College of William & Mary
Department of Computer Science

WM-CS-2012-04

Remora: Sensing Resource Sharing Among Smartphone-based Body Sensor Networks

Matthew Keally, College of William and Mary
Gang Zhou, College of William and Mary
Guoliang Xing, Michigan State University
Jianxin Wu, Nanyang Technological University

July 29, 2012

Abstract

In many body sensor network (BSN) applications, such as activity recognition for assisted living residents or physical fitness assessment of a sports team, users spend a significant amount of time with one another while performing many of the same activities. We exploit this physical proximity with Remora, a smartphone-based Body Sensor Network activity recognition system which shares sensing resources among neighboring BSNs. Compared to individual BSNs, Remora resource sharing provides increased accuracy and significant energy savings. To increase classification accuracy, Remora BSNs share sensors by overhearing neighbors' sensor data transmissions. When sharing, fewer on-body sensors are needed to achieve high accuracy, resulting in energy savings by turning off unneeded sensors. To save phone energy, neighboring BSNs share classifiers: only one classifier is active at a time classifying activities for all neighbors. Remora addresses three major challenges of sharing with physical neighbors: 1) Sharing only when the energy benefit outweighs the cost, 2) Finding and utilizing the shared sensors and classifiers which produce the best combination of accuracy improvement and energy savings, and 3) Providing a lightweight and collaborative classification approach, without the use of a backend server, which adapts to the dynamics of available neighbors. In a two week evaluation with 6 subjects, we show that sharing provides up to a 30% accuracy increase while extending phone battery lifetime by over 65%.

1 Introduction

Specialized personal sensing applications, especially in the context awareness and activity recognition domain, are ideally suited for body sensor network (BSN) deployments. Specifically, the wide variety of sensor modalities available for on-body nodes provide sensing capability that far exceeds using smartphones alone. Additionally, the use of a smartphone in conjunction with external on-body nodes provides additional sensing power, computational capability, portability, and a user-friendly interface for personal control and runtime feedback. Activity recognition applications that can exploit such a system include assisted living [33], physical fitness assessment [1], and patient monitoring [20] [6]. A physician may administer BSNs for retirement community residents [33] [29] to detect depression, ensuring proper eating, social activity, and exercise. Similarly, a university sports team coach may deploy BSNs on his or her student-athletes to ensure optimal performance [2]. The BSN worn by each student-athlete can not only measure athletic performance but also detect daily living habits that may be detrimental, such as excessive social activity or lack of studying.

Smartphone-based BSN applications which use activity recognition to assess daily living habits, such as those mentioned above, demand high classification accuracy and long system lifetimes. However, many individual BSNs may exhibit poor accuracy due to specific user behavior, background noise, and even difficult to classify activities. For example, an activity classifier may be easily confused between a meeting with colleagues and watching television. Furthermore, smartphone batteries are quickly drained after 8-10 hours of BSN use [16], thus requiring frequent recharges.

Interestingly, users of many BSN applications may be in close proximity to one another, belonging to groups with strong interpersonal ties. Many residents of a retirement community are close friends and engage in many activities together. Athletes on a university sports team will not only practice together but live, eat, study, and party together. In our motivational study and evaluation, we find that subjects spend between 20-50% of waking hours in the proximity of a close friend, family member, or colleague.

Consequently, we propose that BSNs in physical proximity to one another opportunistically share resources. In this paper, we focus on activity recognition and share neighboring resources to extend device lifetimes and increase activity classification accuracy. BSN neighbors, such as family and friends, exploit overheard on-body sensor data transmissions to increase classification accuracy. Increased accuracy by sharing is possible due to available neighbor sensors that are both individually accurate and have complimentary classification capabilities. Through sharing, neighbors can use fewer sensors, allowing more to be disabled to save energy. Furthermore, to increase phone battery life, classifiers are duty cycled among neighbors, allowing the phone to go into a low power sleep.

However, three prominent challenges arise from sharing resources among neighboring BSNs. First, we must determine when sharing provides an energy benefit and when it does not. We must accurately characterize sharing costs and benefits as well as predict when neighboring BSNs will be together long enough to achieve such benefits. Second, we must identify the cases where sharing improves accuracy, such as difficult to classify activities, then find and utilize the resources that provide the best combination of accuracy and energy savings. Lastly, we must provide a lightweight and flexible sharing approach to limit sharing overhead and adapt to the dynamics of available neighbors. This sharing approach must provide an activity classification method which efficiently addresses changes in sensor availability over time. Neighboring BSNs must also easily collaborate to decide which sensors and classifiers to share.

Most existing approaches to activity classification ignore sharing altogether, whether using smartphones [22] [23] or on-body sensors [16] [30]. Other approaches rely extensively on backend servers [4] [18] [11] for classifier training and dissemination of classifiers to phones. One effort [24] shares classifiers and classification results among neighbors but the energy costs and benefits are not fully addressed.

Towards addressing the above challenges and shortcomings, we show through an initial experiment how sharing sensors among neighboring BSNs can increase activity classification accuracy and save sensor energy by using fewer sensors. The insights gained from this experiment motivate the design of our system, Remora. Remora is an opportunistic resource sharing approach which improves classification accuracy and extends system lifetime among BSNs in proximity to one another. With Remora, we first determine the costs and benefits of sharing: we determine energy overhead as well as the proximity duration needed for the sharing energy benefit to outweigh the energy costs. Next, we provide a sharing-aware classification approach which uses an ensemble classifier that efficiently adapts to changes in neighbor and sensor availability. This approach allows sharing BSNs to jointly select sensors to maximize training accuracy and use as few sensors as possible to save sensor energy. To save phone energy, sharing BSNs only use one active classifier per time period. Our main contributions are:

- We analyze the overhead of sharing sensors and classifiers with a time and energy model, only sharing when neighboring BSNs will be together long enough for sharing to benefit.
- We provide an efficient method to share sensors and classifiers among neighboring BSNs. A collaborative approach allows neighbors to share only a small set of accurate and complimentary sensors and duty cycle classifiers to save phone energy.

- With two weeks of evaluation from six subjects, in comparison with using only individual BSN resources, Remora can increase activity classification accuracy by up to 30% and extend battery lifetime by over 65%.

This paper is organized as follows: Section 2 presents related work. We explore the feasibility of sharing and present a motivational study in Section 3. Using our motivational experiment, we present our Remora design and example applications in Section 4. In Section 5, we discuss how BSNs detect neighbors and analyze sharing costs and benefits. We describe our Sharing-Aware Classification approach in Section 6, evaluate Remora performance in Section 7, and present conclusions and future work in Section 8.

2 Related Work

Several collaborative sensing and classification approaches directly share resources among users, but none use sharing to achieve both high accuracy and energy efficiency. In [17], nearby drivers exchange traffic light data to determine optimal driving speed. Speaker recognition classifiers are combined among phones in physical proximity to each other in [24], which increases accuracy. However, an expensive classification method is used which requires the use of a backend server for training. Significant overhead is consumed transmitting trained classifiers from the server to the phone as well as when combining classifiers among phones.

Other collaborative approaches do not directly share sensing resources; all information is relayed or processed using backend servers. This approach is used in collaborative approaches for video editing [4], group activities [3], and a generic opportunistic framework [9]. To increase accuracy, one approach [18] shares classifiers among users with similar behaviors. Other works which use backend servers [29] [12] provide sensing quality tradeoffs, such as an adaptive sampling rate, to save energy.

Many existing on-body sensing and activity classification approaches do not allow any collaboration among users. On-body sensors are used for classification [1] [20] [6], some of which [16] [30] provide energy saving methods. Other approaches [22] [23] use only smartphone sensors for activity classification. A phone-only classification technique [8] provides an explicit energy-latency-accuracy tradeoff, while other smartphone methods [25] [31] [29] achieve energy savings with adaptive sampling.

Other works investigate interactions between multiple subjects but do not address the data or resources being shared as well as accuracy or energy concerns related to such resources. These include user proximity [10] [19], intercontact time [13] [15], mobility prediction [7] [26], and protocols for proximity-based mobile device pairing [27] [14].

3 Feasibility and Motivation

In this section, we discuss the intuition behind our Remora BSN resource sharing system. We first define our activity recognition application, hardware platform, and design goals. We next discuss the feasibility of sharing and how sensors and classifiers are shared among BSNs. Then, in a short experiment, we show the potential accuracy and energy benefits of sharing.

Problem Statement. In this paper, we provide a personal activity recognition application and target personal activities such as walking, working at a desk, having a meeting with colleagues, driving a car, or watching TV. To classify activities, we use smartphone-based body sensor networks, combining on-body wireless sensors with the additional sensing capability, computational resources, and user interface of a smartphone. Using BSN hardware, our goal is to provide an activity recognition system which allows neighboring BSNs to share resources in order to increase activity classification accuracy while saving phone and on-body sensor energy.

3.1 Feasibility

We address three issues that affect BSN resource sharing and its success in increasing classification accuracy and providing energy savings. We first demonstrate that in real scenarios, there is enough opportunity for BSN neighbors to share and provide a benefit. Second, we describe available sensing and classification resources and how best to share them among neighbors. Lastly, we address privacy concerns.

Sharing Opportunity. Our approach targets applications where the users have strong interpersonal ties. In such scenarios, there is sufficient interaction between neighboring BSNs such that sharing can provide a significant impact on overall classification performance and energy savings. In the MIT Reality Mining dataset [10], physical interactions of nearly 100 subjects were recorded using Bluetooth phones over 9 months. Most subjects were friends or colleagues: students and faculty who worked in the same building and spent time together off campus. Analysis of the MIT data yields that, on average, 25% of the time each subject is in proximity with at least one other subject. Our evaluation in Section 7 yields similar results: for each subject, 30-50% of the time was spent in proximity with at least one neighbor.

