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

Matthew Keally[1], Gang Zhou, Guoliang Xing[†], and Jianxin Wu[‡]
College of William and Mary, [†]Michigan State University, [‡]Nanyang Technological University
{makeal,gzhou}@cs.wm.edu, [†]glxing@msu.edu, [‡]jxwu@ntu.edu.sg

***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 other resource sharing approaches, Remora provides both 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 Remora provides up to a 30% accuracy increase while extending phone battery lifetime by over 65%.**

I. 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. Conversely, a smartphone, in conjunction with on-body nodes, provides additional sensing power, computational capability, portability, and a user-friendly interface for personal control and runtime feedback. Activity recognition applications which can exploit BSNs include assisted living [1], physical fitness assessment [2], and patient monitoring [3] [4]. A physician may administer BSNs for retirement community residents to detect depression and ensure 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 [5]. 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 [6], thus requiring frequent recharges.

To address energy and accuracy concerns, a promising approach is to share sensing resources among users; however the existing solutions focus on either accuracy or energy improvements but not both. The authors of [7] provide an energy cost-benefit model for smartphone-based context sharing, but do not address potential accuracy gains. Other energy saving approaches [1] [8] do not share resources in direct proximity, instead using backend servers. Instead of energy, some methods aim to improve application performance, such as activity classification accuracy, through sharing classifiers [9] or sensor data [10] [11].

Consequently, we propose that BSNs in physical proximity to one another opportunistically share resources to improve both energy savings and activity classification accuracy. In our work, BSN neighbors, such as family and friends, exploit overheard on-body sensor data transmissions to increase classification accuracy. Unlike existing work which shares high level context data, our approach permits each on-body sensor to be shared independently. By using available neighbor sensors that are both individually accurate and have complimentary classification capabilities, we can significantly increase classification accuracy. Through sharing, neighbors can use fewer sensors, allowing more to be disabled to save energy. Also, to increase phone battery life, classifiers are duty cycled among neighbors, allowing one phone to make classifications for all neighbors while all other phones go into a low power sleep.

To improve both classification accuracy and extend system lifetime among neighboring BSNs, we present Remora. Remora addresses three prominent challenges that arise from BSN resource sharing. First, we determine when to share. By characterizing the costs and benefits of sharing at a resource level, we determine the energy overhead required for each BSN to collect training data, train new sensor classifiers

This work is supported in part by NSF grants CNS-1250180 and ECCS-0901437 as well as National Natural Science Foundation of China grant 61170296.

[1]Matthew Keally is now at The MITRE Corporation.

978-1-4799-0590-4/13/$31.00 ©2013 IEEE

for newly detected neighbors, and compare against predicted energy benefits. Second, we determine what to share: through a sharing aware classification approach, Remora finds and utilizes the sensors and classifiers that provide the best combination of accuracy and energy savings. Third, we determine how to share: our sharing-aware classification approach uses an ensemble classifier that efficiently adapts to changes in neighbor and sensor availability. Sharing BSNs 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 receive an energy 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: In Section II, we present related work and provide motivation behind our design in Section III. We discuss our Remora design in Section IV, describe our Sharing-Aware Classification approach in Section V, and explain the cost and benefits of resource-level sharing in Section VI. We evaluate Remora performance in Section VII, and present conclusions and future work in Section VIII.

II. Related Work

Several works share sensing resources with the aim of saving energy and do not focus on other performance goals, such as classification accuracy. In [7], an energy cost-benefit model is used to share context among neighboring phones, allowing neighbors to periodically power down sensors to save energy. Furthermore, sharing is only conducted at the context level; sensors that cannot provide context information on their own are not shared individually. Another effort [12] also offloads sensing tasks to nearby fixed sensors but not among multiple users. For detecting neighboring devices, [13] presents an energy efficient protocol. Another effort [8] shares neighboring phone data via backend servers and uses an adaptive sampling rate for energy savings. Also, the authors of [14] offload execution to backend servers to save energy.

