

# Learning to Infer the Status of Heavy-Duty Sensors for Energy-Efficient Context-Sensing

XUEYING LI, University of Science and Technology of China  
HUANHUAN CAO, Nokia Research Center  
ENHONG CHEN, University of Science and Technology of China  
JILEI TIAN, Nokia Research Center

With the prevalence of smart mobile devices with multiple sensors, the commercial application of intelligent context-aware services becomes more and more attractive. However, limited by the battery capacity, the energy efficiency of context-sensing is the bottleneck for the success of context-aware applications. Though several previous studies for energy-efficient context-sensing have been reported, none of them can be applied to multiple types of high-energy-consuming sensors. Moreover, applying machine learning technologies to energy-efficient context-sensing is underexplored too. In this article, we propose to leverage machine learning technologies for improving the energy efficiency of multiple high-energy-consuming context sensors by trading off the sensing accuracy. To be specific, we try to infer the status of high-energy-consuming sensors according to the outputs of software-based sensors and the physical sensors that are necessary to work all the time for supporting the basic functions of mobile devices. If the inference indicates the high-energy-consuming sensor is in a stable status, we avoid the unnecessary invocation and instead use the latest invoked value as the estimation. The experimental results on real datasets show that the energy efficiency of GPS sensing and audio-level sensing are significantly improved by the proposed approach while the sensing accuracy is over 90%.

Categories and Subject Descriptors: I.2.6 [Artificial Intelligence]: Learning; H.3.4 [Information Storage and Retrieval]: Systems and Software

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Context-sensing, energy efficiency, machine learning

## ACM Reference Format:

Li, X., Cao, H., Chen, E., and Tian, J. 2012. Learning to infer the status of heavy-duty sensors for energy-efficient context-sensing. *ACM Trans. Intell. Syst. Technol.* 3, 2, Article 35 (February 2012), 23 pages. DOI = 10.1145/2089094.2089111 <http://doi.acm.org/10.1145/2089094.2089111>

## 1. INTRODUCTION

To enable the intelligent mobile applications of the future, it is important to understand mobile users through the data collected from their mobile devices. In recent years, more and more commercial mobile devices such as smart phones and personal

---

The research is supported in part by Natural Science Foundation of China (grant no. 61073110), the Key Program of National Natural Science Foundation of China (grant no. 60933013), the National Major Special Science and Technology Projects (grant no. 2011ZX04016-071).

Authors' addresses: X. Li, School of Computer Science and Technology, University of Science and Technology of China; H. Cao, Beijing Lab of Nokia Research Center, China; E. Chen (corresponding author), School of Computer Science and Technology, University of Science and Technology of China; email: [cheneh@ustc.edu.cn](mailto:cheneh@ustc.edu.cn); J. Tian, Beijing Lab of Nokia Research Center, China.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from the Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701, USA, fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org).

© 2012 ACM 2157-6904/2012/02-ART35 \$10.00

DOI 10.1145/2089094.2089111 <http://doi.acm.org/10.1145/2089094.2089111>

digital assistants are equipped with multiple context sensors including optical sensors, 3D accelerometers, GPS sensors, etc., which makes it possible to bring to bear intelligent context-aware applications to ordinary mobile users. Moreover, many interesting kinds of knowledge can be discovered from the collected context data (e.g., GPS trajectories and usage logs) through data mining technologies [Farrahi and Gatica-Perez 2011; Zheng and Xie 2011]. However, the battery capacity of mobile devices becomes the bottleneck of context-aware applications because some context sensors are very energy consuming and cannot continuously work for the sake of user experience. For example, as stated in Abdesslem et al. [2009], the battery life of Nokia N95 GB is more than 170 hours when all sensors are turned off. However, when the GPS sensor is turned on, the battery life is reduced to 11.6 hours (indoor) and 7 hours (outdoor), respectively.

Some studies have been done for improving the energy efficiency of context-sensing. For example, Abdesslem et al. [2009] proposed to selectively invoke the GPS sensor by recognizing the user's movement through the 3D accelerometer. Constandache et al. [2009] proposed to dynamically select one sensor from the GPS, the Wifi, and the cell ID sensors to provide location information by balancing the remaining energy and the accuracy. However, these works still need to invoke other high-energy-consuming sensors for improving the energy efficiency of GPS sensing. Moreover, they cannot be applied to improving the energy efficiency of other high-energy-consuming sensors except for GPS sensors.

We argue that the outputs of different context sensors of a mobile device may be more or less correlated since they essentially capture the same context at each time point, even though from different perspectives. Intuitively, we may be able to selectively avoid invoking high-energy-consuming sensors by inferring their statuses from the outputs of other sensors as illustrated by the following example.

*A Motivating Example.* Figure 1 illustrates an example of avoiding the unnecessary reinvocations of the GPS sensor by correctly inferring its status. Suppose that Sam's GPS coordinates rarely change<sup>1</sup>: (1) during work days PM2:00-6:00, (2) when his cell ID is 2344 (indicates his workplace), and (3) he has not operated his mobile phone for more than 40 minutes, a well-trained model based on his historical context data may be able to infer that the GPS sensor of Sam's mobile device is in a stable status given the context that  $\{(Is\ a\ work\ day? : Yes), (Hour : PM3:00), (cell\ ID : 2344), (Inactive\ time : 40(min))\}$ , and thus there is no need to invoke the GPS sensor again if it has been invoked a moment ago.

In this article, we propose to capture the relationship among context sensors through machine learning technologies for improving the energy efficiency of multiple high-energy-consuming context sensors by trading off the sensing accuracy. To be specific, firstly we group context sensors into three categories, namely, the *basic sensors* which are necessary to continuously invoke for providing the basic functions of mobile devices, the *light-duty sensors* which are software based and energy efficient, and the *heavy-duty sensors* which are not necessary to continuously invoke and relatively high energy consuming. Then we learn inference models to capture the relationship between heavy-duty sensors and basic/light-duty sensors and use the model to infer the status of a heavy-duty sensor according to the outputs of basic and light-duty sensors. If the output of a heavy-duty sensor  $s$  at time point  $t_i$  is same as the one at time point  $t_{i-1}$ , we state  $s$  is in a *stable status* at time point  $t_i$ . Otherwise we state  $s$  is in an *unstable status* at time point  $t_i$ . If the model infers that the heavy-duty sensor is likely

<sup>1</sup>In this case we take the loss of GPS signal as no movement when the user is indoors.

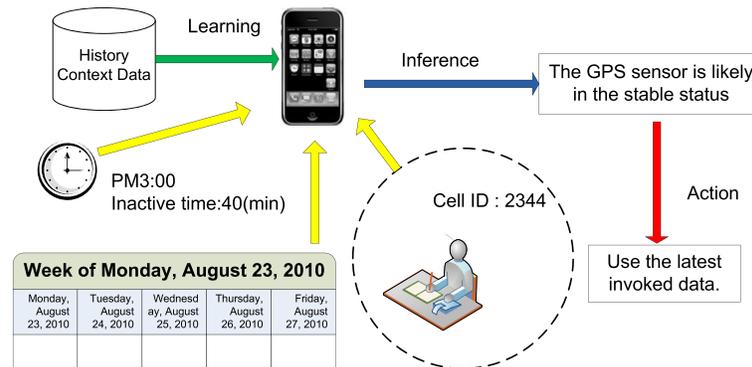


Fig. 1. A well-trained model may be able to infer the status of the GPS sensor and then avoid the unnecessary reinocations.

in a stable status, the context-sensing system will not invoke it again and instead use the latest invoked value.

The invocations of basic sensors do not bring additional energy consumption since they are always on. Though the approach has a need to invoke light-duty sensors, the amount of energy consumed by the light-duty sensors is expected to be much less than that of the heavy-duty sensors. Moreover, the training of models for inferring the statuses of heavy-duty sensors can be done offline, such as when the mobile device is charging, for avoiding the additional burden on the battery capacity.

The contributions of this article are summarized as follows.

First, we propose to leverage machine learning technologies for learning inference models from personal historical context data which can naturally fit the personalized context-sensing environments by reducing as much energy consumption as possible while keeping the necessary accuracy.

Second, we propose a general approach for energy-efficient context-sensing which can be applied to improving the energy efficiency of multiple heavy-duty sensors, such as GPS sensors, audio-level sensors, and Bluetooth sensors, etc.

Finally, we simulate several variants of the context-sensing system with different learning models and conduct extensive experiments on real mobile context datasets to evaluate their effectiveness for two kinds of heavy-duty sensors, namely, GPS sensors and audio-level sensors. From the experimental results we have some inspiring conclusions.

The rest of this article is organized as follows. In Section 2, we give an overview of the idea of learning-based context-sensing. Then in Section 3, we propose two kinds of inference models for inferring the statuses of heavy-duty sensors. In Section 4 we propose a framework of the learning-based context-sensing system. The experimental results on real datasets are reported and analyzed in Section 5. In Section 6, we briefly review some related works. Last, we conclude this article and introduce our future research plan for this problem in Section 7.