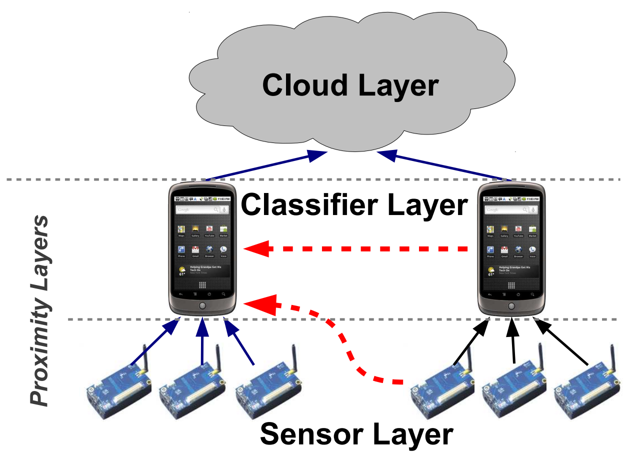

Figure 1: Sharing Hierarchy. We focus on the proximity layers, sharing classifiers and sensors among neighbors.

Sharing Hierarchy. In Figure 1, we present a hierarchy of BSN resources which are eligible for sharing among users. We use the bottom two (Sensor and Classifier) to exploit proximity, improving accuracy and energy use. In the Sensor Layer, when 2 or more BSNs are in proximity to each other, the phone for each BSN overhears the transmissions of the others' sensor nodes. Neighboring BSNs freeride, opportunistically using the overheard data directly to train their own classifiers and make activity classification decisions. Through collaboration, neighbors select a set of sensors that achieves higher accuracy and uses fewer combined sensors compared with individual classification.

Previous work demonstrates that continuously active classification can drain a phone battery in as little as 8 hours [16], so extending phone lifetime is critical. At the Classifier Layer, neighbors duty cycle classifiers so that at any given time, only one active classifier is running, allowing all other phones to go into a low power sleep state. Since the active classifier makes classification decisions for all neighbors, neighbors only share if they are all performing the same activity. However, as demonstrated by our evaluation, neighbors in close proximity are likely to be performing the same activity. Additionally, a short duty cycle time allows quick detection of activity changes while sharing.

Most existing efforts for sensing resource sharing perform at the top level, or Cloud Layer, where shared classifiers are trained on a backend server [24] or data is relayed [9] among different users. However, most cloud-level systems rely on expensive classification algorithms, may incur high communication overhead to transfer sensor and classifier information, and also do not fully exploit proximity.

Privacy. We provide several features to address privacy concerns. First, previous work has established that people are more likely to share with others in close physical proximity [32], such as friends and colleagues. Because users share only with neighbors in physical proximity, sharing neighbors are already able to visually identify the activities being performed. Second, a user can define "private" sensors which are not shared while "public" sensors are shared and broadcast data to neighbors. For example, a user may refuse to share a wireless heart rate monitor or pulse oximeter, defining such sensors as private. Furthermore, sensors such as these may not help neighbor classification accuracy. Third, each on-body node aggregates sensor data samples before transmitting, providing coarse-grained aggregated data to both the local and neighboring BSNs. A similar technique [9] is used to obfuscate personally identifiable characteristics for phone sensor data.

3.2 Motivation

To motivate our Remora design, we present a pilot study which demonstrates that sharing resources among BSNs yields significant accuracy improvement and energy savings. We first detail our experimental configuration, which we extend upon in the evaluation in Section 7. Then, we show results from our motivation experiment in which two subjects wear BSNs simultaneously, performing shared activities.

3.2.1 Experimental Configuration

Each subject in our experiments wore four TinyOS-based Crossbow IRIS motes, shown in Figure 2. Each mote is wirelessly linked to a TelosB base station, which is connected via USB to an Android HTC Hero smartphone. Our solution can be extended beyond the research-based TinyOS devices to work with more ergonomic commercial sensors. We present details on sensors, sampling, and classification:

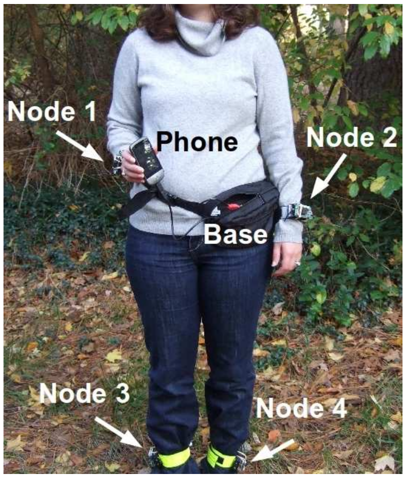

Figure 2: BSN configuration: 4 on-body wireless sensor nodes communicate with a base station node which is attached to an Android phone.

Node	Location	Sensors
Phone	R. Waist	3-Axis Acc., GPS/WiFi (velocity)
IRIS	L. Wrist	2-Axis Acc., Mic., Light, Temp.
IRIS	R. Wrist	2-Axis Acc., Mic., Light, Temp.
IRIS	L. Ankle	2-Axis Acc., Mic., Light, Temp.
IRIS	R. Ankle	2-Axis Acc., Mic., Light, Temp..

Table 1: Deployment Configuration.

Sensors. The sensor configuration is summarized in Table 1. On the phone, which we attach to the waist, we make use of the 3-axis accelerometer as well as velocity from WiFi and GPS, with GPS active only when WiFi is unavailable. On the mote, we use an MTS310 sensorboard with the following sensors: 2-axis accelerometer, microphone, light, and temperature. In addition to the sensors on the mote, the base station also collects RSSI information from each received packet, which has been previously demonstrated [28] to provide insight into body posture. Each subject makes all on-body sensors public (shared with all neighbors) and all phone sensors private (used by the local BSN only).

Sampling and Aggregation. For the microphones and accelerometers, raw ADC values are sampled at 20ms intervals to ensure quick body movements can be captured, with light and temperature ADC readings sampled at 1s intervals, and GPS/WiFi sampled every 10s. To reduce communication overhead, data for each sensor is aggregated locally on each node at 10s intervals, which is well within the time granularity of the activities we classify. During local aggregation, light and temperature sensor readings are averaged since these sensor readings remain relatively stable for each activity. Except for GPS/WiFi, all other sensors compute the difference between the highest and lowest readings for each aggregation interval, for the change in readings indicate body movement or sound.

Aggregated data for all sensors on a mote is combined into a single packet and broadcasted to the local phone and any neighboring phones. Motes transmit at the lowest available sending power to save energy and reduce congestion while a reliable communication scheme with the local phone eliminates packet loss with fewer than 1% retransmissions.

Classification. At each aggregation interval, aggregated data is used to classify activities with a Bootstrap Aggregating (Bagging) [5] classifier, detailed in Section 6. During the experiment, subjects recorded

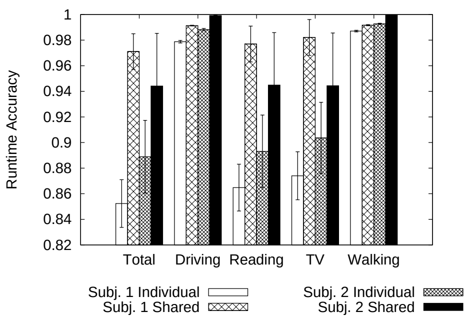

Figure 3: Runtime accuracy for shared and individual BSN configurations.

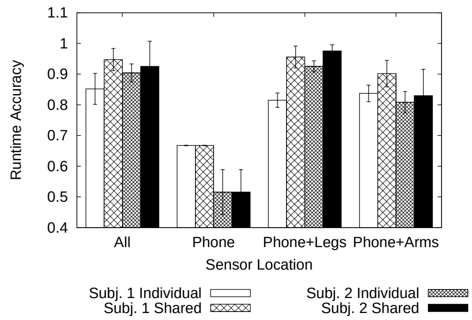

Figure 4: Accuracy for different combinations of on-body sensors.

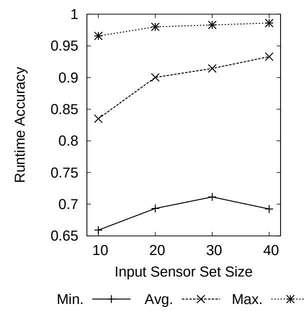

Figure 5: Randomly selected sensor clusters.

all activity ground truth in order to evaluate the accuracy of training data (training accuracy) and runtime accuracy.

3.2.2 Identifying Sharing Benefits

Through a shared activity experiment with 2 BSNs, we show how sharing can improve accuracy and provide energy savings. Two subjects performed four shared activities (driving, reading, walking, and watching TV) for over four hours in length. We use the same data to compute both individual and shared classification results, using 10 initial observations per activity as training data. Since Bagging trains nondeterministically, we plot average runtime accuracy and standard deviation over 30 runs in Figure 3, demonstrating stable performance. From the figure, when both BSNs share each other's sensors, this results in a total accuracy increase of 12% points for Subject 1 and 5% points for Subject 2. This is because the reading and watching TV activities are performed in the same room and are often confused when only individual sensors are available. However, due to their different locations, sensors from a neighboring BSN provide complimentary information and can be exploited to provide higher accuracy for both activities.

In Figure 4, we compare the accuracy of sensors at different body locations. The figure shows that on-body sensors improve accuracy significantly compared with using phone sensors only. For both the individual and shared scenarios, accuracy is improved by over 25% when using all available on-body sensors. Leg sensors give the greatest boost, for they remain still during sitting activities and in motion while walking, which is easily captured by accelerometers.

Lastly, we show that we can save energy by choosing only the most capable sensors and turning off unneeded sensors. For Subject 1, we generate 100 random sensor clusters of sizes 10 through 40 from all available sensors, including public sensors on Subject 2. Training classifiers for each cluster, we plot the minimum, maximum, and average accuracy in Figure 5. The figure shows that if we only choose 10 sensors, we can still achieve 97% accuracy, as long as we choose the right 10 sensors. This result motivates us to provide an efficient sensor selection approach for shared BSNs, described in Section 6.2, that chooses a small number of sensors to achieve both high accuracy and node energy savings.