Other works share resources with the explicit goal of improving application performance, but do not focus on energy savings. In [10], 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 [9] [15], which increases accuracy. Rather than share with physical neighbors, users share data on a backend server for collaborative video editing [16] and classifying

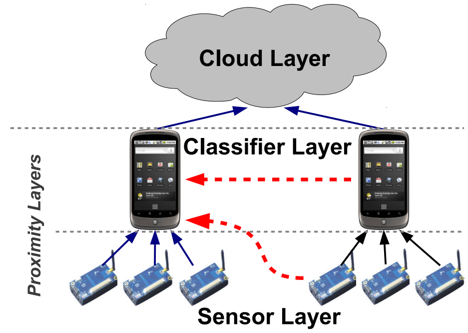


Fig. 1: Sharing Hierarchy.

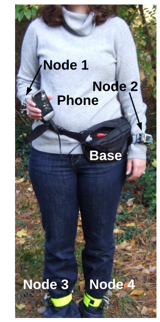


Fig. 2: BSN setup.

user interactions [17] [1] as well as share training data or classification models [11] [18] to reduce training overhead.

Many existing on-body sensing and activity classification approaches do not allow any collaboration among users. On-body sensors are used for classification [2] [4], some of which [6] [19] provide energy saving methods. Other approaches [20] [21] use only smartphone sensors for activity classification. A phone-only classification technique [22] provides an energy-latency-accuracy tradeoff, while other methods [23] [24] achieve energy savings with adaptive sampling.

III. Feasibility and Motivation

In this section, we first discuss sharing feasibility and how BSNs can share sensors and classifiers. Then, we present our experimental configuration, and lastly, we show in a short experiment the potential accuracy and energy benefits of sharing.

A. Feasibility

We first discuss three issues concerning the feasibility of sharing: opportunity, resources, and privacy:

Sharing Opportunity. Our approach targets applications where users have strong interpersonal ties. Along these lines, the MIT Reality Mining [25] dataset, which analyzed physical interactions among students and faculty, demonstrates that on average each subject was in proximity with at least one other subject 25% of the time. Our evaluation in Section VII yields similar results: subjects were collocated 30-50% of the time.

Sharing Hierarchy. In Figure 1, we present BSN resources which are eligible for sharing among users. We use the bottom two layers (Sensor and Classifier) to exploit proximity, improving accuracy and energy use. We leave the Cloud Layer for future work. 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. Neighbors collaborate to select a set of sensors that achieves higher accuracy and uses fewer combined sensors compared with individual classification.

To save phone energy, 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

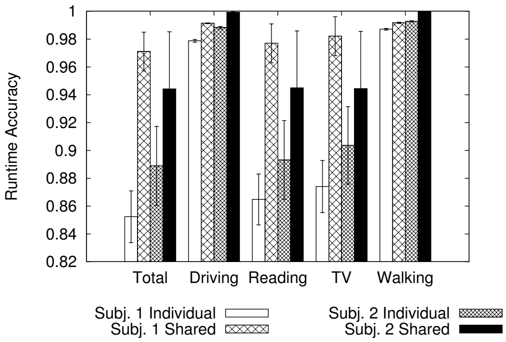


Fig. 3: Accuracy for shared and individual BSNs.

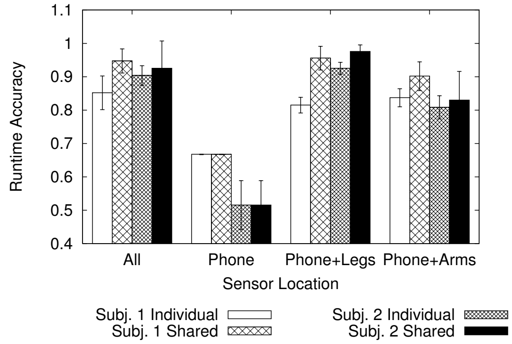


Fig. 4: Accuracy for different combinations of sensors.

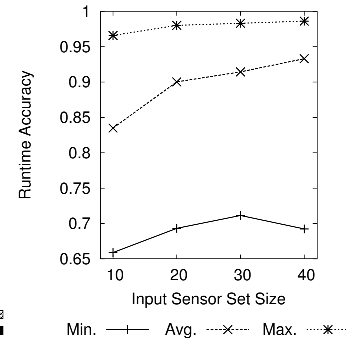


Fig. 5: Randomly sensor clusters.

classification decisions for all neighbors, neighbors only share if they are all performing the same activity. However, in 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.