## 2. LEARNING-BASED CONTEXT-SENSING

To support different context-aware applications, a mobile device usually invokes multiple context sensors for providing multiple types of context information, which is referred as context-sensing. It is worth noting that the term of “context sensor” does not only indicate the physical sensors such as GPS sensors and 3D accelerometers, but also indicates software-based sensors such as a program monitoring the inactive time

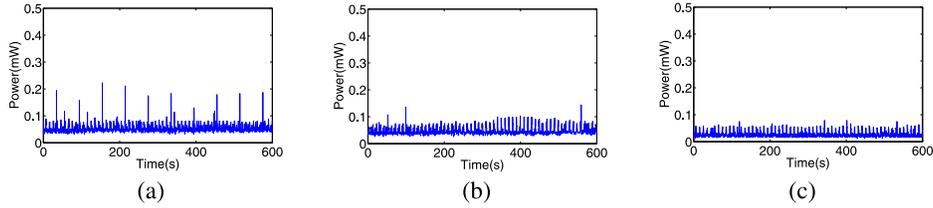


Fig. 2. Energy consumption of invoking several typical light-duty sensors including (a) running process monitor, (b) front application monitor, and (c) system inactive timer (to calculate the inactive time of the device).

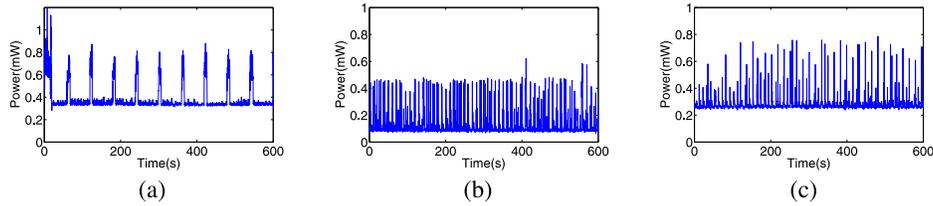


Fig. 3. Energy consumption of invoking several typical heavy-duty sensors including (a) GPS sensor, (b) 3D accelerometer, and (c) audio-level sensor.

of the mobile device. The context sensors of mobile devices can be grouped into three categories as follows.

*Basic Sensor.* The first category includes the context sensors which have to work all the time for supporting the basic functions of mobile devices, such as Cell ID sensors and time sensors (clock). We refer to this kind of context sensors as basic sensors. The improvement of energy efficiency for basic sensors mainly relies on the progress of electronic and material technologies.

*Light-Duty Sensor.* The second category includes all software-based sensors which do not have too much energy consumption. We refer to this kind of sensors as light-duty sensors.

*Heavy-Duty Sensor.* The last category includes the remaining context sensors which have relatively high energy consumption and are not necessarily always on, such as GPS sensors and audio-level sensors (microphone). We refer to these sensors as heavy-duty sensors. In this article, we focus on reducing the energy cost of heavy-duty sensors to achieve an energy-efficient strategy for context-sensing.

Figure 2 and Figure 3 show the energy consumptions of invoking several typical light-duty sensors and heavy-duty sensors, respectively. The experimental device is a Nokia N97 smart phone with 128M main memory. The experimental results are recorded by a professional Symbian platform-based energy monitoring software called Nokia Energy Profile. From the figures we can intuitively see that the energy consumption of light-duty sensors is much less than that of heavy-duty sensors.

Essentially, at each time point, the different context sensors of a mobile device capture the same context from different perspectives. It implies that the outputs of different context sensors at the same time point may be more or less correlated. As illustrated by the example in Figure 1, this assumption is usually true in practice. Therefore, we may be able train a model from the historical context-sensing data of a mobile user and use it for inferring the status of a particular heavy-duty sensor

according to the outputs of basic and light-duty sensors. For simplicity, we refer to such a model as a *Stable Status Inference* (SSI) *model*.

With an SSI model, we can selectively invoke a heavy-duty sensor if the model infers that it is probably in an unstable status at  $t_i$  time point by considering the outputs of basic and light-duty sensors at  $t_i$  time point. Commonly, the output of the invoked heavy-duty sensor after the inference step can be regarded the same as the real value at  $t_i$  time point because the time cost of inference is usually trivial (less than 10 ms in our experiments). Otherwise, if the model infers that the heavy-duty sensor is probably in a stable status, we can avoid invoking the heavy-duty sensor and instead just use its output at  $t_{i-1}$  time point. Therefore, the total energy consumption of context-sensing will be dramatically reduced if there are a lot of stable statuses practically and many of them are inferred correctly.

In practice, we determine whether a heavy-duty sensor  $s$  is in a stable status by taking into account the requirement on the accuracy of the outputs. For example, suppose the system requires the accuracy of GPS coordinates be at ten meters. If the GPS coordinate indicates that the user has moved less than ten meters from the previous sampled GPS coordinate, the system will determine that the GPS sensor is still on a stable status.

One may argue for a model of inferring the exact output of a heavy-duty sensor according to the outputs of basic and light-duty sensors. With such a model we can avoid invoking the heavy-duty sensor forever and reduce more energy consumption. However, training such a model is very challenging in practice. The training data of such a model should contain enough cases for each possible output of the heavy-duty sensor. Though this requirement may be feasible to be satisfied for some heavy-duty sensors which have limited different outputs (e.g., audio-level sensors), there are still many heavy-duty sensors with a big quantity of, even infinite, different possible outputs (e.g., GPS sensors). It is difficult to collect a comprehensive set of training data for a model to infer the exact outputs of these heavy-duty sensors. In contrast, SSI models provide a more practical solution for most heavy-duty sensors. The collection of a good training set for SSI models is much simpler because they only need learning to infer two statuses for all heavy-duty sensors. The experimental results in this article also show that SSI models outperform exact inference models.

### 3. STABLE STATUS INFERENCE MODELS

The SSI model is a key component in the learning-based context-sensing approach. In this section, we explore two kinds of SSI models, namely, the *Current Status Inference* (CSI) *model* and the *Status Interval Inference* (SII) *model*. To be specific, for each model, we firstly introduce the basic idea and then propose some typical implementations.

#### 3.1. Current Status Inference Models

**3.1.1. Overview of CSI Models.** For the outputs of basic and light-duty sensors at one time point, we can build a feature vector  $\mathbf{x}$ . First, we divide all basic and light-duty sensors into two categories. One category has numeric outputs, such as audio-level sensors. The other category has nonnumeric outputs, such as Cell ID sensors. For the sensors with numeric outputs, we directly take one sensor as a feature and take the particular output as the corresponding feature value. However, for the sensors with nonnumeric outputs, this method is not applicable because the different outputs of one sensor are difficult to be quantized. Instead, we take each particular output of such basic and light-duty sensors as a feature and take the occurrence of the output as a boolean feature value, that is, 1 indicates the feature appears and 0 indicates the

opposite case. Then, given  $K$  features extracted from the training data, we can build a  $K$ -dimension feature vector from the outputs of basic and light-duty sensors.

Given a status variable  $y$  for the heavy-duty sensor  $s$  and a feature vector extracted from the outputs of basic and light-duty sensors  $\mathbf{x}$ , a CSI model  $\mathcal{F}_s$  can be formally defined as a mapping function  $y = \mathcal{F}_s(\mathbf{x})$ , where  $y \in [-1, 1]$  indicates whether  $s$  is in a stable status ( $y = 1$ ) or not ( $y = -1$ ). A CSI model infers the status of a heavy-duty sensor  $s$  at time point  $t_i$  according to the outputs of basic and light-duty sensors at the same time point. Obviously, building a CSI model  $\mathcal{F}_s$  for the heavy-duty sensor  $s$  can be transformed to the problem of training a binary classifier. The trained classifier will take the outputs of basic and light-duty sensors as observed features and classify the status of a heavy-duty sensor  $s$  into the stable status category or the unstable status category. Therefore, we need a training dataset which contains sufficient samples of stable statuses and unstable statuses for  $s$  with the according outputs of basic and light-duty sensors. To prepare such a training dataset, we firstly transform the raw context records to a context record sequence by sorting them in order of timestamps. Then we assign each context record a status label for  $s$  by checking whether the output of  $s$  is different from the one in the previous context record. Finally, for each labeled context record we extract the corresponding feature vector and the status label for  $s$  to build a training sample.

Given a set of training samples, multiple existing classification approaches can be applied to building a CSI model. In this article, we explore two typical classification approaches, namely, support vector machine [Cortes and Vapnik 1995] and decision tree [Quinlan 1993].

*3.1.2. Support Vector Machine-Based CSI Model.* The basic idea of the support vector machine is to map the training samples into a high-dimension space as training data points, and then find out a set of hyperplanes to divide the training data points of several categories. Given  $K$  features and  $n$  training samples with category labels 1 (unstable status) or -1 (stable status), we map each training sample into a training data point in a  $K$ -dimension space. To be specific, if a training sample has the value  $v$  for the  $k$ -th feature, the corresponding mapped data point has the value  $v$  in the  $k$ -th dimension. Then the objective is to learn a hyperplane  $y = \mathbf{w} \cdot \mathbf{x} - b$ , where  $\mathbf{w}$  denotes a  $K$ -dimension vector and  $\mathbf{x}$  indicates a  $K$ -dimension data point. The optimization problem is as follows. We have

$$\min_{\mathbf{w}, \xi} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i \right\},$$

subject to

$$c_i(\mathbf{w} \cdot x_i - b) \geq 1 - \xi_i,$$

where  $C$  is a constant value,  $\xi_i$  denotes the degree of misclassification of the  $i$ -th training data point  $x_i$ , and  $c_i$  indicates the category label of  $x_i$ .