4 Architecture and Applications

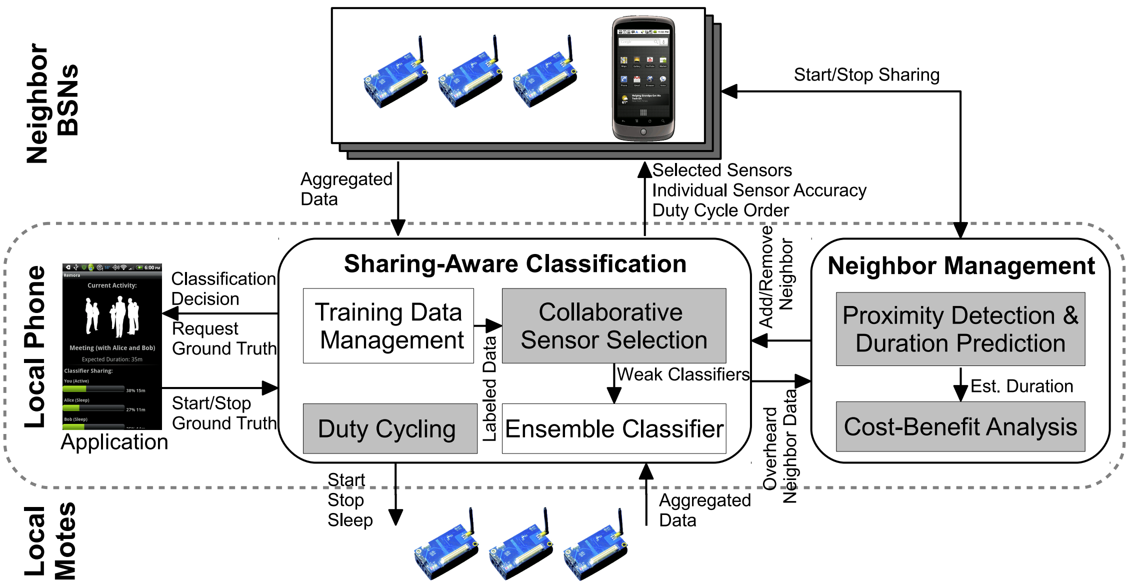

Figure 6: Remora Architecture. Neighbor Management determines if sharing with detected neighbors will provide an energy benefit. Sharing-Aware Classification collaborates with neighbors to select sensors for shared classification, classifies sensor data from the local phone, local motes, and neighboring motes, and duty cycles classifiers among neighboring phones.

With our goal of energy and accuracy gains through BSN resource sharing, we present the Remora system architecture in Figure 6. Each BSN consists of TinyOS-based motes and an Android smartphone with no reliance on a backend server. For each BSN, multiple on-body motes (Local Motes) communicate wirelessly with the phone (Local Phone). While our system uses a USB-connected base station as an 802.15.4 relay between other phones and motes, Remora can also use other communication modalities, such as Bluetooth.

During runtime, Neighbor Management detects neighbors and initiates sharing with neighbors only if sharing will provide an energy benefit. Sharing-Aware Classification trains classifiers and classifies activities using local sensors as well as neighbor sensors made available by Neighbor Management. Sharing-Aware Classification also duty cycles classifiers among sharing BSNs to save phone energy. We now describe the core of our Remora system, with our significant contributions highlighted in gray in Figure 6:

Neighbor Management. The Neighbor Management module characterizes the costs and benefits of sharing and initiates sharing only when an energy benefit is possible. Proximity Detection and Duration Prediction detects neighboring BSNs and estimates how long detected neighbors will be in proximity. When a neighbor is detected, Cost-Benefit Analysis determines the energy costs and benefits of sharing using an empirically generated time and power model as well as the proximity duration estimate. Sharing is initiated when Cost-Benefit Analysis determines that a neighbor will be in proximity long enough for the energy benefit of sharing to exceed any additional energy cost to collect new ground truth and train a shared classifier.

Sharing-Aware Classification. The Sharing-Aware Classification module provides a classification and training approach which adapts to the dynamics of available neighbors, utilizing neighbor and local resources which provide the best combination of high accuracy and energy savings. To efficiently classify activities on the phone in the presence of changing neighbor availability, we use Bagging [5], an ensemble classifier. Bagging allows a Remora BSN to quickly create an accurate classifier by combining weak classifiers from available local and neighbor sensors. At each aggregation interval, a decision classified by the ensemble is pushed to the application as well as pulled by other neighbors whose phones recently returned from a low power sleep state.

Collaborative Sensor Selection allows neighboring BSNs to find and utilize the shared resources which, in comparison with individual classification, provides higher accuracy and greater sensor energy savings. At the start of runtime, each BSN uses a classifier for individual classification based on available training data (Training Data Management) or a classifier previously trained through Collaborative Sensor Selection. When sharing is initiated, Collaborative Sensor Selection first labels training data with user input if no training data is present for the neighboring sensors. BSNs then collaborate to choose only the most capable sensors, creating a Bagging classifier that achieves high training accuracy for all neighbors yet uses fewer on-body sensors. Unused sensors are disabled during runtime to save energy.

To save phone energy, sharing BSNs duty cycle classifiers (Duty Cycling). One active neighbor makes a classification decision for the group at each aggregation interval. For each inactive BSN, as long as users are not interacting with the phone, the phone enters a low power sleep state to save energy. After sharing is initiated and training is complete, neighbors collaborate to establish a duty cycle order. Duty cycle times are short enough (5 min. in evaluation) so that changes in neighbor dynamics can be captured. Sleeping phones can be woken by the user to poll the active classifier for the current activity. Since duty cycling classifiers allows an active BSN to make decisions for all sleeping neighbors, all neighbors must be performing the same activity. However, neighbors performing different activities can still share sensors to increase accuracy. In our evaluation, subjects rarely perform different activities together for long durations, so our design focuses on both sensor and classifier sharing and we leave classifying different simultaneous activities to future work.

UI and Applications. To provide a user-friendly front end for Remora, we implement an Android app to allow easy configuration, ground truth labeling, and storage for sensor data and trained classifiers. In the application, the user configures and chooses which sensors are available for sharing, selects neighbors with whom to share, and starts and stops classification. Figure 7 depicts activity feedback during runtime, indicating shared neighbors, expected activity duration, and time each neighbor has spent sleeping or awake. Figure 8 shows, in realtime, which sensors on which nodes are used in shared classification. We also provide a dialog to prompt the user to label ground truth before training a classifier.

With a web-based application, depicted in Figure 9, users can visualize individual and shared activity inferences. Each BSN user is able to see how his or her activities and locations intertwine with friends and colleagues. For example, in the figure, two users arrive separately on a university campus, conduct a meeting, and then leave together.

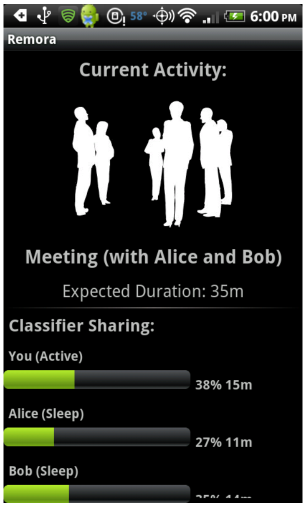

Figure 7: Activity status.

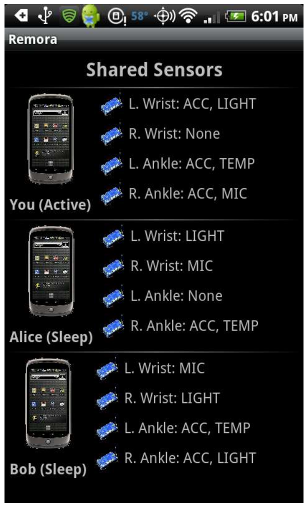

Figure 8: Shared resources.

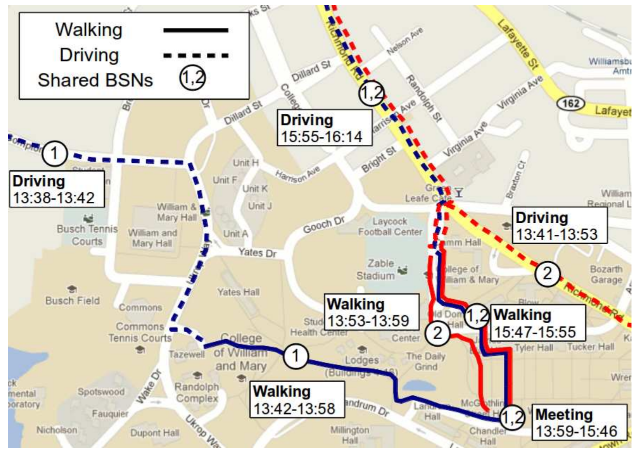

Figure 9: Map view of individual and shared activities.

5 Neighbor Management

Neighbor Management allows a BSN to detect and address the dynamics of neighbor availability, initiating resource sharing only when it is beneficial. First, Proximity Detection detects potential sharing neighbors and ensures that all are performing the same activity. Next, through the use of a shared online calendar, Duration Prediction estimates the proximity duration of detected neighbors. Third, Cost-Benefit Analysis integrates a time and energy model with the predicted duration to determine if sharing with a detected neighbor will provide an energy benefit.

5.1 Proximity Detection and Duration Prediction

In our implementation, each BSN detects neighbors through overhearing neighbors' on-body sensor data transmissions, however, detection can also be performed through phone-based Bluetooth discovery. Our neighbor detection scheme compensates for transmission delays or short periods of disconnection, such as when a neighbor briefly leaves a room and returns. We determine that a neighbor is in proximity if we overhear at least two packets from each neighboring mote within the last five aggregation intervals, or 50 seconds. If neighbors are detected, the local BSN exchanges the current activity with each neighbor to ensure all BSNs are performing the same activity. As mentioned previously, sharing neighbors must be performing the same activity so that neighbors can duty cycle classifiers. While duty cycling, one active classifier makes a single classification decision for all neighbors, allowing inactive neighbors to sleep.

Proximity Detection is also responsible for determining when to stop sharing. If a local BSN is currently sharing with a neighbor that is no longer detected, we inform the Sharing-Aware Classification module to stop sharing and revert to individual classification. Similarly, if the local BSN is sharing as the active classifier and the currently classified activity changes, it informs all neighbors to stop sharing as they wake up, for the activity change may not hold for the other sleeping BSNs.

If at least one detected neighbor is performing the same activity as the local BSN, we then estimate the proximity duration of the detected neighbors to determine if the neighbors will be present long enough for sharing to benefit. Before runtime, each BSN user defines events in a shared online calendar outlining his or her expected activities. The calendar is updated during runtime based on classified activities. When neighbors are detected during runtime, the current time and locally classified activity is compared against the current calendar activity for each BSN. If all BSNs have a calendar entry for the currently classified activity, the expected duration is computed as the earliest finishing time of any current calendar entry.