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 [26], 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 define a wireless heart rate monitor or pulse oximeter as private. 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 method [27] obfuscates personally identifiable characteristics of phone sensor data.

B. 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 devices. We present details on sensors, sampling, and classification:

Sensors. On the phone, which we attach to the waist, we use 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 mote sensors, the base station collects RSSI from received packets, which has been shown [28] to provide insight into body posture. Each subject makes all on-body sensors public and all phone sensors private.

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) [29] classifier, detailed in Section V. During the experiment, subjects recorded all activity ground truth in order to evaluate the accuracy of training data (training accuracy) and runtime accuracy.

C. Motivation: Identifying Sharing Benefits

Through a shared activity experiment with 2 BSNs, we show how sharing can improve accuracy and save energy. Two subjects performed four shared activities (driving, reading, walking, and watching TV) for over four hours. We use the same data to compute individual and shared classification results, using 10 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

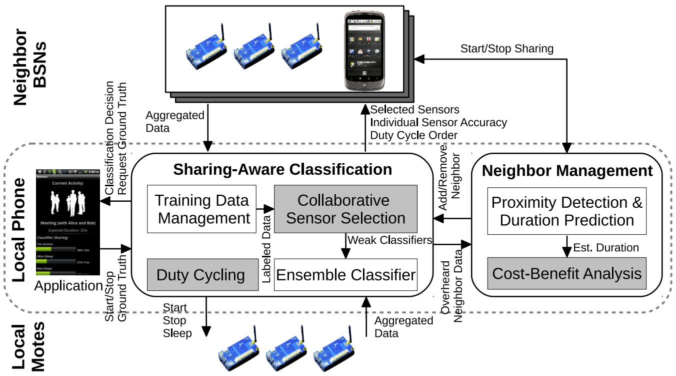


Fig. 6: Remora Architecture. Neighbor Management determines if sharing with detected neighbors will provide an energy benefit. Sharing-Aware Classification collaborates with neighbors to select sensors for shared classification, classifies sensor data from the local phone, local motes, and neighboring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

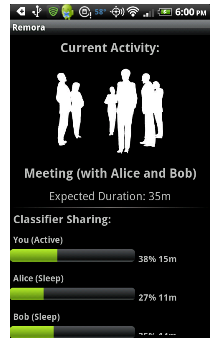


Fig. 7: Activity status.

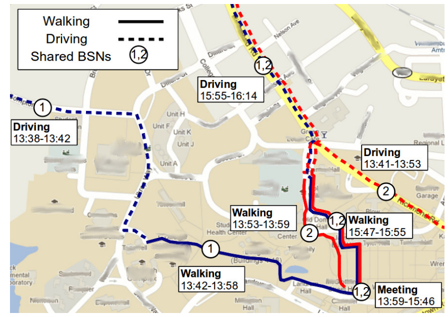


Fig. 8: Map view of individual and shared activities.

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 V-C, that chooses a small number of sensors to achieve both high accuracy and node energy savings.

IV. Design Overview

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 phone 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 determines the costs and benefits of sharing and initiates sharing only when an energy benefit is possible. Using Proximity Detection and Duration Prediction, a BSN detects a neighbor when it overhears a neighbor's sensor data radio transmissions. Since detected neighbors must be in proximity long enough for sharing to provide an energy benefit, we predict neighbor proximity in a manner similar to [7], where duration is predicted based on the type of neighbor (colleague, friend, relative, or stranger) and the current activity. For example, in [7], work related activities with colleagues average 200 minutes compared to encounters with strangers or eating, which average less than 45 minutes. To increase duration prediction accuracy, we combine this method with online shared calendar entries provided by each user as in [30].

The proximity duration estimate is then combined with an empirical model in Cost-Benefit Analysis to determine the energy costs and benefits of sharing. 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 [29], 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.

At the start of runtime, each BSN uses a classifier for individual classification based on available training data (Training Data Management) or a previously trained classifier. When sharing is initiated, Collaborative Sensor Selection allows BSNs to work together 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 while sharing, one active neighbor makes a classification decision for the group at each aggregation interval (Duty Cycling). 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.

UI and Applications. 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. 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 8, 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.