There exist several specialized algorithms for quickly solving the optimization problem, mostly reliant on heuristics for breaking the problem down into smaller, more manageable chunks. In our work, we use the SMO-type decomposition methods [Fan et al. 2005] to build CSI models.

*3.1.3. Decision Tree-Based CSI Model.* It is desirable to build SSI models for mobile users in their mobile devices instead of a remote server for the sake of privacy control. In this case, we have to take into account the complexity of training an SSI model because the computation resources of mobile device are usually limited. Therefore, we

also explore to build CSI models with decision trees, which are also successfully applied to classification but have less computation cost than the support vector machine. In our problem, we need a decision tree to determine the mostly possible status of a heavy-duty sensor according to the outputs of basic and light-duty sensors where each node of the tree is a decision rule to determine the mostly possible status of the heavy-duty sensor's output by taking into account one feature.

There exist several specialized algorithms for building decision trees. In our implementation, we use CART [Breiman et al. 1984] and adopt a binary recursive partitioning technology. Our experiments on real datasets show that the trained decision trees have limited degrees and can be stored in mobile devices with a proper threshold of the target accuracy.

### 3.2. Status Interval Inference Models

**3.2.1. Overview of SII Models.** Given a variable  $y$  for the heavy-duty sensor  $s$  and a feature vector  $\mathbf{x}$  extracted from the outputs of basic and light-duty sensors, an SII model  $\mathcal{F}_s$  can be formally defined as a function  $y = \mathcal{F}_s(\mathbf{x})$ , where  $y$  indicates the number of intervals before the next unstable status of  $s$ . An SII model  $\mathcal{F}_s$  infers the possible intervals between the current time point and the time point when the next unstable status of  $s$  occurs according to the current outputs of basic and light-duty sensors. Specially, if  $\mathcal{F}_s$  infers  $s$  will be most likely in an unstable status after  $n$  time points, the system will continuously use the latest invoked real output of  $s$  to estimate the output of  $s$  until  $t_{i+n}$  time point. Then the system invokes  $s$  and uses  $\mathcal{F}_s$  to infer the number of intervals before the next unstable status again. The advantage of SII models is that they don't need to infer at each time point and then reduce the system computation cost. For example, considering the motivating example that Sam's cell ID is 2344 (indicates his office) and the time is near PM3:00, a well-trained SII model may be able to infer that his GPS coordinate will change after two hours (the time of going off work) and the model can idle until that time.

The training of an SII model  $\mathcal{F}_s$  can be transformed to training a regression model from a training dataset where each training sample contains a number of intervals before the next unstable status for  $s$  with a feature vector extracted from the outputs of basic and light-duty sensors. To prepare such a training dataset, similar to preparing the training data for CSI models, we firstly transform the raw context records to a context record sequence by sorting them in the order of timestamps. Then we assign each context record a stable status interval number for  $s$  by checking the output sequence of  $s$ . Finally, for each context record, we extract the feature vector from the outputs of basic and light-duty sensors and the stable status interval number to build a training sample.

**3.2.2. Multiple Linear Regression-Based SII Model.** By extracting  $K$  features from the training samples as explanatory variables, we can implement an SII model by training a multiple linear regression [Williamus 2002] model in the following form. We have

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \varepsilon, \quad (1)$$

where  $\mathbf{y} = (y_1, y_2, \dots, y_n)$  denotes the vector of stable status intervals for each training sample,  $\mathbf{X} = (\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_n^T)$  denotes a matrix which consists of  $n$  mapped  $K$ -dimension training vectors  $\mathbf{x}_i$  ( $1 \leq i \leq n$ ),  $\mathbf{w}$  denotes a  $K$ -dimension vector, and  $\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)$  denotes the error term vector.

Generally speaking, given  $\mathbf{X}$ ,  $\varepsilon$  is assumed to follow a normal distribution with the zero mean, that is,  $\varepsilon|X \sim N[0, \sigma^2 I]$ , and the objective is to estimate  $\mathbf{w}$  and  $\sigma$ . It is easy

to prove that  $\omega = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$  is an unbiased estimation of  $\mathbf{w}$ . Moreover, an unbiased estimator of  $\sigma$  is

$$s = \sqrt{\frac{\mathbf{e}^T \mathbf{e}}{n - K}}, \quad (2)$$

where  $\mathbf{e} = \mathbf{y} - \mathbf{X}\omega$ .

Given a new feature vector  $\mathbf{x}_0$ , the inferred stable status interval of  $s$  must fall in the following range with a confidence  $\alpha$ . We have

$$\mathbf{x}_0^T \omega \pm \delta,$$

where  $\delta = t_{\alpha/2, n-K} s \sqrt{\mathbf{x}_0^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}_0}$ .

The inferred interval range may be not bounded by integers. Thus, to map the interval into the discrete time points, we determine the practical interval range as  $[\lfloor \mathbf{x}_0^T \omega - \delta \rfloor, \lceil \mathbf{x}_0^T \omega + \delta \rceil]$ .

**3.2.3. Principle Component Regression-Based SII Model.** As is well known, the general multiple linear regression approach may encounter the multicollinearity problem [Jolliffe 2002] which means that some explanatory variables are highly correlated. Consequently, the estimation of regression coefficients will be unstable and may mislead the estimation of the regression formulas. There are two widely used approaches for solving the multicollinearity problem. The first one is the stepwise regression [Derksen and Keselman 1992] which selects a subset of explanatory variables which are not highly correlated for the regression analysis and then avoid the multicollinearity problem. The drawback of this approach is that it may miss some useful information captured by the omitted explanatory variables. The second one is the Principle Components Regression (PCR) [Jolliffe 2002] which uses the principle components of explanatory variables instead of the raw explanatory variables for the regression analysis. As principle components are uncorrelated, the multicollinearity problem can be avoided while all related information for regression are maintained as much as possible. Our implementation takes advantage of the PCR approach to solve the multicollinearity problem. To be specific, we first express the training samples in the form of weighted vectors of principle components by the matrix  $\mathbf{Z}$  as follows. We have

$$\mathbf{Z} = \mathbf{X}\mathbf{A},$$

where the  $(i, k)$ -th element of  $\mathbf{Z}$  indicates the weight of the  $i$ -th training sample for the  $k$ -th principle components,  $\mathbf{A}$  is a  $K \times K$  matrix whose  $k$ -th column is the  $k$ -th eigenvector of  $\mathbf{X}^T \mathbf{X}$ , that is, a principle component. Therefore, the new form of the regression problem becomes

$$\mathbf{y} = \mathbf{Z}\gamma + \varepsilon,$$

where  $\gamma$  is a  $K$ -dimension vector and one of its unbiased estimations is  $\hat{\gamma} = (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{y}$ .

In the inference stage, the principle component regression-based SII model also needs to transform the feature vector generated from the outputs of basic and light-duty sensors  $\mathbf{x}_0$  to  $\mathbf{z}_0$ , where  $\mathbf{z}_0^T = \mathbf{x}_0^T \mathbf{A}$ . Then the stable status interval is inferred to fall into the following range with a confidence  $\alpha$ .

$$\mathbf{z}_0^T \hat{\gamma} \pm t_{\alpha/2, n-K} s \sqrt{\mathbf{z}_0^T (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{z}_0}$$

Similar to the inference of a general multiple regression-based SII model, the inferred range of stable status intervals should also be mapped into discrete time points.

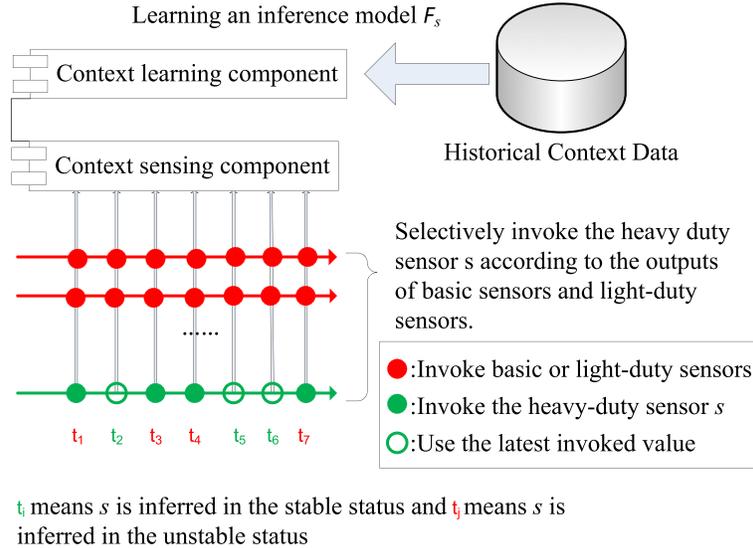


Fig. 4. The framework of a learning-based context-sensing system.