We use shared calendars to initiate the bootstrapping process when no large scale proximity information is available. The calendar approach allows new users to deploy Remora without collecting any proximity information a priori. Note that the shared calendars are only a rough approximation of expected activities; they are not meant to be exhaustive or completely accurate and are used only as a guide in determining if sharing will be beneficial. Similarly, previous work [21] investigates the use of shared calendars as sensors and concludes that with extra context information, such as classified activities, the calendar can be a powerful tool for monitoring human interaction. When an extensive history of interactions is available, we can use it to increase proximity prediction accuracy and provide a more holistic approach.

If at least one BSN does not have a calendar entry or is not performing the currently classified activity, the local BSN prompts the user via a dialog to ask if he or she expects all neighbors to be together long enough to share. Based on the cost model, in the sharing dialog, Remora will supply the user with the exact time length required for sharing to benefit. This manual sharing decision will bypass Cost-Benefit Analysis and, based on the yes/no decision of the user, will either directly initiate sharing or exclude the newly detected neighbor from shared classification.

5.2 Cost-Benefit Analysis

After a neighbor is detected and using the estimated neighbor proximity duration, we use a cost and benefit model to determine if sharing will result in energy savings. In our evaluation, most on-body motes ran without battery replacement during the two week experiment while phones had an approximate battery life of 10 hours for individual classification. Since phone battery life is the limiting factor in BSN lifetime, we focus on improving it through classifier duty cycling and Cost-Benefit Analysis. We define the costs in terms of time and energy to train a new classifier if a classifier has not already been trained using the current neighbors' resources. We define the benefits in terms of energy saved during low power duty cycling among neighbors compared with always-on individual classification. In Section 6.4, we empirically determine the cost model parameters and extend the model to our specific sharing design and BSN hardware.

We first define the phone energy required to collect new ground truth and train a new classifier while still performing individual classification, E_{tr}:

$$E_{tr} = [(T_{GT} + T_{tr})(P_{class} + P_{sensor})] + (T_{GT} \cdot P_{GT}) + (T_{tr} \cdot P_{tr}) \quad (1)$$

In Equation 1, $T_{GT} + T_{tr}$ refers to the total time needed to collect ground truth and train a new classifier. P_{class} refers to the base power required to perform individual classification, while P_{sensor} is the power consumed by sensors on the phone, including GPS and radio connectivity. Additionally, P_{GT} and P_{tr} refer to the additional power needed to collect ground truth and train a new classifier, respectively.

Next, we define the phone energy required to perform shared classification with neighboring resources, E_{share}:

$$E_{share} = [T_{prox} - (T_{GT} + T_{tr})] \cdot \left[\frac{1}{b}(P_{class} + P_{sensor}) + \left(1 - \frac{1}{b}\right) P_{sleep}\right] \tag{2}$$

In Equation 2, T_{prox} is the predicted proximity duration, with $T_{prox} - (T_{GT} + T_{tr})$ representing the estimated time spent in shared classification after subtracting the time needed to collect ground truth and train a classifier $T_{GT} + T_{tr}$. Also, b is the number of sharing BSNs, and P_{sleep} is the power consumed by a BSN while it is in a low power sleep state. Note that each BSN spends an equal amount of time classifying; in this chapter, we ensure energy use fairness and leave a lifetime optimization scheme to future work.

Third, we define the phone energy required to classify as an individual BSN, E_{ind}:

$$E_{ind} = T_{prox} \cdot (P_{class} + P_{sensor}) \tag{3}$$

Equation 3 predicts the energy consumed by a BSN if it spends the expected proximity duration in individual classification instead of shared classification.

Lastly, using the above equations, we share when the energy to train a classifier and perform shared classification is less than performing individual classification for the expected proximity duration:

$$T_{prox} > (T_{tr} + T_{GT}) \text{ and } E_{tr} + E_{share} < E_{ind} \tag{4}$$

In Equation 4, we also ensure that the predicted proximity duration is longer than the time needed to collect new ground truth and train a classifier. If a neighbor is detected and the above conditions hold, sharing is initiated by notifying Sharing-Aware Classification, which we describe next.

6 Sharing-Aware Classification

In this section, we first explain details on our classifier as well as what happens when a neighbor is detected and Cost-Benefit Analysis initiates sharing. Second, if sharing is initiated and a new classifier is needed, we then explain how neighboring BSNs train new classifiers by collaborating to determine a set of sensors to use for shared classification. Third, we explain how BSNs share classifiers and duty cycle them to significantly increase phone battery life. Lastly, through experimentation, we derive a cost model specific to our shared sensor selection approach and hardware.

We use an ensemble technique, Bootstrap Aggregating (Bagging) [5] for activity classification. Bagging is a lightweight approach appropriate for phones that makes classification decisions based on the majority vote of an ensemble of weak classifiers. In our Bagging classifiers, each weak classifier is a Naive Bayes classifier trained from the training data of a single sensor [16]. Other sharing approaches use more complex techniques, such as GMMs trained offline [24] or Boosting [18].

Bagging is exceptionally useful for addressing the dynamics of available neighbors: in addition to its quick training time and unlike many other classification methods, we can efficiently combine two existing Bagging classifiers into one large classifier, which we exploit during Collaborative Sensor Selection. Specifically, during Collaborative Sensor Selection, BSNs first train Bagging classifiers for individual *sensor classifiers* by training an ensemble of weak classifiers from the training data of a single sensor. Then, BSNs choose the best sensor classifiers and integrate them into a single *composite classifier* (*classifier* in previous sections) which is used to make runtime decisions for either an individual BSN or both local and neighbor BSNs.

Runtime and Sharing Initialization. At the start of runtime, each BSN either trains a new composite classifier for individual classification or loads a previously trained classifier from flash storage. Initial training is performed using Collaborative Sensor Selection but using local sensors only. During runtime, following detection of a neighbor, Cost-Benefit Analysis initiates sharing when it determines that sharing is beneficial. In a radio packet, a BSN announces an intent to share to its neighbors. Neighboring BSNs receive the packet, perform their own Cost-Benefit Analysis, and return a reply to indicate if they will participate. If at least two neighbors agree to share, either a previously trained classifier is reused or more training data is collected and a new classifier is trained through Collaborative Sensor Selection.

Classifier Reuse. Composite classifiers are stored in flash memory for reuse. If a combination of neighbors meet, train classifiers, perform shared classification, and later meet again while performing the same activity, the previously trained classifiers are loaded from flash and used again. This saves significant sharing training and energy costs and allows sharing for short periods of time (5-10 min. in evaluation) with the same combination of neighbors and activities.

Ground Truth and Sensor Classifier Training. If neighboring BSNs do not have previously trained composite classifiers or the current activity is different than what these neighbors last performed, the neighbors collect new training data for the current activity. Training data is labeled with the current activity by the user and is collected for all local sensors and public neighbor sensors. When enough data is collected (5min in evaluation), each neighbor trains a sensor classifier for each available sensor and broadcasts its intent to start Collaborative Sensor Selection.

6.1 Sensor Selection Motivation

Before we explain our Collaborative Sensor Selection algorithm, we first provide intuition for its design using data from the motivation experiment. The challenge is to identify properties of both local and neighbor sensors such that, based on these properties, all neighbors can choose sensors to create composite classifiers that are accurate for all neighbors. Previous work [34] demonstrates that in order for ensemble classifiers, such as Bagging, to be trained successfully, two properties must hold: 1) the individual weak classifiers must be accurate, and 2) weak classifiers must produce diverse classification results. We analyze these properties as they pertain to choosing sensor classifiers and adding them to a composite classifier. We conclude that choosing sensors based on individual accuracy (Figure 10) and decision correlation (Figure 11) will create an accurate composite classifier with a small number of capable sensors.

We first show in Figure 10 that we can discriminate best between activities by choosing sensors with the

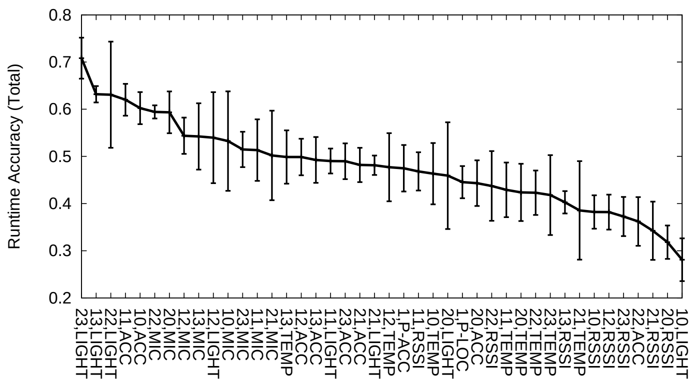

Figure 10: Subject 1: Individual sensor accuracy.

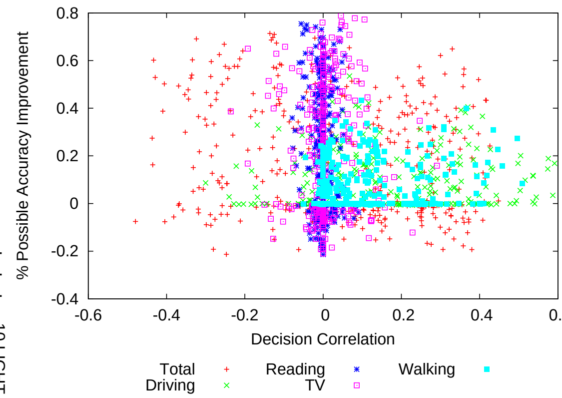

Figure 11: Decision correlation and accuracy improvement.

best individual accuracies. Using the motivation data, we train sensor classifiers for each sensor available to Subject 1, including publicly shared sensors from Subject 2. Each sensor classifier comprises 30 weak classifiers and we plot the average runtime accuracy of 10 classifier trainings on the y-axis. Each sensor is labeled by its on-body node ID and modality, where IDs starting with 1 are sensors from Subject 1 and IDs starting with 2 are sensors from Subject 2. Ranked by individual sensor accuracy, we can see that sensors on both Subject 1 and 2 exhibit the highest accuracy, indicating that sharing gives Subject 1 more accurate sensors from which to choose. Furthermore, we can see that the light, accelerometer, and microphone sensors perform the best, for they are able to distinguish between indoor and outdoor light levels, limb movements (walking), and sounds, such as from a TV.