V. Sharing-Aware Classification

In this section, we first explain details on our classifier as well as what happens when Cost-Benefit Analysis initiates sharing. Second, if sharing is initiated and a new classifier is needed, we provide a motivation experiment and then explain how neighboring BSNs train new classifiers by collaboratively choose sensors for shared classification. Lastly, we explain how BSNs share classifiers and duty cycle them to significantly increase phone battery life.

A. *Classification and Sharing Initialization*

We use an ensemble technique, Bootstrap Aggregating (Bagging) [29] 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 as is performed in [6]. Other sharing approaches use more complex techniques, such as GMMs trained offline [9] or Boosting [11].

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, when neighbors detect each other, Collaborative Sensor Selection proceeds only for the BSNs that agree that sharing will be beneficial.

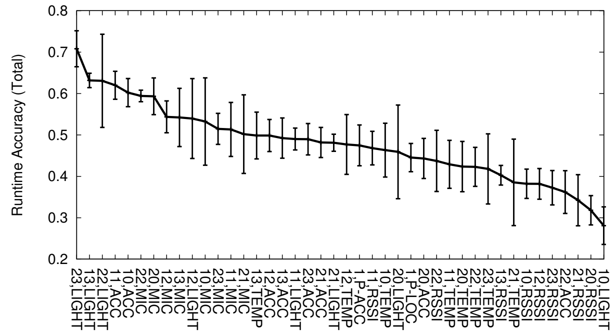


Fig. 9: Subject 1: Individual sensor accuracy.

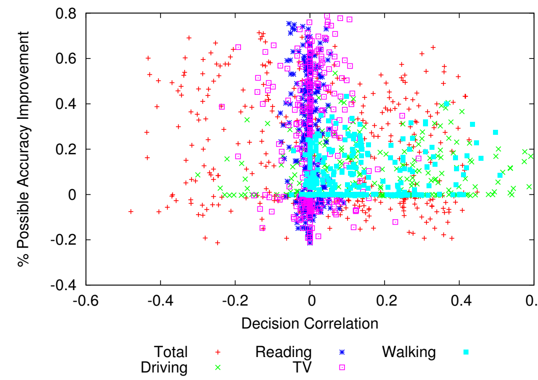


Fig. 10: Decision correlation and accuracy improvement.

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 (5 min. in evaluation), each neighbor trains a sensor classifier for each available sensor and broadcasts its intent to start Collaborative Sensor Selection.

B. *Sensor Selection Motivation*

We provide intuition for our Collaborative Sensor Selection design using data from our evaluation in Section VII. The challenge is to identify properties of both local and neighbor sensors such that all neighbors can choose sensors to create composite classifiers that are accurate for all neighbors. Previous work [31] 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 9) and decision correlation (Figure 10) will create an accurate composite classifier with a small number of capable sensors.

We first show in Figure 9 that we can discriminate best between activities by choosing sensors with the best individual accuracies. Using the evaluation data for 2 subjects, 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 each node ID starts with the subject ID. We can see that sensors on both Subject 1 and 2 exhibit high accuracy, indicating that sharing gives Subject 1 more accurate sensors from which to choose.

Next, we show how to find sensors with complimentary classification capability, locating a combination of sensors that is not only accurate but contains few sensing redundancies. Figure 10 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.

C. Collaborative Sensor Selection

Algorithm 1 Collaborative Sensor Selection

Input Sensor classifiers for local sensors and public neighbors

Output Composite classifier for local BSN C_i

// BSN initiates sharing

1: **function** START

2: $C_i = \emptyset$; Send public sensor accuracies to neighbors

// Receive sensor accuracy or correlation values from all BSNs

3: **event** CHOOSESENSOR(C)

4: Compute weight for unchosen sensors with Equation 2

5: Add private sensors with highest weight to C_i

6: **if** only one public sensor has the highest weight **then**

7: Add sensor with highest weight to C_i

8: **else if** local BSN ID is the lowest of all neighbors **then**

9: Local BSN is tiebreaker

10: Choose random sensor with the highest weight

11: Add random sensor to C_i, notify neighbors of choice

12: **else**

13: Add tiebreaker's sensor choice to C_i

14: **if** acc(C_i) < 1 and unchosen sensors remain: **then**

15: Transmit unchosen public sensor correlation values

Based on the motivation results, 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 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 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 2.