## 4. LEARNING-BASED CONTEXT-SENSING SYSTEMS

### 4.1. Framework

The context-sensing system is a middleware in mobile devices which provides unified APIs of context information to context-aware applications. We have three assumptions for a learning-based context-sensing system as follows. First, the sampling intervals of all light-duty sensors and basic sensors are the same. Second, the sampling intervals of heavy-duty sensors are integer multiples of that of the light-duty sensors and basic sensors. Last, the first invocations for all context sensors are synchronous when the system starts. These assumptions are reasonable for a learning context-sensing system because it needs as many outputs of basic and light-duty sensors as possible at a particular time point to infer the status of a heavy-duty sensor. It is worth noting that a context-aware application can avoid using the context-sensing system and directly invoke a context sensor with different sampling intervals if it has a special need. But it is beyond the scope of the context-sensing system so we do not discuss more details for this problem.

The framework of a learning-based context-sensing system is shown in Figure 4. The figure only illustrates the case of one heavy-duty sensor, but the framework can be easily generalized to the case of multiple heavy-duty sensors. From the figure we can see that the system includes two components, namely, the context learning component and the context-sensing component. The former is used for learning an SSI model and the latter is used to control context-sensing with the guidance of the learnt model.

The learning-based context-sensing system works as follows. Firstly, for each heavy-duty sensor  $s$ , the system collects the outputs of all context sensors with the sampling interval of  $s$  spanning for a predefined period, such as one week. The collected data are organized as a set of context records with timestamps where each context record contains the outputs of both  $s$  and basic and light-duty sensors.

Secondly, for each heavy-duty sensor  $s$ , the context learning component learns an SSI model  $F_s$  from the collected context records of the mobile user. The model  $F_s$  reflects the relationship between  $s$  and basic/light-duty sensors for the particular user.

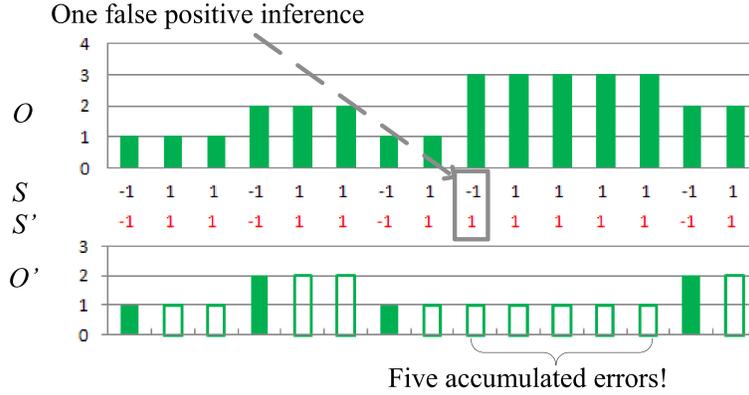


Fig. 5. An example of the error accumulation caused by a false positive inference. The system selectively invokes the audio-level sensor and reads the real output (filled bar) according to the inference that that sensor is probably in an unstable status ( $-1$ ). Otherwise ( $1$ ), it uses the latest invoked output of the audio-level sensor (blank bar).

Last, the context-sensing component selectively invokes the heavy-duty sensor  $s$  if the model  $\mathcal{F}_s$  infers that  $s$  is in an unstable status according to the outputs of basic and light-duty sensors. Otherwise, it provides the latest updated real output of  $s$  to context-aware applications as the estimation of  $s$ 's output at the current time point.

#### 4.2. Reducing the Errors of Estimations

A learning-based context-sensing system may have two kinds of inference errors. One is falsely inferring that a heavy-duty sensor  $s$  is in an unstable status, which is denoted as *false negative inference* for short. The other one is falsely inferring that a heavy-duty sensor  $s$  is in a stable status, which is denoted as *false positive inference* for short. If the system makes a false negative inference for a heavy-duty sensor  $s$ , we won't lose the real output of  $s$  at that time point. The only penalty is the unnecessary energy consumption which could be avoided by a correct inference on the status of  $s$ . However, if the system makes a false positive inference for  $s$ , the real output of  $s$  at that time point will be lost. What is worse, the errors of the estimated outputs may accumulate and thus cause more contextual information loss.

For example, Figure 5 illustrates the error accumulation caused by a false positive inference on an audio-level sensor. The outputs of the audio-level sensor fall into the range of  $[1,5]$  where the higher the background audio level, the larger the output. In the figure, both the sequence of real outputs (denoted as  $O$ ) and the sequence of estimated outputs (denoted as  $O'$ ) are shown. The corresponding real stable status sequence (denoted as  $S$ ) and the inferred stable status sequence (denoted as  $S'$ ) are also shown for the ease of understanding how the system works. From the figure we can see that the false positive inference occurring at the ninth time point causes five errors of the estimated outputs.

To this end, we introduce the maximum update interval  $T_{max}$  to control the time how long the system invokes the heavy-duty sensors mandatorily for reducing the probable accumulated errors in their estimations. To be specific, if the system has not invoked a heavy-duty sensor  $s$  within  $T_{max}$ , it will mandatorily invoke  $s$  and read the real output to reduce the risk of accumulated errors. Take the case in Figure 5 for example, if we set  $T_{max}$  to be three sampling intervals, when the false positive inference happens, the system does not always estimate the output of  $s$  by the value read at the seventh time point. Instead, it mandatorily invokes the audio-level sensor when it has not invoked

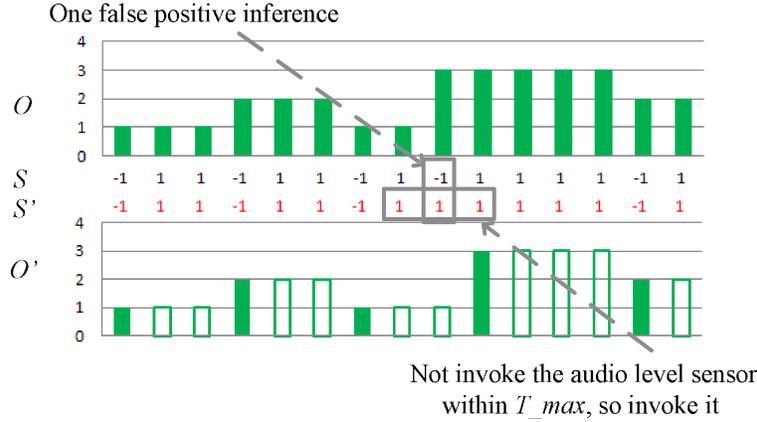


Fig. 6. An example of reducing the accumulated errors of the estimated outputs through  $T_{max}$  where  $T_{max} = 3$ .

the sensor within three sampling intervals. Thus, four accumulated errors are avoided. Figure 6 illustrates how  $T_{max}$  reduces the accumulated errors of the estimated outputs.

## 5. EXPERIMENTS

To evaluate the effectiveness and efficiency of the proposed approach for context-sensing, we conduct extensive experiments on real datasets for the GPS and audio-level sensors. In this section, the detailed experimental results are reported and discussed.

### 5.1. GPS Sensing

GPS sensors are major high energy-consuming context sensors and widely used by many context-aware applications. In this section, we evaluate several implementations of the proposed learning-based context-sensing approach for GPS sensing.

The evaluation is conducted on the context data collected from mobile devices but not directly conducted on mobile devices. It is because the main objective of this article is to explore the feasibility of applying machine learning technologies to energy-efficient context-sensing and comprehensively evaluate the performance of varying implementations in terms of complexity, sensing accuracy, and energy efficiency. Conducting the evaluation on benchmark datasets but not on real mobile devices makes our experiments repeatable.

**5.1.1. Datasets.** We collect 10 college volunteers' context data through their smart phones spanning for one month. The collected data include both indoor data and outdoor data, and the geographical area of the data collection basically covers the three campuses of the university. All of the smart phones are installed with the Symbian60v3 operating system and equipped with multiple sensors such as GPS sensors, application monitoring sensors, and 3D accelerometers. The collected data include rich types of context data listed in Table I. Among the basic and light-duty sensors used in the experiments, only the battery-level sensor and the inactive time sensor have numeric outputs. However, we convert outputs of the time sensor to numeric values for reflecting the continuity property of the time information though their raw format are strings. Specially, we convert the time  $hh : mm$  to an integer as  $hh \times 60 + mm$ .

In the data collection, the sampling intervals of all sensors are set to be one minute. It is worth noting that although we set a unified sampling interval, the numbers of

Table I. The Collected Context Data

Sensor type	Context	Value range
Basic	Day name	Monday, Tuesday, ... , Sunday
	Time	AM0:00, AM0:01,..., PM24:00
	Battery level	Percentage
	Profile type	General, Silent, Meeting, Outdoor
	Cell ID	Strings.
	Area ID	Strings.
light-duty	Inactive time	Integers (minute).
	Running Apps	Web, Message, Music, ..., Game
heavy-duty	GPS coordinate	Pair of longitude and latitude.
	Accelerometer	Triple of accelerating parameters.