Next, in Figure 11, we illustrate how to find sensors with complimentary classification capability, locating a combination of sensors that is not only accurate but contains few sensing redundancies. In the figure, we show that when we add a sensor classifier to an existing composite classifier consisting of many other sensor classifiers, the greatest accuracy increase is observed when the sensor and composite classifier have uncorrelated classification decisions. The figure shows the decision correlation between a composite classifier generated from the data of 10 random sensors and a classifier generated from a single sensor not used by the randomly created composite classifier. We compute the accuracy change of combining the sensor classifier with the composite classifier. To compute decision correlation, each correct runtime decision is recorded as 1 and each incorrect decision is recorded as 0. From the figure, which contains 340 random composite classifiers, we can determine that by choosing sensors with decision correlation close to zero, we will ensure that each sensor we choose will produce a meaningful contribution towards an accurate composite classifier.

6.2 Collaborative Sensor Selection

Based on the motivation experiment, we provide a collaborative approach to training a composite classifier for shared classification. This approach is also used by a single BSN to train a composite classifier for

Algorithm 1 Collaborative Sensor Selection

Input Classifiers for local sensors and public neighbors S, classifiers for private sensors P
Output Composite classifier for local BSN C_i

// BSN initiates sharing
function START(S)
  $C_i = \emptyset$
  Send to neighbors: $\forall s_j \in S, \text{acc}_i(s_j)$
// Receive sensor classifier accuracy or correlation values from all neighbors; choose a sensor classifier based on weights and neighbor composite ensembles C
event CHOOSESENSOR(S,C)
  $\forall s_j \in S$ compute weight w_j with Equation 6
  // First, add private sensors with highest weight
  repeat
    Get W, sensor classifiers with highest weight
    for all $w_i \in W \cap P$ **do**
      ADDSENSOR(w_i,S,C_i)
      $W = W \setminus w_i$
  until $|W| > 0$
  // Next, choose public sensor
  // Only one sensor $w_0 \in W$ has highest weight, add it
  if $|W| = 1$ **then**
    ADDSENSOR(w_0,S,C_i)
  // Several sensors with same weight; Local BSN wins tiebreaker
  else if local BSN ID is the lowest of all neighbors **then**
    Choose random classifier $w_r \in W$
    ADDSENSOR(w_r,S,C_i)
    Notify neighbors of choice w_r
  // Several sensors with same weight; Local BSN loses tiebreaker
  else
    Wait for tiebreaker BSN to choose random classifier $w_r \in W$
    ADDSENSOR(w_r,S,C_i)
  if $\text{acc}_i(C_i) < 1$ and $|S| > 0$ **then**
    Send to neighbors: $\forall s_i \in S, r_{C_i,s_j}$
// Add sensor s_j to composite classifier C_i if s_j improves accuracy
function ADDSENSOR(s_j,S,C_i)
  if $\text{acc}_i(C_i \bigcup s_j) > \text{acc}_i(C_i)$ **then**
    $C_i = C_i \bigcup s_i$
  $S = S \setminus s_i$

individual classification when no neighbors are present. The main idea is for neighboring BSNs to iteratively add one sensor classifier at a time to their respective composite classifiers. At each iteration, all neighbors agree on a sensor classifier to add to their composite classifiers based on sensor classifier accuracy and decision correlation. A neighbor participates in sensor selection until it either maximizes training accuracy or all available sensors are added to its composite classifier. Using Algorithm 1, Collaborative Sensor Selection is explained in detail:

Each BSN i in the set of neighbors B first initializes a null composite classifier C_i (line 1 in Algorithm 1). Then, each BSN transmits to its neighbors the accuracies of each trained sensor classifier $\mathrm{acc}(s_j)$. Then, after all accuracy values are exchanged, each BSN ranks each sensor classifier $s_j \in S$ by the following weight, $w(C, s_j)$, in Equation 6.

$$w_i(C_i, s_j) = \alpha \cdot \mathrm{acc}_i(s_j) + (1-\alpha)\left(1 - |r_{C_i,s_j}|\right) \quad (5)$$

$$w(C, s_j) = \frac{1}{B}\sum_{i=1}^{B} w_i(C_i, s_j) \quad (6)$$

In Equation 6, each sensor classifier for BSN i and sensor s_j is weighted by its accuracy, decision correlation r with the current composite classifier C_i, and the number of BSNs B. α provides a weight to emphasize either accuracy or decision correlation when weighting (we use $\alpha = 0.5$). At first, when the composite classifier is null, each sensor classifier is weighted only by accuracy. Also, if the classifier is for a private sensor, no neighbors have accuracy or decision correlation information for the classifier, so the weight is computed using Equation 5 only.

After computing weights (line 5 in Algorithm 1), each BSN then chooses the sensor classifier with the highest weight. Since each BSN computes the same weight values independently, each BSN will choose the same sensor classifier. However, if there are multiple sensor classifiers with the same weight, the BSN with the lowest BSN ID value chooses a sensor and broadcasts its choice to neighboring BSNs (lines 14-17). If a private sensor (sensor a user does not share with neighbors) classifier has the highest weight, it is chosen along with one other public sensor classifier to ensure all BSNs choose the same classifier (lines 6-11). Once a sensor classifier is chosen, it is only added to the composite classifier if it increases the composite classifier accuracy.

After a sensor classifier is chosen, a BSN stops sensor selection if there are no more remaining sensors to choose from or the BSN has achieved perfect training accuracy. While adding more sensor classifiers to a composite classifier with perfect training accuracy may improve runtime accuracy [5], we focus on reducing training costs and stop when we achieve perfect training accuracy. Remaining BSNs then compute decision correlation r between the ensemble classifier and each remaining sensor classifiers and broadcast the correlation values. Another sensor classifier is then chosen in the manner above and the process repeats.

6.3 Classifier Sharing and Duty Cycling

After sharing is initiated by Cost-Benefit Analysis and a classifier is trained or reused, all neighbors collaborate to define a duty cycle order where at any given time period, only one phone is active and classifying activities. Neighbors exchange a random integer concatenated by a BSN ID integer, with the duty cycle order following the ascending order of the generated values. While we can optimize duty cycle periods for each phone to maximize lifetime, we leave that to future work and use a round robin duty cycling scheme to ensure fairness in energy consumption. We choose a duty cycle of appropriate length (5 min. in evaluation)

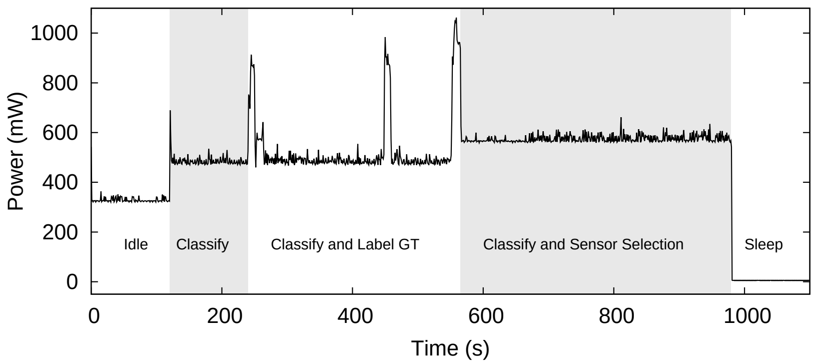

Figure 12: Remora power profile.

Figure 13: Power Meter Setup.

so that sleeping neighbors are able to timely wake up and detect changes in available neighbors, especially if the active classifier departs.

After a new classifier is trained, it is saved to flash storage to facilitate both reuse and duty cycling, for when a phone goes to sleep, memory may be purged. Upon waking up as the classifying neighbor, a BSN loads the classifiers back into memory including the one for the current neighbor combination. Saving and loading is nearly instantaneous, requiring little overhead. Upon waking up, if at least one sharing neighbor has departed, the BSN reverts to individual classification and notifies all remaining sharing neighbors.

6.4 Empirical Sharing Cost Model

Using Collaborative Sensor Selection, we empirically define a sharing cost model specific to our system which extends the general model defined in Section 5.2. Using an HTC Hero smartphone and four on-body sensors as described in Section 3.2.1, we perform time and power benchmarks of our Remora implementation. We use the benchmarks to define the training time, training power, and minimum proximity duration needed for sharing to provide an energy benefit.

We measure power consumption by connecting an HTC Hero smartphone-based BSN running Remora to a Monsoon Technologies Power Meter, demonstrating that we achieve massive phone energy savings by duty cycling classifiers. In Figure 12, we create a power profile of each system behavior with WiFi and GPS disabled. The power meter setup is depicted in Figure 13. At time 0, we show the idle power consumption of the phone with CPU active and start individual classification at 150s. Touching the screen to start classification briefly consumes about 700mW. Classification consumes an additional 150mW of power until a neighbor is encountered at 230s and sharing starts. The user labels ground truth three times (screen usage spikes) over five minutes until enough new training data is collected to train a shared composite classifier at 570s. Training while performing individual classification lasts more than 5 min. and consumes about 200mW more than classification alone, and upon completion of training, the phone immediately goes to sleep with the neighbor actively classifying for both BSNs. A sleeping phone consumes fewer than 10mW

Base Power		Additional Power			
Classify	Sleep	Ground Truth	Train	GPS	WiFi
486.43	5.25	+47.62	+88.51	+194.0	+31.31

Table 2: Remora Power Consumption (mW).

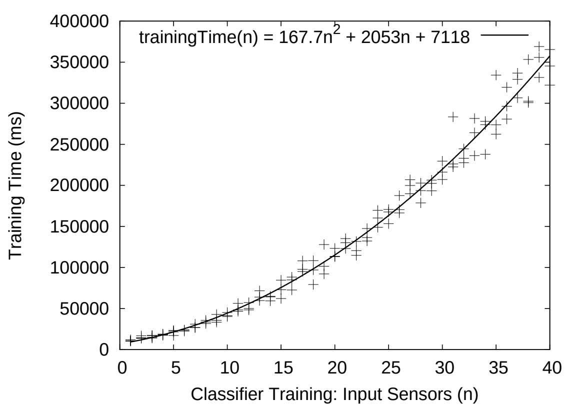

Figure 14: Phone training time: sensor classifiers and Collaborative Sensor Selection

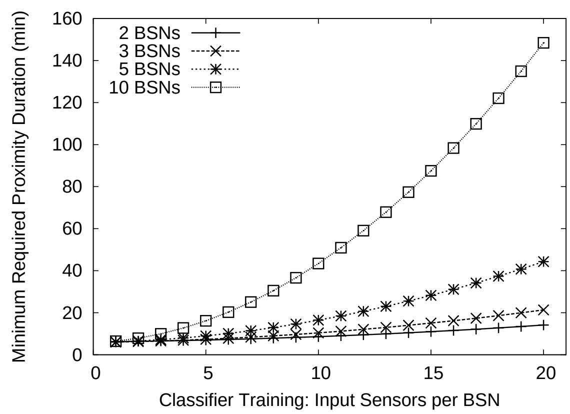

Figure 15: Proximity duration needed for sharing to benefit.

of power, which is much less than the nearly 500mW required for classification.