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

$$w(C, s_j) = \frac{1}{B}\sum_{i=1}^{B} w_i(C_i, s_j) \quad (2)$$

In Equation 2, 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 1 only.

After computing weights (line 4 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 8-12). 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 (line 5). 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 [29], 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.

D. Classifier Sharing and Duty Cycling

After sharing is initiated and a classifier is trained, all neighbors collaborate to define a duty cycle order where only one phone at a time is 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. We use a round robin duty cycling scheme to ensure fairness in energy consumption and choose a duty cycle of appropriate length (5 min. in evaluation) so that sleeping neighbors are able to quickly wake up and detect changes in available neighbors. Upon waking up, if at least one sharing neighbor has departed, the BSN reverts to individual classification using a saved classifier and notifies all remaining sharing neighbors.

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 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.

VI. Cost-Benefit Analysis

Following neighbor detection and proximity duration prediction, 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 about 10 hours of battery life using 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 first describe a general energy model in Section VI-A, where we define costs in terms of training time and energy and define the benefits in terms of energy saved during low power duty cycling among neighbors compared with always-on individual classification. Then, in Section VI-B, we empirically determine the cost model parameters based on our BSN hardware and sharing aware approach.

A. Energy Model

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

$$\begin{aligned} E_{tr} = & [(T_{GT} + T_{tr})(P_{class} + P_{sensor})] \\ & + (T_{GT} \cdot P_{GT}) + (T_{tr} \cdot P_{tr}) \end{aligned} \tag{3}$$

In Equation 3, $T_{GT} + T_{tr}$ 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}:

$$\begin{aligned} E_{share} = & [T_{prox} - (T_{GT} + T_{tr})] \\ & \cdot \left[\frac{1}{b}(P_{class} + P_{sensor}) + \left(1 - \frac{1}{b}\right) P_{sleep}\right] \end{aligned} \tag{4}$$

In Equation 4, 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 to ensure energy fairness.

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

$$E_{ind} = T_{prox} \cdot (P_{class} + P_{sensor}) \tag{5}$$

Equation 5 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:

$$T_{prox} > (T_{tr} + T_{GT}) \text{ and } E_{tr} + E_{share} < E_{ind} \tag{6}$$

In Equation 6, 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.

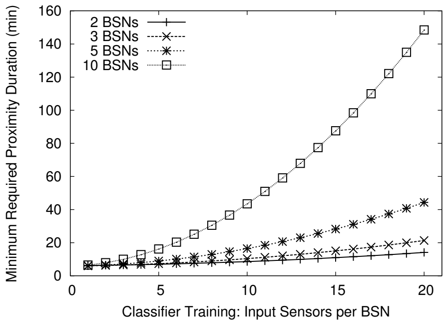
B. Empirical Sharing Cost Model

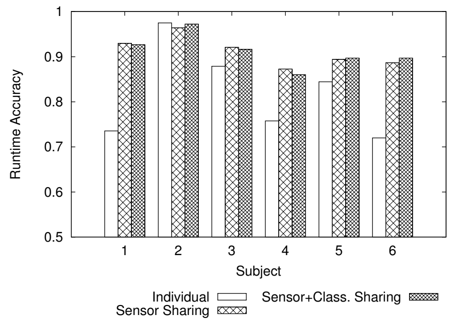
Using our Remora Collaborative Sensor Selection implementation, we perform time and power benchmarks using an HTC Hero smartphone and four on-body sensor motes as described in Section III. We use the benchmarks to define the training time, training power, and minimum proximity duration needed for sharing to provide an energy benefit. We measure power consumption by connecting the smartphone BSN component to a Monsoon Technologies Power Meter as in [6], demonstrating that we achieve massive phone energy savings by duty cycling classifiers. The average power consumption for each state is provided in Table I: 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. The table demonstrates that a sleeping phone (duty cycling) consumes fewer than 10mW of power, which is much less than the nearly 500mW required for classification.

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

TABLE I: Remora Power Consumption (mW).