Table II. The Details of Each Dataset

Owner ID	#Context records	#Features	Unstable ratio
A	8846	180	0.0797
B	27787	848	0.1776
C	25365	799	0.3560
D	23166	530	0.1530
E	30700	90	0.1045
F	19976	270	0.1398
G	25873	817	0.1054
H	25626	183	0.0795
I	21293	198	0.0840
J	22239	77	0.0403

context records collected from each volunteer may be different because volunteers can manually turn off the context data collection software in the case of low battery level. Moreover, some context records lack the GPS coordinates because the GPS sensor cannot find the satellite signal when the user is indoors. For such context records, we use the GPS coordinates occurring in the nearest context record as its GPS coordinates. This approach is reasonable because in this way SSI models take the indoor status as a stable status for the GPS sensor and do not invoke it in this case.

Table II shows the details of each volunteer's context dataset, where the *Owner ID* indicates the owner of the dataset, *#Context records* indicates the number of context records the dataset contains, *#Features* indicates the number of features extracted from the dataset by the method mentioned in Section 3, and *Unstable ratio* indicates the ratio of unstable statuses of the GPS sensor. We determine a GPS sensor is in an unstable status if and only if its output denotes the user has moved more than ten meters compared with the last GPS coordinates. Otherwise, we determine the GPS sensor is in a stable status.

*5.1.2. Evaluation.* The measurements of learning-based context-sensing systems for a heavy-duty sensor  $s$  should at least take into account two aspects, namely, the sensing accuracy and the energy efficiency. The sensing accuracy for  $s$  means the ratio of correct system outputs. The energy efficiency for  $s$  can be measured by the *Ratio of Reduced Invocations (RRI)* of  $s$ . The RRI of  $s$  is calculated by  $\frac{N-N_i}{N}$ , where  $N_i$  denotes the number of invocations of  $s$  and  $N$  denotes the number of test cases. Furthermore, for the sake of privacy protection, it is desirable to learn SSI models in device for

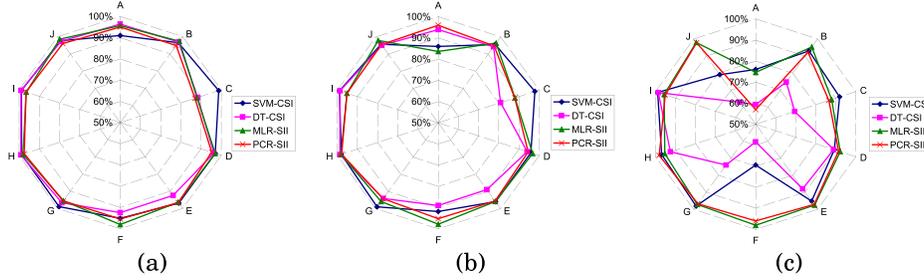


Fig. 7. Spherical comparison for GPS sensing accuracy with (a)  $T_{max} = 30\text{min}$ , (b)  $T_{max} = 60\text{min}$ , and (c)  $T_{max} = \infty$ .

avoiding the leak of the historical context data of mobile users. In this case, the computation cost of learning an SSI model is also taken into account as a measurement for learning-based context-sensing systems since the computation power of mobile devices is relatively limited compared with PCs.

We evaluate several implementations of the learning-based context-sensing approach with varying SSI models in terms of GPS sensing accuracy, energy efficiency, and the cost of training and storage. For CSI models, we implement the Support Vector Machine-based CSI (SVM-C SI) model and the Decision-Tree-based CSI (DT-C SI) model. For SII models, we implement the Multiple Linear Regression-based SII (MLR-S II) model and the Principle Component Regression-based SII (PCR-S II) model. Moreover, for each volunteer’s dataset, we use the data of the first three weeks as the training set and use the data of the last week as the test set.

The first baseline is Continuous GPS (CGPS) sensing which always invokes the GPS sensor with the same sampling interval. Since there is nonlearning-based energy context-sensing strategy reported by previous works, we use a nonlearning-based approach proposed in Abdesslem et al. [2009] as another baseline. The basic idea of this approach is to invoke the GPS sensor only when the 3D accelerometer indicates the mobile device is moving. For simplicity, we denote it as Accelerometer-based GPS (AGPS) sensing.

*Accuracy.* Similar to determining the unstable status, we take the estimated GPS coordinates correct if the distance error is less than ten meters. Figure 7 compares the GPS sensing accuracy of varying SSI models for each volunteer. From this figure we can see that with a proper  $T_{max}$ , the performance of all models for GPS sensing accuracy are comparable. For each volunteer’s context data, the GPS sensing accuracy of any SSI model is bigger than 90%, which implies that all SSI models are effective for GPS sensing. However, when the  $T_{max}$  is set to be  $\infty$ , which means that there is no threshold of the continuous estimations to reduce the accumulated estimation errors, the performance of the DT-C SI model for GPS sensing accuracy significantly drops while other SSI models still keep good performance. It implies that the DT-C SI model is more sensitive to the accumulated estimation errors. In other words, it is less robust than other models. We conduct experiments with other  $T_{max}$  settings. The experimental results show the GPS sensing accuracy of the DT-C SI is always good with a wide range of  $T_{max}$  settings (from tens to hundreds). The accuracy of both baselines is 100% for each volunteer because there is no uncertain factor for them. Learning-based approaches underperform the rule-based approach and the naive approach in terms of GPS sensing accuracy, but the difference is not big. Since the accuracies of the two baselines are same for each volunteers, we do not show their accuracy curves in Figure 7 for keeping its clarity.

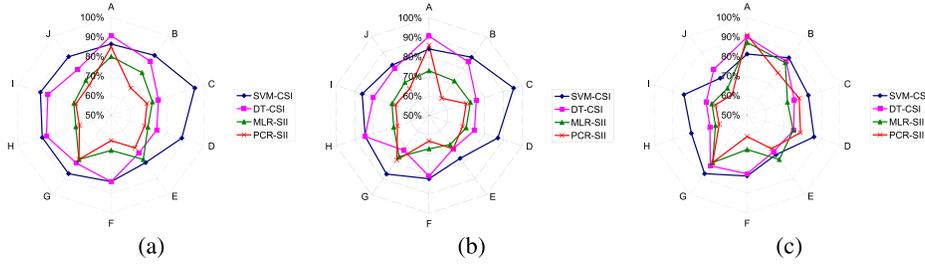


Fig. 8. Spherical comparison for GPS sensing accuracy when users are moving with (a)  $T_{max} = 30\text{min}$ , (b)  $T_{max} = 60\text{min}$ , and (c)  $T_{max} = \infty$ .

We also evaluate the GPS sensing accuracy of exact inference models. To be specific, we take advantage of the support vector machine to build a GPS coordinates classifier. As expected, the best accuracy on each dataset is less than 10%. It may be because the outputs of GPS are too diverse to be inferred according to the outputs of basic and light-duty sensors.

Sometimes the general GPS sensing accuracy cannot reflect the real effectiveness of GPS sensing due to the data imbalance. Extremely, when a user rarely moves, a model which always infers stable statuses can achieve a good accuracy though actually it has no ability of inference. Therefore, we also evaluate varying SSI models with respect to GPS sensing accuracy when users are moving, that is, the users' positions are continuously changing. In this case, the GPS sensor is always in an unstable status and it is more difficult to achieve a good accuracy. Figure 8 compares the GPS sensing accuracy of varying SSI models for each volunteer when users are moving. From this figure we can see that with proper  $T_{max}$ , SVM-CESI achieves the best GPS sensing accuracy and DT-CESI underperforms it a little. For each volunteer's context data, the GPS sensing accuracy of any SSI model is bigger than 70%. For SVM-CESI and DT-CESI, the index is even over 80% for most datasets.

We do not show the GPS sensing accuracy when the user stays at a location, that is, the GPS sensor is always in a stable status. It is because there will be no information loss if a stable status is not inferred correctly since the correct GPS result will still be output. The only effect is that the sensing system will need additional unnecessary invocations of the GPS sensor and consume more energy.

One may argue for another metric which evaluates the SSI models by comparing their ROC curves. Indeed, among the four proposed SSI models, it is only possible to generate ROC curves to evaluate the performances of SVM-CESI and DT-CESI for stable status inference since they are both classifiers of stable status. However, for the other two SII models, which are essentially regression models, it may not make sense to generate ROC curves. The reason is that these two SII models are not directly used for classifying the stable status of each test case. Instead, they are used for estimating the time interval  $T$  between two output changes of a heavy-duty sensor. Then,  $T$  is used for segmenting the sequence of test cases. In other words, all test cases during  $T$  time intervals are classified as stable status. Since the relationship between the precision of  $T$  and the precision of the classifying results is not clearly modeled, the ROC curves for  $T$  may not be proper to be used for evaluating the performance of these two SII models for stable status inference. As a result, we do not compare the ROC curves of four SSI models for evaluating their effectiveness for GPS sensing.