The average power consumption for each state is provided in Table 2: base power for sleep or classification (consumption is the same for individual and shared classification), and additional power required for collecting ground truth (screen use), training a composite classifier through Collaborative Sensor Selection, and GPS and WiFi use.

On the phone, we also measure the training time required to train new sensor classifiers and perform Collaborative Sensor Selection. We vary the number of sensors given to Collaborative Sensor Selection and plot the results in Figure 14. Our training algorithm has polynomial time complexity: with respect to the number of sensors n, training sensor classifiers is $O(n)$. Collaborative Sensor Selection is $O(n^2)$: each time a sensor classifier is added to the composite classifier, decision correlation is computed for each unchosen sensor classifier. From the figure and training time data points, we perform polynomial curve fitting to determine the training time $T_{tr}(n)$ in Equation 7 for use in our cost model, where n is the number of sensors.

$$T_{tr}(n) = 167.7n^2 + 2053n + 7118 \tag{7}$$

The polynomial time complexity indicates that training times are significantly faster when providing fewer sensors as training input. When training an individual composite classifier, we give all local sensors as input (20 on-body and 2 phone), which requires about two minutes of training. However, if a BSN uses all local and public neighbor sensors when training a shared composite classifier (42 for 2 BSNs), training can take more than 7 minutes. With more neighbors, training can take even longer. Instead, when neighbors are present, we ensure that only sensors chosen by each BSN for individual classification are given as input for shared classification. This ensures that shared training overhead is reduced. Furthermore, Figure 14 and Equation 7 illustrate that sharing with a fewer number of neighbors will provide a greater energy benefit and also allow sharing over shorter durations due to the lower training costs.

In Figure 15, we provide more evidence that sharing with a small number of neighbors is most beneficial. Using Equation 7, our power consumption results, and our cost model from Section 5.2, we compute the minimum proximity duration needed for sharing to provide an energy benefit. In our evaluation, each BSN uses 10 sensors on average for individual classification, which is the input size when building a shared classifier. This indicates that the minimum proximity duration is under 20 minutes for up to 5 neighbors, which is reasonable under most circumstances. However, with 10 neighbors, training more than doubles to over 40 minutes, which is impractical for many dynamic scenarios.

7 Evaluation

We evaluate our Remora shared activity classification system using the configuration and data collection methods in Section 3.2.1. We combine the motivation experimental data with two weeks of new data. In the evaluation, six subjects, including the first two subjects from the motivation, perform a combination of the following activities: riding a bus, riding in or driving a car (driving), meeting, reading, running, watching TV, walking, and working at a desk. The subjects we use meet our design target of sharing with neighbors with strong interpersonal ties: all are graduate students or family members who spend a significant amount of time together. Each subject has an initial training set of 30 observations per activity (5 min.) and trains an individual classifier at the start of runtime, with each sensor classifier trained using 30 weak classifiers. We evaluate three scenarios using the same data and activity ground truth: individual classification only, sensor sharing only, and sensor and classifier sharing. In Section 7.1, we first look at sharing opportunities. Then, we demonstrate sharing accuracy improvements in Section 7.2. In Section 7.3, we highlight the benefits of sharing as well as significantly improved battery life in Section 7.4. Lastly, in Section, 7.5, we highlight Remora CPU and memory usage.

7.1 Proximity and Sharing

In this section, we show that subjects are in proximity to each other for a significant portion of time, giving ample sharing opportunity. We also show that Remora uses 96% of the total proximity duration to share sensors and classifiers.

First, in Table 3, we show the total time at least one neighbor is in proximity for each subject (Prox. Duration). While the average percentage of time sharing is greater than the MIT dataset [10], the subjects we evaluate are a close knit group of students and family. Furthermore, since the focus of the evaluation is on sharing performance, most of the subjects do not wear the BSN for all individual activities. However, Subject 1 performs a large number of individual activities and is more representative of the neighbor proximity duration that would be expected from such a group.

		Sharing	
Subj.	Prox. Duration (%Total)	Reuse (% Prox. Duration)	No Reuse (% Prox. Duration)
1	22h36m (39%)	21h30m (95%)	17h42m (78%)
2	04h06m (82%)	03h54m (96%)	03h34m (87%)
3	10h30m (65%)	10h06m (96%)	08h39m (82%)
4	09h18m (64%)	08h48m (95%)	07h15m (78%)
5	06h06m (63%)	05h54m (96%)	04h39m (76%)
6	07h06m (68%)	06h42m (95%)	04h57m (70%)

Table 3: Total Proximity and Sharing Duration.

Next, Table 3 demonstrates when we perform sharing with classifier reuse (Reuse) that an average of 96% of the total proximity time is utilized. The remaining 4% difference is due to Cost-Benefit Analysis rejection due to short proximity durations or different simultaneous activities. The time difference also includes sharing overhead: time to collect ground truth and perform sensor selection. With classifiers reused for multiple encounters with the same neighbor combination, Remora can quickly adapt to share with available neighbors. Also, classifier reuse accounts for 90% of sharing encounters among all BSNs. Without classifier reuse (No Reuse), however, sharing overhead is much higher, with an average of 78% of the total proximity time utilized.

7.2 Accuracy Improvement

In Figure 16, we highlight overall accuracy performance for each BSN for individual classification, sharing sensors, and sharing both sensors and classifiers, analyzing only the periods where sharing is possible to make a fair comparison. From the figure, all subjects except for Subject 2 receive an accuracy benefit from sharing sensors and classifiers, with Subjects 1, 4, and 6, receiving the greatest accuracy gains of over 20% points, or nearly 30% over individual classification. The figure also demonstrates that duty cycling classifiers among neighboring BSNs has no impact on accuracy. The 5 min. duty cycle period is short enough for each BSN to capture changes in its own activities as well as neighbor departures and stop sharing if such a change is detected.

From Figure 16, Subject 2 has the highest accuracy of all subjects, which is because Subject 2 does not perform as many activities as the other subjects. Conversely, Subject 1 performs a multitude of activities and has the lowest individual accuracy. In Figure 17, we can see that Subject 1 exhibits confusion between the meeting and working activities in addition to the reading and watching TV activities presented in Section 3.2.2. The additional sensors provided by neighbors are able to overcome these challenging activities. Similar confusion between meeting and working can also be witnessed in Figure 18 and 19 for Subjects 4 and 6, respectively, where total accuracy is also significantly improved by sharing. Lastly, we note that Collaborative Sensor Selection ensures that accuracy is only improved by sharing; high individual accuracy is maintained during sharing while low individual accuracy is improved.

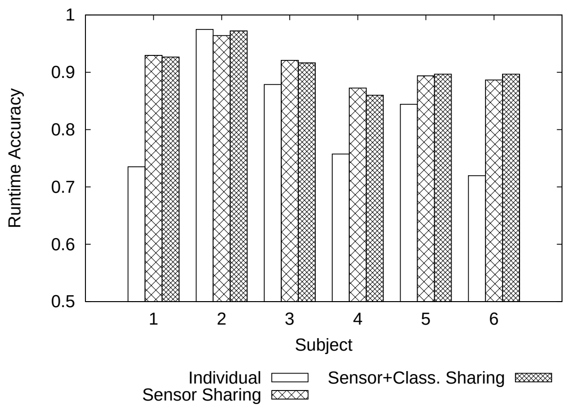

Figure 16: Runtime accuracy for each subject.

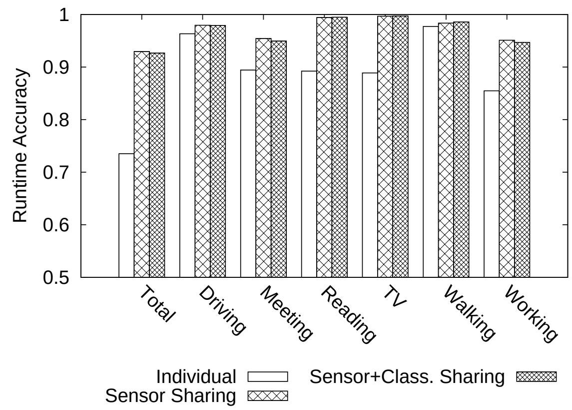

Figure 17: Subject 1 accuracy.

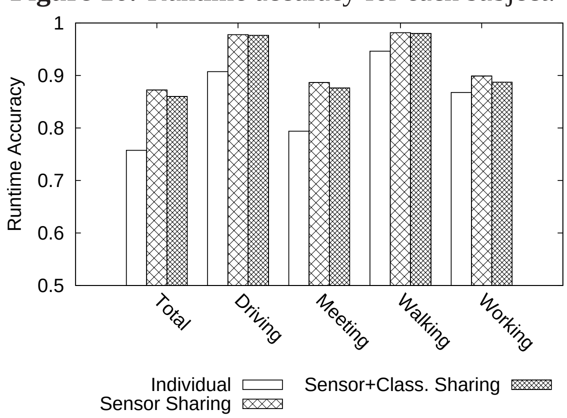

Figure 18: Subject 4 accuracy.

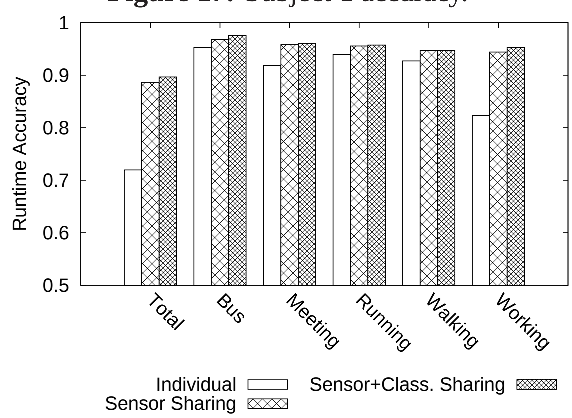

Figure 19: Subject 6 accuracy.