We also measure the training time required to train new sensor classifiers and perform Collaborative Sensor Selection. 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





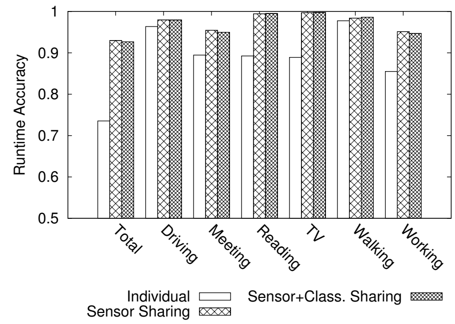


Fig. 11: Prox. duration needed for sharing benefit.

Fig. 12: Runtime accuracy for each subject.

Fig. 13: Subject 1 accuracy.

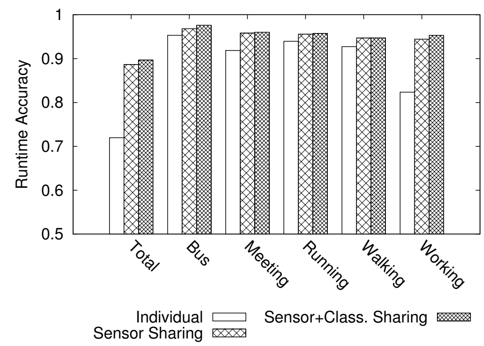


Fig. 14: Subject 6 accuracy.

correlation is computed for each unchosen sensor classifier. By varying the number of sensors as input to Collaborative Sensor Selection, we perform polynomial curve fitting to determine the training time in milliseconds for a composite classifier of n sensors. For use in our cost model, Equation 7 presents the training time $T_{tr}(n)$:

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

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 reduce training overhead by ensuring that only sensors chosen by each BSN for individual classification are given as input for shared classification. Furthermore, Equation 7 illustrates 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 11, 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 VI-A, 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. However, with more neighbors, more time is required: for 10 neighbors, over 40 minutes is needed to realize an energy benefit. Since [7] demonstrates that 60% of shared activities with non-strangers are longer than 40 minutes, this performance is reasonable.

VII. Evaluation

For our evaluation, 6 subjects perform the following activities over a two week period: riding a bus, riding in or driving a car (driving), meeting, reading, running, watching TV, walking, and working at a desk. Each of these activities was performed both individually and with at least one neighbor in proximity. The subjects all have 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. All on-body sensors are public, while phone sensors are not shared. 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 VII-A, we demonstrate sharing accuracy improvements. Then, in Section VII-B, we highlight the benefits of sharing, and lastly, we show significantly improved battery life in Section VII-C.

A. Accuracy Improvement

In Figure 12, 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 12, Subject 2 has the highest accuracy, 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 13, we can see that Subject 1 exhibits confusion between

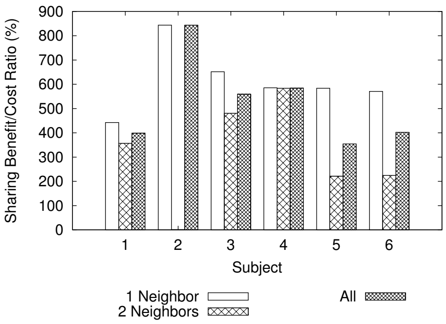


Fig. 15: Energy benefit to cost ratio.

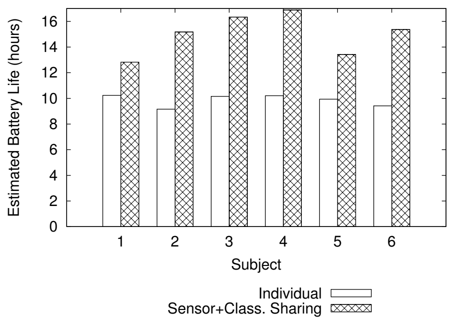


Fig. 16: Battery Life.

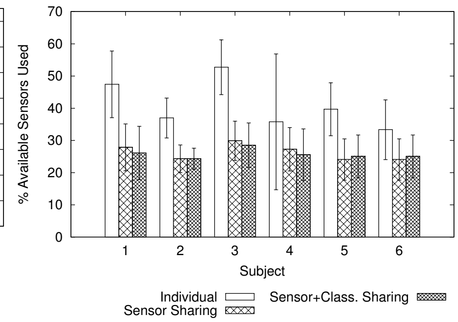


Fig. 17: Percentage of available sensors chosen.