**Energy Efficiency.** The energy efficiency of GPS sensing is measured by the Ratio of Reduced Invocations (RRI) of the GPS sensor. Notice that in our experiments “invocation” is simulated by reading the real data of the test data. Obviously, the bigger the

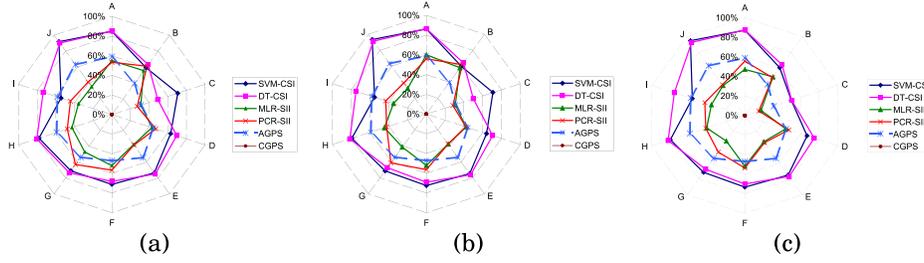


Fig. 9. Spherical comparison for energy efficiency in terms of RRI with (a)  $T_{max} = 30\text{min}$ , (b)  $T_{max} = 60\text{min}$ , and (c)  $T_{max} = \infty$ .

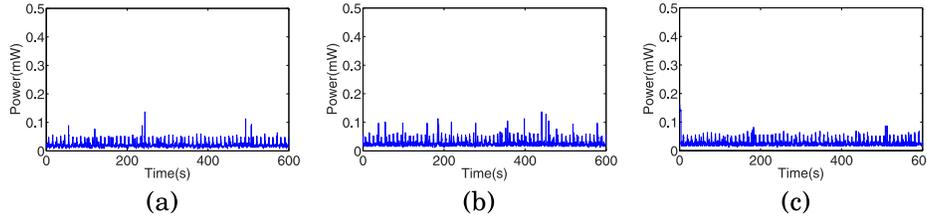


Fig. 10. Energy consumption of inference for (a) DT-CSI, (b) MLR-SII, and (c) PCR-SII.

RRI, the better the performance of the system in terms of energy efficiency. However, for the AGPS, we also need to take into account the energy consumption from the continuous invocations of the 3D accelerometer since its energy consumption more than 25% of that of the GPS sensor [Abdesslem et al. 2009] and cannot be omitted. Therefore, we calculate the RRI of AGPS by  $\alpha - 25\%$  to reflect its additional energy cost of GPS sensing, where  $\alpha$  is the raw RRI of AGPS.

Figure 9 compares the RRI of the two baselines and varying SSI models for each volunteer. From this figure we can see that the SVM-CSI model and the DT-CSI model consistently outperform other SSI models and the two baselines in terms of energy efficiency. Specially, they averagely reduce more than 70% invocations of GPS sensors for all volunteers.

One may argue that the reduction of invocations for heavy-duty sensors may not reduce the whole energy consumption because the inferences of SSI models need additional energy. Figure 10 shows the energy consumption of inference for varying SSI models in a Nokia N97 smart phone with 128M main memory. We do not show the energy consumption of inference for SVM-CSI because it needs too much memory to run. From the figure we can see that the energy consumption of inference for varying SSI models is very trivial compared with the invocations of heavy-duty sensors (Figure 3).

We also show the energy consumption of the same smart phone when no process is running (standby mode) in Figure 11. We can see that the energy consumption of inference for SSI models is comparable to standby mode.

**Training Cost.** Because the two baselines have no training cost, we only evaluate the cost of training varying SSI models. Figure 12(a) compares the time cost for training varying SSI models on a Core 2  $\times$  2.0G CPU, 2G main memory PC. The training algorithm of SVM-CSI is implemented by C++ based on a widely used SVM implementation libSVM [Chang and Lin 2001]. The training algorithm of DT-CSI is implemented by C++ based on CART [Steinberg and Colla 1997]. The training algorithms of MLR-SII and PCR-SII are implemented by MATLAB7.0. From the figure we can see that the time cost of training an SVM-CSI model is much larger than those of other

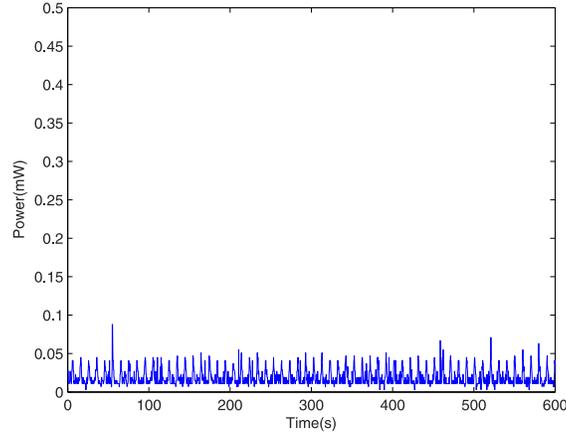


Fig. 11. The energy consumption of standby mode.

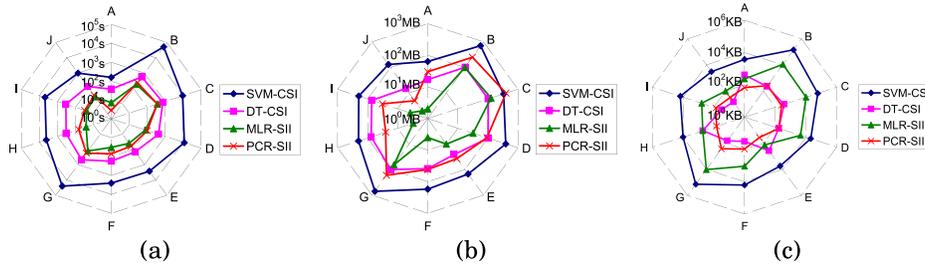


Fig. 12. Spherical comparison for (a) time cost of training, (b) memory requirement of training, and (c) storage cost.

SSI models. Considering in practice it needs to train multiple SVM-CSI models with different parameter settings, that is,  $C$  and then perform the model selection, the computation cost of training an SVM-CSI model is almost unacceptable for mobile devices. In contrast, the training of the MLR-SII model and the PCR-SII model is very efficient. The DT-CSI is also a good alternative since the training time is usually less than ten seconds.

Figure 12(b) compares the memory cost for training varying SSI models. From this figure we can see that the memory costs for training DT-CSI and PCR-SII are remarkably smaller than other SSI models, while the memory cost for training SVM-CSI is biggest.

Figure 12(c) compares the storage cost for varying SSI models. From this figure we can see that the storage costs of DT-CSI and PCR-SII are remarkably smaller than other SSI models. Furthermore, the SVM-CSI model needs the most space of storage.

We do not show the time cost of inference for varying SSI models because the computation cost of the inference stage is very trivial. All of the implemented SSI models can output the inference result less than 10 ms.

To comprehensively compare the performance of different SSI models and two baselines for GPS sensing, we list their performance in terms of different measurements in the Table III. For each method, the performance in terms of each measurement is categorized to “Good”, “OK”, or “Bad” by taking into account the comparison with others. It is worth noting that the two baselines are not considered in the comparisons

Table III. The Comprehensive Evaluation of Varying SSI Models for GPS Sensing

Method	Accuracy	RRI	Training cost	Storage cost
SVM-CSI	Good	Good	Bad	Bad
DT-CSI	Good	Good	Good	Good
MLR-SII	Good	Ok	Good	Ok
PCR-SII	Good	Ok	Good	Good
AGPS	Good	Ok	-	-
CGPS	Good	Bad	-	-

of training cost and storage cost because they are not learning-based approaches. The evaluation is conducted in the case that a proper  $T_{max}$  is set. From the table we can see that the performance of the DT-CSI model is best by taking into account all measurements.

## 5.2. Audio-Level Sensing

To study the generality of the proposed approach, we also evaluate it for another kind of heavy-duty sensors, that is, audio-level sensors. Audio-level sensors are used to provide the audio level of the background noise to some noisy-sensitive applications and its average energy consumption is more than 80% of the GPS sensor [Abdesslem et al. 2009].

*5.2.1. Dataset.* We use a public dataset named NokiaContextData [2004] as the experimental data for audio-level sensing. The data consists of rich type of context data of a volunteer spanning for 21 days including the time information, the running applications, the Cell ID, the area ID, and the environmental information such as pressure, humidity, audio level, and temperature. The context data collection is conducted by a smart phone, a sensor box, and a laptop PC when the user goes to his workplace or goes back home.

The audio level is expressed by 1–5 scores to differentiate the background noise level. In the experiments, we use the outputs of basic sensors including the time information and the Cell ID, and the area ID, and the output of the light-duty sensor, that is, the running applications to infer the status of the audio-level sensor. We extract 148 features from 127,698 context records for training SSI models and use the context data of the first two weeks as the training set. The remaining context data is used as the test set.