In Figure 20, we show the accuracy improvement of sharing sensors and classifiers over individual classification. The figure compares BSN performance for each activity and neighbor combination. The figure demonstrates that the neighbors with the worst individual accuracies gain the most accuracy benefit by sharing. For example, Subject 1 gains over 30% points accuracy in the meeting activity when sharing with Subjects 5 and 6, which is also reflected in the accuracy improvement for all shared meeting activities in Figure 17. Similar improvements are also witnessed for Subjects 4, 5, and 6. While Subjects 2 and 3 do not have as high accuracy gains by sharing, these two subjects have high individual accuracy, and thus the marginal improvement is lower.

7.3 Sharing Costs and Benefits

We evaluate Neighbor Management and Cost-Benefit Analysis to show that we can accurately determine when and how much sharing will benefit. In Figure 21, we plot the sharing decision accuracy when using

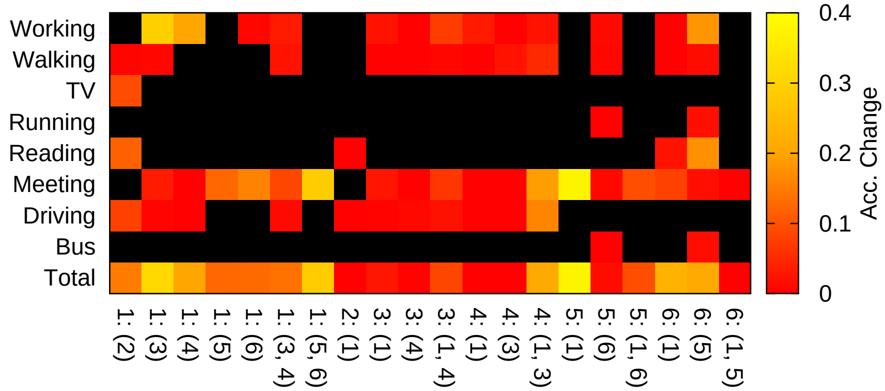

Figure 20: Accuracy improvement over individual classification for each BSN, activity, and neighbor combination. For example, the first column is the accuracy increase for BSN 1 when sharing with BSN 2. Black indicates no data.

the shared calendar or manually asking the user for proximity duration when the calendar has no result. Accuracy is computed as the number of true positive sharing decisions (sharing initiated and lasted long enough to achieve an energy benefit) plus true negative sharing decisions (sharing not initiated and proximity duration was too short for sharing to benefit) compared with the total number of sharing decisions. Both the calendar and manual prediction have similar performance and achieved over 90% accuracy for all subjects. We also show that the calendar is used to make between 50% and 70% of all sharing decisions, which significantly reduces user invasiveness since Remora does not have to ask the user to estimate proximity duration.

Next, we illustrate that for every subject, the benefits of sharing outweigh the costs by two orders of magnitude. First, from our cost model, in Figure 22, we compute the ratio between the energy savings gained through duty cycling classifiers and the additional energy costs required to collect ground truth and train shared classifiers. The average net energy savings for the shared periods is about 400% for all subjects. During the experiment, each subject was in proximity with no more than two neighbors simultaneously, however, we see that the bulk of the energy savings comes from sharing with one neighbor. The marginal benefit of sharing with an additional neighbor is negative with the exception of Subject 4. This demonstrates that sharing with a small number of BSNs achieves high accuracy with low cost; sharing with a large number of BSNs is impractical and will be rejected by our Cost-Benefit analysis.

7.4 Energy Savings

To further highlight classifier sharing as well as the ability of Remora to adapt classification to available neighbors, Figure 23 presents a timeline of a shared classification period between Subject 1 and 2. In the figure, we present power usage as well as mark correct and incorrect activity classifications for each classification decision. Subject 1 and 2 perform individual activities until 7 minutes, where Subject 2 enters

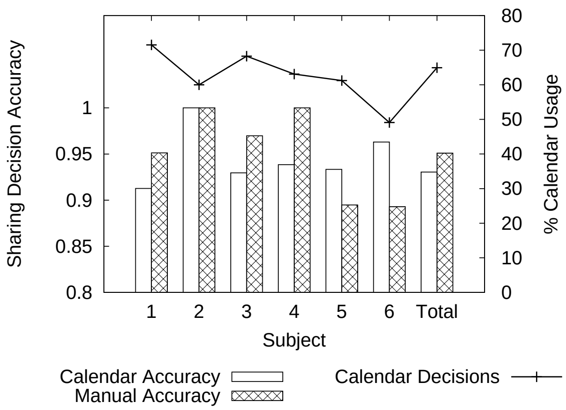

Figure 21: Sharing Decisions.

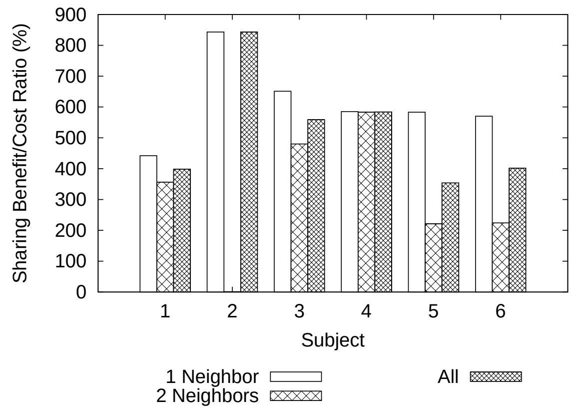

Figure 22: Energy benefit to cost ratio.

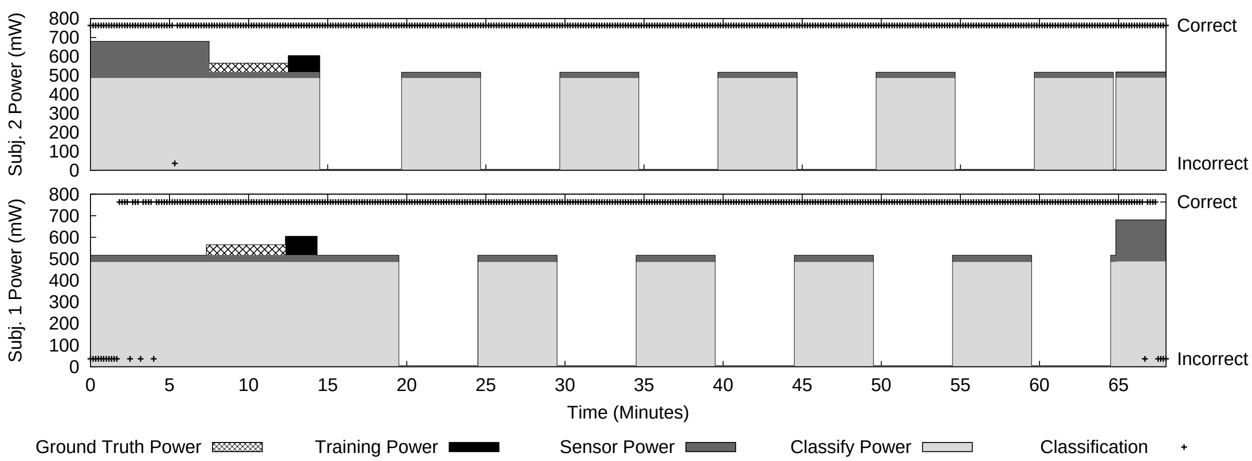

Figure 23: Sharing timeline: energy and accuracy profile.

a building after being outside and meets Subject 1 (note that the GPS is active and consumes more sensor energy). After Subject 1 and 2 meet, Proximity Detection and Duration Prediction estimates the length of the proximity period while Cost-Benefit Analysis quickly determines that sharing will provide an energy benefit, initiating ground truth labeling and classifier training. During the individual periods, Subject 1 makes many misclassifications but after ground truth is logged and a new classifier trained, Subject 1 exhibits high accuracy with no misclassifications. After training is complete at 14 minutes, both BSNs trade off as the active classifier, alternately going to sleep until Subject 1 leaves and goes outside, returning to individual classification with several misclassifications.

We now demonstrate that by sharing and duty cycling classifiers, we can increase phone battery life up to 65%. We also show that we can save mote energy while sharing sensors to reduce the number of sensors needed by nearly 50%. To compute battery life for each BSN, we determine as a percentage of the total running time: time spent during active classification and sleep, phone sensor use, training time and ground truth labeling. Combined with power consumption in Table 2 and a 1500mAh battery per phone, we present results in Figure 24. For each BSN, individual classification yields about 10 hours of battery life. However, with duty cycling through classifier sharing, battery life can be extended from 13 hours for Subject 1 to almost 17 hours for Subject 4. This represents an increase ranging from 25% to over 65%.

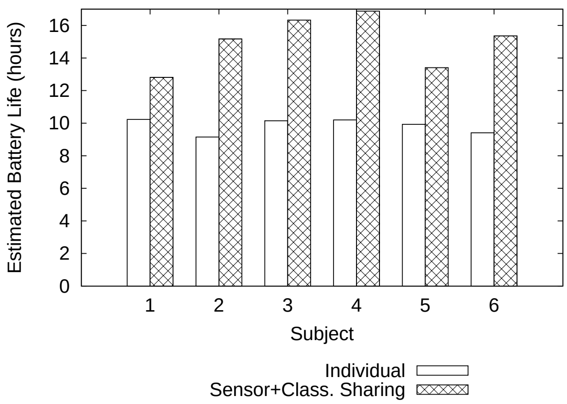

Figure 24: Battery Life.

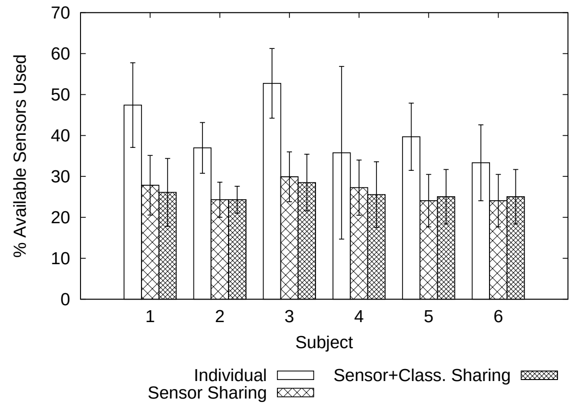

Figure 25: Percentage of available sensors chosen.