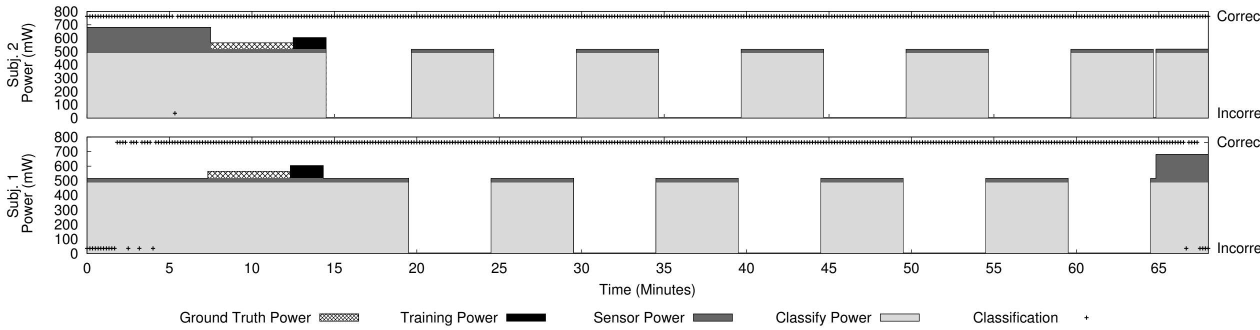


Fig. 18: Sharing timeline: energy and accuracy profile.

meeting and working in addition to reading and watching TV. 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 14 for Subject 6, where total accuracy is also significantly improved by sharing. Lastly, we note that Collaborative Sensor Selection ensures that accuracy can only be improved by sharing: high individual accuracy is maintained during sharing while low individual accuracy is improved.

B. Sharing Costs and Benefits

For every subject, using our empirical Cost-Benefit Analysis model, the benefits of sharing outweigh the costs by two orders of magnitude. First, from our cost model, in Figure 15 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 2 others at a time, however, the bulk of the energy savings comes from sharing with one neighbor. The marginal benefit of sharing with an additional neighbor decreases 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 Cost-Benefit Analysis.

Training Overhead and Classifier Reuse. When we perform sharing with classifier reuse, an average of 96% of the total proximity duration is utilized. The remaining 4% difference includes Cost-Benefit Analysis rejection due to short proximity durations or different simultaneous activities as well as 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, however, sharing overhead is higher, with an average of 78% of the total proximity time utilized.

C. Energy Savings

We now demonstrate that by sharing and duty cycling classifiers, we can increase phone battery life by over 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 I and a 1500mAh battery per phone, we present results in Figure 16. 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%.

While accuracy is increased over individual classification, sharing sensors can also reduce the total number of sensors used by all neighbors. In Figure 17, 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.

To further highlight classifier sharing as well as the ability of Remora to adapt classification to available neighbors, Figure 18 presents a timeline of energy use and classification decisions for Subject 1 and 2 during shared classification. Subject 1 and 2 perform individual activities until 7 minutes, where Subject 2 enters 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.

VIII. Conclusion and Future Work

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%.

There are several areas that can benefit from further investigation. First, in our evaluation, neighbors perform the same activity 95% of the time, but this may not always be the case. To address this, we will investigate approaches that can classify multiple activities simultaneously such as in [32] while still allowing phones to duty cycle; cloud-based sharing is one possible solution. Second, when many new neighbors are encountered, the user burden of collecting new training data may be fairly high. Incorporating semi-supervised training methods as in [9] can help alleviate these cases. Lastly, studying the privacy implications of sharing different sensor types is also a topic for future work.

References

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

[2] F. Albinali, S. Intille, W. Haskell, and M. Rosenberger, "Using Wearable Activity Type Detection to Improve Physical Activity Energy Expenditure Estimation," in *ACM UbiComp*, 2010.

[3] A. Vahdatpour, N. Amini, and M. Sarrafzadeh, "On-Body Device Localization for Health and Medical Monitoring Applications," in *IEEE PerCom*, 2011.