*5.2.2. Evaluation.* Similar to the experiments for GPS sensing, we evaluate the context-sensing system with varying SSI models in terms of sensing accuracy, energy efficiency (RRI), and training cost. Since there is no previous works for the strategies of energy-efficient audio-level sensing, we only use Continuous invocations of the Audio-Level (CAL) sensor as the baseline.

*Accuracy.* Figure 13 compares the audio-level sensing accuracy of varying SSI models. From this figure we can see that with a proper setting of the  $T_{max}$ , the performance of all models for audio-level sensing accuracy are acceptable since each SSI model's accuracy is over 85%. DT-CSI and SVM-CSI clearly outperform others with a proper  $T_{max}$  but underperform others without the constraint of  $T_{max}$ . The accuracy of the baseline CAL is obviously 100% so we do not show it in the figure.

Similar to the experiments for GPS sensing, we also evaluate the audio-level sensing accuracy of exact inference models. We build a support vector machine-based

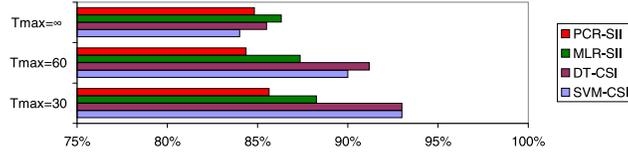


Fig. 13. The audio-level sensing accuracy of varying SSI models.

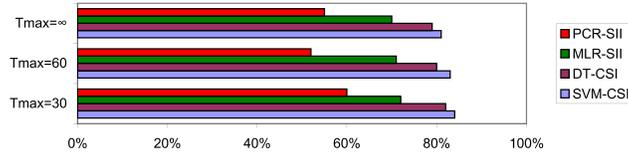


Fig. 14. The audio-level sensing accuracy of varying SSI models when the audio level changes.

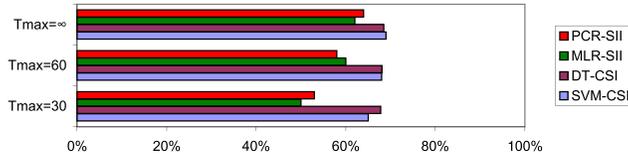


Fig. 15. The energy efficiency of the systems in terms of RRI for audio-level sensing with varying SSI models.

audio-level classifier and find that the best accuracy on each dataset is less than 4%, which implies that exact inference models are not proper for audio-level sensing.

Figure 14 compares the audio-level sensing accuracy of varying SSI models when the audio level really changes. From this figure we can see that with a proper setting of the  $T_{max}$ , the performance of all models for audio-level sensing accuracy are over 60% while DT-C SI and SVM-C SI's accuracy can reach 80%.

*Energy Efficiency.* The energy efficiency for audio-level sensing is also measured by RRI. Figure 15 compares the RRI of varying SSI models for each volunteer. From this figure we can see that the SVM-C SI model and the DT-C SI model dramatically outperform other SSI models in terms of energy efficiency. Both of them reduce more than 60% invocations of the audio-level sensor with varying  $T_{max}$  in the experiment. Moreover, all models at least reduce 50% real invocations of the audio-level sensor. The RRI of the baseline CAL is always zero so the figure does not show the value.

*Training Cost.* Because there is no additional training cost for the baseline, we only evaluate the cost of training varying SSI models. Figures 16(a), (b), and (c) compare varying SSI models for audio-level sensing in terms of training time cost, training memory requirements, and storage space, respectively. Similar to the experiments for GPS sensing, we can see that SVM-C SI needs the most computation cost and the largest storage space. DT-C SI outperforms the two SII models in terms of main memory requirements and space cost but underperforms them in terms of training time cost.

Table IV lists the performance of varying SSI models and the baseline on the audio-level sensor in terms of different measurements. The evaluation is conducted in the case a proper  $T_{max}$  is set. From the table we can see that the performance of the DT-C SI model is best by taking into account all measurements, similar to the conclusion in the experiments for GPS sensing.

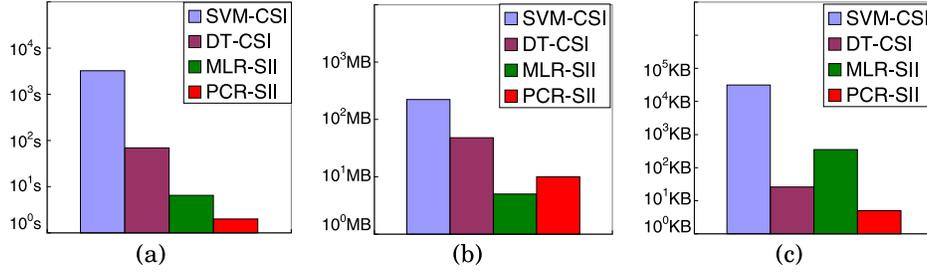


Fig. 16. (a) The training time cost, (b) the training memory requirement, and (c) the storage space of varying SSI models for audio-level sensing.

Table IV. The Comprehensive Evaluation of Varying SSI Models for Audio-Level Sensing

Method	Accuracy	RRI	Training cost	Storage cost
SVM-CSI	Good	Good	Bad	Bad
DT-CSI	Good	Good	Ok	Good
MLR-SII	Ok	Ok	Good	Ok
PCR-SII	Ok	Ok	Good	Good
CAL	Good	Bad	-	-

### 5.3. Discussion

The extensive experimental results of the proposed approach on GPS sensing and audio-level sensing show that learning-based context-sensing approaches, if properly used, can dramatically reduce the energy consumption of continuous sensing for multiple heavy-duty context sensors while keeping acceptable accuracies. Specially, for GPS sensing, we evaluate four learning-based context-sensing models, namely, SVM-CSI, DT-CSI, MLR-SII, and PCR-SII and find that they can more or less reduce the energy consumption of continuous GPS sensing with reasonable sensing accuracy. Moreover, among these approaches, SVM-CSI and DT-CSI dramatically outperform the other two and a baseline method with respect to saving the energy consumption of continuous GPS sensing. In fact, they averagely reduce more than 70% invocations of GPS sensors for all volunteers. However, the experimental results also show that the training and storage cost of SVM-CSI is expensive for mobile devices. To this end, DT-CSI can be regarded as the target method for continuous GPS sensing. For audio-level sensing, the experimental results show similar facts, in that SVM-CSI and DT-CSI dramatically outperform the other two learning-based context-sensing approaches with respect to saving the energy consumption of continuous audio-level sensing. Both of them reduce more than 60% invocations of the audio-level sensor in the experiment. Meanwhile, DT-CSI dramatically outperforms SVM-CSI with respect to training and storage cost.

## 6. RELATED WORK

### 6.1. Energy-Efficient Context-Sensing

The most related subject of this article is energy-efficient context-sensing. The problem of mobile context-aware computing has been studied for more than a decade. For example, Schilit et al. [1994] defined the problem of context-aware computing and introduced four categories of context-aware applications. Abowd et al. [1997] presented the Cyberguide project, in which prototypes of a mobile context-aware tour guide were built. Knowledge of the user's current location, as well as the history of past locations, were used to provide several kinds of services expected from a real tour guide.

Lemlouma and Layaida [2004] discussed the problem of content adaption for the mobile devices of clients, that is, client context-aware content adaption. The energy efficiency of context-sensing is crucial for the practical application of context-aware computing systems because the battery capacity has become a bottleneck compared with the fast evolution of the computation ability of mobile devices. Unfortunately, though the improvement of energy efficiency of sensors can directly address the problem, the technology of low-power sensors has not made significant progress in recent years.

Recently, some studies have been reported for improving the energy efficiency of context-sensing by a smart sensing strategy instead of more energy-efficient sensors. For example, for Wifi sensing, Chen et al. [2006] proposed an energy-efficient strategy for selection access points. For GPS sensing, Abdesslem et al. [2009] proposed to selectively invoke the GPS sensor by recognizing the user's movement through the 3D accelerometer. Constandache et al. [2009] proposed to dynamically select one sensor from the GPS sensor, the Wifi sensor, and the Cell ID sensor to provide location information by balancing the remaining energy and the accuracy. Some systems have integrated other heavy-duty sensors including accelerometers and Bluetooth beacons for GPS positioning [Kjærgaard et al. 2009; Paek et al. 2010; Zhuang et al. 2010]. Generally, these works can only apply to improving the energy efficiency of single sensors. Moreover, some of them need to invoke other heavy-duty sensors and then more or less counteract the improvement of energy efficiency. In contrast, the approach proposed in this article can be applied to multiple heavy-duty sensors without additional energy consumption.