While accuracy is increased over individual classification, sharing sensors can also reduce the total number of sensors used by all neighbors. In Figure 25, we plot the average percentage of available sensors chosen during Collaborative Sensor Selection. The figure shows that between 10% and 20% points fewer sensors are used while sharing sensors or classifiers compared with individual classification. This is because Collaborative Sensor Selection is able to identify and use only the sensors that provide the most accuracy benefit. When neighbors are present, there are more sensors to choose from and more sensors that provide a large contribution towards providing high classification accuracy.

7.5 CPU and Memory Usage

Mode	CPU	Memory
Classify	23%	8.1MB
Classify + Ground Truth	23%	8.1MB
Classify + Train	96%	9.2MB
Sleep	0%	8.1MB

Table 4: Remora CPU and memory benchmarks.

In Table 4, we illustrate CPU and memory usage as determined by the Android SDK toolkit. Since classifiers for the same neighbor combination are reused once trained, most of the time Remora incurs about

23% CPU overhead during classification. While training, Remora maximizes CPU availability, however, this lasts no more than 10-15 minutes depending on the number of sensors and BSNs available. Furthermore, because Remora training and classification is run as a background process, the CPU scheduler gives priority to other applications running in the foreground so that the phone remains responsive. Lastly, memory usage remains relatively constant during classification as well as during training, ranging from about 8-9MB.

8 Conclusion and Future Work

In this paper, we propose Remora, a smartphone-based body sensor network system for activity classification which exploits physical proximity of neighboring BSNs to provide increased accuracy and energy savings. First, through a time and energy cost-benefit analysis, we determine when sharing provides an energy benefit. Second, our Collaborative Sensor Selection approach efficiently chooses a small number of sensors that provides high accuracy for all shared BSNs. Third, classifiers among sharing neighbors are duty cycled to provide a significant boost in phone battery life. Our multi-week evaluation demonstrates an accuracy improvement of up to 30% and battery life improvement of over 65%. In future work, we propose to maximize phone battery life with an optimal duty cycling scheme while also considering energy use of other applications. We also intend to extend sensor data and classifier sharing to the cloud to further improve accuracy, extend sharing support to social networking applications, and investigate duty cycling classifiers when neighbors perform different activities.

References

[1] F. Albinali, S. Intille, W. Haskell, and M. Rosenberger. Using Wearable Activity Type Detection to Improve Physical Activity Energy Expenditure Estimation. In *UbiComp '10*, pages 311–320. ACM, 2010.

[2] M. Baechlin, K. Foerster, and G. Troester. SwimMaster: A Wearable Assistant for Swimmer. In *UbiComp '09*, pages 215–224. ACM, 2009.

[3] R. Bajcsy, A. Giani, C. Tomlin, A. Borri, and M. D. Benedetto. Classification of Physical Interactions Between Two Subjects. In *BSN '09*, pages 187–192. IEEE, 2009.

[4] X. Bao and R. Choudhury. MoVi: Mobile Phone based Vidoe Highlights via Collaborative Sensing. In *MobiSys '10*, pages 357–370. ACM, 2010.

[5] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[6] O. Chipara, C. Lu, T. Bailey, and G.-C. Roman. Reliable Clinical Monitoring using Wireless Sensor Networks: Experience in a Step-down Hospital Unit. In *SenSys '10*, pages 155–168. ACM, 2010.

[7] D. Choujaa and N. Dulay. Predicting Human Behaviour from Selected Mobile Phone Data Points. In *UbiComp '10*, pages 105–108. ACM, 2010.

[8] D. Chu, N. Lane, T. Lai, C. Pang, X. Meng, Q. Guo, F. Li, and F. Zhao. Balancing Energy, Latency, and Accuracy for Mobile Sensor Data Classification. In *SenSys '11*. ACM, 2011.

[9] T. Das, P. Mohan, V. Padmanabhan, R. Ramjee, and A. Sharma. PRISM: Platform for Remote Sensing Using Smartphones. In *MobiSys '10*, pages 63–76. ACM, 2010.

[10] N. Eagle, A. Pentland, and D. Lazer. Inferring Social Network Structure using Mobile Phone Data. *PNAS*, 106(36), 2009.

[11] M. Faulkner, M. Olson, R. Chandy, J. Krause, K. Chandy, and A. Krause. The Next Big One: Detecting Earthquakes and other Rare Events from Community-based Sensors. In *IPSN '11*, pages 13–24. ACM/IEEE, 2011.

[12] S. Gaonkar, J. Li, R. Choudhury, L. Cox, and A. Schmidt. Micro-Blog: Sharing and Querying Content Through Mobile Phones and Social Participation. In *MobiSys '08*, pages 174–186. ACM, 2008.

[13] T. Hossmann, T. Spyropoulos, and F. Legendre. Putting Contacts into Context: Mobility Modeling beyond Inter-Contact Times. In *MobiHoc '11*. ACM, 2011.

[14] A. Kalamandeen, A. Scannell, E. de Lara, A. Sheth, and A. LaMarca. Ensemble: Cooperative Proximity-based Authentication. In *MobiSys '10*, pages 331–344. ACM, 2010.

[15] T. Karagiannis, J. L. Boudec, and M. Vojnovic. Power Law and Exponential Decay of Intercontact Times between Mobile Devices. *IEEE TMC*, 9(10), 2010.

[16] M. Keally, G. Zhou, G. Xing, J. Wu, and A. Pyles. PBN: Towards Practical Activity Recognition Using Smartphone-Based Body Sensor Networks. In *SenSys '11*, pages 246–259. ACM, 2011.

[17] E. Koukoumidis, L. Peh, and M. Martonosi. SignalGuru: Leveraging Mobile Phones for Collaborative Traffic Signal Schedule Advisory. In *MobiSys '11*. ACM, 2011.

[18] N. Lane, Y. Xu, H. Lu, S. Hu, T. Choudhury, A. Campbell, and F. Zhao. Enabling Large-scale Human Activity Inference on Smartphones using Community Similarity Networks (CSN). In *UbiComp '11*, pages 355–364. ACM, 2011.

[19] K. Li, T. Sohn, S. Huang, and W. Griswold. PeopleTones: A System for the Detection and Notification of Buddy Proximity on Mobile Phones. In *MobiSys '08*, pages 160–173. ACM, 2008.

[20] K. Lorincz, B. Chen, G. Challen, A. Chowdhury, S. Patel, P. Bonato, and M. Welsh. Mercury: A Wearable Sensor Network Platform for High-Fidelity Motion Analysis. In *SenSys '09*, pages 183–196. ACM, 2009.

[21] T. Lovett, E. O'Neill, J. Irwin, and D. Pollington. The Calendar as a Sensor: Analysis and Improvement Using Data Fusion with Social Networks and Location. In *UbiComp '10*, pages 3–12. ACM, 2010.

[22] H. Lu, W. Pan, N. Lane, T. Choudhury, and A. Campbell. SoundSense: Scalable Sound Sensing for People-Centric Sensing Applications on Mobile Phones. In *MobiSys '09*, pages 165–178. ACM, 2009.

[23] H. Lu, J. Yang, Z. Liu, N. Lane, T. Choudhury, and A. Campbell. The Jigsaw Continuous Sensing Engine for Mobile Phone Applications. In *SenSys '10*, pages 71–84. ACM, 2010.

[24] E. Miluzzo, C. Cornelius, A. Ramaswamy, T. Choudhury, Z. Liu, and A. Campbell. Darwin Phones: The Evolution of Sensing and Inference on Mobile Phones. In *MobiSys '10*, pages 5–20. ACM, 2010.

[25] E. Miluzzo, N. Lane, K. Fodor, R. Peterson, H. Lu, and M. Musolesi. Sensing Meets Mobile Social Networks: The Design, Implementation and Evaluation of the CenceMe Application. In *SenSys '08*, pages 337–350. ACM, 2008.

[26] D. Peebles, T. Choudhury, H. Lu, N. Lane, and A. Campbell. Community-Guided Learning: Exploiting Mobile Sensor User to Model Human Behavior. In *AAAI*, 2010.

[27] C. Peng, G. Shen, Y. Zhang, and S. Lu. Point and Connect: Intention-based Device Pairing for Mobile Phone Users. In *MobiSys '09*, pages 137–150. ACM, 2009.

[28] M. Quwaider and S. Biswas. Body Posture Identification using Hidden Markov Model with a Wearable Sensor Network. In *Bodynets '08*, pages 19:1–19:8. ICST, 2008.

[29] K. Rachuri, C. Mascolo, M. Musolesi, and P. Rentfrow. SociableSense: Exploring the Trade-offs of Adaptive Sampling and Computation Offloading for Social Sensing. In *MobiCom '11*. ACM, 2011.

[30] E. Shih, A. Shoeb, and J. Guttag. Sensor Selection for Energy-Efficient Ambulatory Medical Monitoring. In *MobiSys '09*, pages 347–358. ACM, 2009.

[31] Y. Wang, J. Lin, M. Annavaram, Q. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh. A Framework of Energy Efficient Mobile Sensing for Automatic User State Recognition. In *MobiSys '09*, pages 179–192. ACM, 2009.

[32] J. Wiese, P. Kelley, L. Cranor, L. Dabbish, J. Hong, and J. Zimmerman. Are you close with me? Are you nearby? Investigating social groups, closeness, and willingness to share. In *UbiComp '11*, pages 197–206. ACM, 2011.

[33] A. Wood, J. Stankovic, G. Virone, L. Selavo, Z. He, Q. Cao, T. Doan, Y. Wu, L. Fang, and R. Stoleru. Context-Aware Wireless Sensor Networks for Assisted-Living and Residential Monitoring. *IEEE Network*, 22(4), 2008.

[34] Z.-H. Zhou, J. Wu, and W. Tang. Ensembling Neural Networks: Many Could Be Better Than All. *Artificial Intelligence*, 137:239–263, 2002.