[4] 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 *ACM SenSys*, 2010.

[5] M. Baechlin, K. Foerster, and G. Troester, "SwimMaster: A Wearable Assistant for Swimmer," in *ACM UbiComp*, 2009.

[6] M. Keally, G. Zhou, G. Xing, J. Wu, and A. Pyles, "PBN: Towards Practical Activity Recognition Using Smartphone-Based Body Sensor Networks," in *ACM SenSys*, 2011.

[7] Y. Lee, Y. Ju, C. Min, S. Kang, I. Hwang, and J. Song, "CoMon: Cooperative Ambience Monitoring Platform with Continuity and Benefit Awareness," in *ACM MobiSys '12*, 2012.

[8] S. Gaonkar, J. Li, R. Choudhury, L. Cox, and A. Schmidt, "Micro-Blog: Sharing and Querying Content Through Mobile Phones and Social Participation," in *ACM MobiSys*, 2008.

[9] 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 *ACM MobiSys*, 2010.

[10] E. Koukoumidis, L. Peh, and M. Martonosi, "SignalGuru: Leveraging Mobile Phones for Collaborative Traffic Signal Schedule Advisory," in *ACM MobiSys*, 2011.

[11] 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 *ACM UbiComp*, 2011.

[12] K. Rachuri, C. Efstratiou, I. Leontiadis, C. Mascolo, and P. Rentfrow, "METIS: Exploring mobile phone sensing offloading for efficiently supporting social sensing applications," in *IEEE PerCom*, 2013.

[13] M. Bakht, M. Trower, and R. Kravets, "Searchlight: Won't You Be My Neighbor?" in *ACM MobiCom*, 2012.

[14] S. Yang, Y. Kwon, Y. Cho, J. Youn, and Y. Paek, "Fast Dynamic Execution Offloading for Efficient Mobile Cloud Computing," in *IEEE PerCom*, 2013.

[15] M. Rossi, O. Amft, M. Kusserow, and G. Troster, "Collaborative Real-Time Speaker Identification for Wearable Systems," in *IEEE PerCom*, 2011.

[16] X. Bao and R. Choudhury, "MoVi: Mobile Phone based Vidoe Highlights via Collaborative Sensing," in *ACM MobiSys*, 2010.

[17] R. Bajcsy, A. Giani, C. Tomlin, A. Borri, and M. D. Benedetto, "Classification of Physical Interactions Between Two Subjects," in *IEEE BSN*, 2009.

[18] N. Lane, H. Lu, S. Eisenman, and A. Campbell, "Cooperative Techniques Supporting Sensor-Based People-Centric Inferencing," in *Pervasive*, 2008.

[19] E. Shih, A. Shoeb, and J. Guttag, "Sensor Selection for Energy-Efficient Ambulatory Medical Monitoring," in *ACM MobiSys*, 2009.

[20] H. Lu, W. Pan, N. Lane, T. Choudhury, and A. Campbell, "SoundSense: Scalable Sound Sensing for People-Centric Sensing Applications on Mobile Phones," in *ACM MobiSys*, 2009.

[21] H. Lu, J. Yang, Z. Liu, N. Lane, T. Choudhury, and A. Campbell, "The Jigsaw Continuous Sensing Engine for Mobile Phone Applications," in *ACM SenSys*, 2010.

[22] 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 *ACM SenSys*, 2011.

[23] 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 *ACM SenSys*, 2008.

[24] 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 *ACM MobiSys*, 2009.

[25] N. Eagle, A. Pentland, and D. Lazer, "Inferring Social Network Structure using Mobile Phone Data," *PNAS*, vol. 106, no. 36, 2009.

[26] 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*. ACM, 2011, pp. 197–206.

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

[28] M. Quwaider and S. Biswas, "Body Posture Identification using Hidden Markov Model with a Wearable Sensor Network," in *Bodynets*, 2008.

[29] L. Breiman, "Bagging predictors," *Machine Learning*, 1996.

[30] 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 *ACM UbiComp*, 2010.

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

[32] R. Helaoi, M. Niepert, and H. Stuckenschmidt, "Recognizing Interleaved and Concurrent Activities: A Statistical-Relational Approach," in *IEEE PerCom*, 2011.