenticated as being on the same body as the victim). Even if such an attack is discounted, the accelerometer data itself might be privacy sensitive because accelerometer data may be used to recognize a victim's activity. Some activities are clearly privacy sensitive, and some of those sensitive activities might be detected from accelerometer data alone.

In a practical system, one must consider energy and computational costs. In our model, the sensor node sends raw acceleration data to the mobile node. If this proves to be too expensive, then the sensor node could compute features from a window of acceleration and communicate those features instead. We leave exploring this delicate balance between extendability (allowing use of other features in the future), computability (due to limited computational capabilities on a sensor node), and energy requirements (with trade-offs specific to the technology in a sensor node) as future work. In terms of the mobile node, we assume the cellphone will be more than capable of computing correlations, but the energy costs of these functions is unknown and require more careful analysis. Should the computation prove to be too expensive or time consuming, then one may need to explore optimizations or approximations or the assistance of a back-end server, with due consideration to the trade-off of computational overhead, accuracy, and privacy.

9 Conclusion

Mobile health will play a major role in the future of healthcare. Wearable health sensors will enable physicians to monitor their patients remotely, and allow patients better access to information about their health. The method presented in this paper provides the foundation for any mobile-health system, because, in order for the data to be useful, physicians need confidence that the data supposedly collected about a patient actually came from that patient. We provide the first step in that verification process: generically authenticating that all the sensor nodes bound to a mobile node are the same body. We show that our method can achieve an accuracy of 80% when given 18 seconds of accelerometer data from different locations on the body, and our method can be generically applied regardless of the sensor type and without user-specific training data. In summary, we make the following contributions:

- We describe a novel problem in the mobile healthcare domain and provide a solution to the weak version of the one-body authentication problem.
- We extend Lester et al. [11] to sensors carried at different locations on the body – wrist, ankle, and waist – by extracting used for activity recognition.
- We provide empirical results to our solution using a dataset of seven users walking for 22 minutes to show that it is feasible.

Acknowledgements

This research results from a research program at the Institute for Security, Technology, and Society at Dartmouth College, supported by the National Science Foundation under Grant Award Number 0910842 and by the Department of Health and Human Services (SHARP program) under award number 90TR0003-01. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the sponsors.

We also thank the anonymous reviewers, and our colleagues in the Dartmouth TISH group, for their valuable feedback.

References

[1] Apex Fitness. *BodyBugg*, October 2010. `http://www.bodybugg.com/`.

[2] L. Bao and S. S. Intille. Activity Recognition from User-Annotated Acceleration Data. In *Proceedings of the Third International Conference on Pervasive Computing (Pervasive)*, pages 1–17, 2004.

[3] A. T. Barth, M. A. Hanson, J. Harry C. Powell, D. Unluer, S. G. Wilson, and J. Lach. Body-coupled communication for body sensor networks. In *Proceedings of the ICST 3rd international Conference on Body Area Networks*, BodyNets '08, 2008.

[4] T. Brezmes, J.-L. Gorricho, and J. Cotrina. Activity Recognition from Accelerometer Data on a Mobile Phone. In *Proceedings of the Tenth International Work-Conference on Artificial Neural Networks (IWANN)*, pages 796–799, 2009.

[5] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[6] C. Cornelius and D. Kotz. On Usable Authentication for Wireless Body Area Networks. In *Proceedings of the First USENIX Workshop on Health Security and Privacy (HealthSec)*, 2010.

[7] Fitbit, Inc. *Fitbit*, October 2010. `http://www.fitbit.com/`.

[8] Freescale Semiconductor. *Freescale Xtrinsic accelerometers optimize resolution and battery life in consumer devices*, September 2010. Press release available at `http://media.freescale.com/phoenix.zhtml?c=196520&p=irol-newsArticle&ID=1470583`.

[9] K. S. Kunze and P. Lukowicz. Dealing with sensor displacement in motion-based onbody activity recognition systems. In *Proceedings of the Tenth International Conference on Ubiquitous Computing (UbiComp)*, pages 20–29, 2008.

[10] K. S. Kunze, P. Lukowicz, H. Junker, and G. Tröster. Where am I: Recognizing On-body Positions of Wearable Sensors. In *Proceedings of the First International Workshop on Location- and Context-Awareness (LoCa)*, pages 264–275, 2005.

[11] J. Lester, B. Hannaford, and G. Borriello. "Are You with Me?" - Using Accelerometers to Determine If Two Devices Are Carried by the Same Person. In *Proceedings of the Third International Conference on Pervasive Computing (Pervasive)*, pages 33–50, 2004.

[12] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher. Activity Recognition and Monitoring Using Multiple Sensors on Different Body Positions. In *Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, pages 113–116, 2006.

[13] R. Mayrhofer and H. Gellersen. Shake Well Before Use: Intuitive and Secure Pairing of Mobile Devices. *IEEE Transactions on Mobile Computing*, 8(6):792–806, 2009.

[14] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman. Activity Recognition from Accelerometer Data. In *Proceedings of the Twentieth National Conference on Artificial Intelligence (AAAI)*, pages 1541–1546, 2005.

[15] SparkFun Electronics. *WiTilt v2.5*, October 2010. Data sheet available at `http://www.sparkfun.com/datasheets/Sensors/WiTilt_V2_5.pdf`.

[16] J. C. Sriram, M. Shin, T. Choudhury, and D. Kotz. Activity-aware ECG-based patient authentication for remote health monitoring. In *Proceedings of the Eleventh International Conference on Multimodal Interfaces (ICMI)*, pages 297–304, 2009.