## 6.2. Dynamic Energy Management

Another related field is dynamic energy management. This technique reduces power dissipation by selectively turning off (or reducing the performance of) system components when they are idle (or partially unexploited). These works can be grouped into two categories. One category is to study a general model of energy management. For example, Benini et al. [1999] and Paleologo et al. [1998] proposed power-managed systems using a finite-state, stochastic model. They took the problem of finding an optimal policy which maximizes the average performance level of a system, subject to a constraint on the power consumption, as a stochastic optimization problem. Moreover, Simunic et al. [2000] proposed the time-indexed SMDP model for deriving an optimal policy for dynamic energy management in portable systems. The model can be applied to portable devices, laptop hard disks and WLAN cards. Another category is to study the dynamic policy of energy management for specific components. For example, Garg et al. [2009] proposed a Markov-model-based disk idleness prediction scheme that can be used for reducing disk power consumption when used with a three speed disk. Our work borrows the idea of *selectively turning off some components* from these works and studies the problem of *selectively turning off some context sensors* for energy saving. However, we assume that the idleness of a heavy-duty sensor, that is, the time interval between two changes of its outputs, depends on the outputs of basic and light-duty sensors. By contrast, most works on dynamic energy management assume the idleness of each component is independent. Consequently, the technique of our work is totally different from the aforesaid works.

## 6.3. Learning-Based Sensing in Wireless Sensor Networks

The last related field is energy-saving sensing technology for wireless sensor networks. Smart sensing strategy-based energy-saving sensing is widely studied for wireless

sensor networks. These works are inspiring for studying smart sensing strategy-based energy-saving context-sensing approaches for mobile devices.

Since the energy consumption mainly comes from the data transmission, most of the literature focus on energy-efficient protocols of data transmission. For example, Heinzelman et al. [2000] proposed a clustering protocol named LEACH for periodical data gathering. LEACH is divided into rounds of two-phase operations. In the cluster formation phase, LEACH elects some cluster heads and the rest of the sensor nodes choose the proper cluster to join according to the signal strength from the cluster heads. In the data transmission phase, the cluster heads aggregate the data from their cluster members, and send the aggregated data to the base station by single-hop communication. Since the energy consumption of a sensor is highly related to the distance between the sensor and the data receiving node, the through energy consumption of a wireless sensor network with such a clustering protocol is far less than that with a naive “sending data to one center node” protocol. Several clustering protocols are proposed after LEACH for improving the energy efficiency further, such as HEED [Younis and Fahmy 2004], TPC [Choi et al. 2004], and ACE [Chan and Perrig 2008]. Moreover, Campobello et al. [2008, 2009] proposed a novel approach which splits the original messages in several packets such that each node in the network will forward only small subpackets. The simple splitting procedure is particularly helpful for those forwarding nodes that are more solicited than others, due to their position inside the network.

Though data-transmission-centric energy-saving protocols have been demonstrated successful in the application of wireless sensor networks, they cannot be applied to mobile context sensing because the energy consumption of data transmission for mobile context sensing is trivial compared with the energy consumption of the context sensors (e.g., 3D accelerometers, GPS sensors, microphones, etc.). In recent years, some learning-based wireless sensor network protocols have been proposed to directly control the sensors for saving energy. For example, a context-aware energy-saving sensor controlling approach is proposed in Chong et al. [2005]. This approach firstly learns context from the history sensor data and then controls the energy-saving functionalities of sensors according to the current context. Chong et al. [2008] proposed to use adaptive rule triggers on sensors to save energy of wireless sensor networks. Association rules between sensors are mined from the history sensor data and the outputs of some sensors are predicted instead of sampling according to the outputs of other sensors which are identified as highly related by the mined rules. Krause et al. [2009] proposed to optimize the placements and active time of multiple sensors for saving energy. These works inspire us to learn whether we can leverage machine learning technologies to control the context sensors of mobile devices for energy saving.

## 7. CONCLUSION AND FUTURE WORK

In this article, we proposed learning to infer the status of heavy-duty sensors for energy-efficient context-sensing. To be specific, we leveraged the historical context data of mobile users to train models which were used for inferring the statuses of heavy-duty sensors according to the outputs of basic and light-duty sensors. The experimental results on real datasets show that the proposed approach can improve the energy efficiency of GPS sensing and audio-level sensing for more than 70% and 60%, respectively, while the sensing accuracy is over 90%. Moreover, comprehensively considering the sensing accuracy, the energy efficiency, the training cost, and the storage cost, the decision tree-based implementation is the best choice for both sensors.

The robustness of SSI models with respect to training data is crucial for the practical application of the proposed approach. In the future, we will further study the sensitiveness of the proposed approach for the quality of the historical context data, such as the time length, the sparseness of the data.

## REFERENCES

- ABDESSLEM, F. B., PHILLIPS, A., AND HENDERSON, T. 2009. Less is more: Energy-Efficient mobile sensing with senseless. In *Proceedings of the 1st ACM Workshop on Networking, Systems, and Applications for Mobile Handhelds (MobiHeld)*. 61–62.
- ABOWD, G. D., ATKESON, C. G., HONG, J., LONG, S., KOOPER, R., AND PINKERTON, M. 1997. Cyberguide: A mobile context-aware tour guide. *Wirel. Netw.* 3, 5, 421–433.
- BENINI, L., BOGLIOLO, A., PALEOLOGO, G. A., AND DE MICHELI, G. 1999. Policy optimization for dynamic power management. *IEEE Trans. Comput.-Aid. Des. Integr. Circ. Syst.*, 182–187.
- BREIMAN, L., FRIEDMAN, J. H., STONE, C. J., AND OLSHEN, R. A. 1984. *Classification and Regression Trees*. CRC Press.
- CAMPOBELLO, G., LEONARDI, A., AND PALAZZO, S. 2008. Ace: An emergent algorithm for highly uniform cluster formation. In *Proceedings of the IEEE International Conference on Communications (ICC)*. 154–171.
- CAMPOBELLO, G., LEONARDI, A., AND PALAZZO, S. 2009. A novel reliable and energy-saving forwarding technique for wireless sensor networks. In *Proceedings of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. 269–278.
- CHAN, H. AND PERRIG, A. 2008. On the use of chinese remainder theorem for energy saving in wireless sensor networks. In *Proceedings of the 1st European Workshop on Sensor Networks (EWSN)*.
- CHANG, C. AND LIN, C. 2001. *LIBSVM: A Library for Support Vector Machines*. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- CHEN, Y., YANG, Q., AND YIN, J. 2006. Power-Efficient access point selection for indoor location estimation. *IEEE Trans. Knowl. Data Engin.* 18, 877–888.
- CHOI, W., SHAH, P., AND DAS, S. 2004. Framework for energy-saving data gathering using two-phase clustering in wireless sensor networks. In *Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous)*.
- CHONG, S. K., GABER, M. M., AND KRISHNASWAMY, S. 2005. A context-aware approach to conserving energy in wireless sensor networks. In *Proceedings of the 1st International Workshop on Sensor Networks and Systems for Pervasive Computing*.
- CHONG, S. K., GABER, M. M., AND KRISHNASWAMY, S. 2008. Arts: Adaptive rule triggers on sensors for energy conservation in applications using coarse-granularity data. In *Proceedings of the International Conference on Embedded Software and Systems (ICCESS)*. 314–321.
- CONSTANDACHE, I., GAONKAR, S., AND SAYLER, M. 2009. Enloc: Energy-Efficient localization for mobile phones. In *Proceedings of the INFOCOM Conference*. IEEE, 2716–2720.
- CORTES, C. AND VAPNIK, V. 1995. Support-vector networks. *Mach. Learn.* 20, 2, 273–297.
- DERKSEN, S. AND KESELMAN, H. J. 1992. Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables. *Brit. J. Math. Statist. Psychol.* 45, 265–282.
- FAN, R. E., CHEN, P. H., AND LIN, C. J. 2005. Working set selection using second order information for training svm. *J. Mach. Learn. Res.* 6, 1889–1918.
- FARRAHI, K. AND GATICA-PEREZ, D. 2011. Discovering routines from large-scale human locations using probabilistic topic models. *ACM Trans. Intell. Syst. Technol.* 2, 3:1–3:27.
- GARG, R., SON, S. W., KANDEMIR, M., RAGHAVAN, P., AND PRABHAKAR, R. 2009. Markov model based disk power management for data intensive workloads. In *Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID)*. 76–83.
- HEINZELMAN, W., CHANDRAKASAN, A., AND BALAKRISHNAN, H. 2000. Energy-Efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd International Conference on System Sciences (HICSS)*.
- JOLLIFFE, T. 2002. *Principal Component Analysis* 2nd Ed. Springer.
- KJÆRGAARD, M. B., LANGDAL, J., AND GODSK, T. 2009. Entracked: Energy-Efficient robust position tracking for mobile devices. In *Proceedings of the 7th Annual International Conference on Mobile Systems, Applications and Services (MobiSys)*. 221–234.
- KRAUSE, A., RAJAGOPAL, R., GUPTA, A., AND GUESTRIN, C. 2009. Simultaneous placement and scheduling of sensors. In *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*. 181–192.
- LEMLOUMA, T. AND LAYAÏDA, N. 2004. Context-aware adaptation for mobile devices. In *Proceedings of the 5th IEEE International Conference on Mobile Data Management (MDM)*. 106–111.
- NOKIACONTEXTDATA. 2004. [http://www.pervasive.jku.at/Research/Context\\_Database/](http://www.pervasive.jku.at/Research/Context_Database/).

- PAEK, J., KIM, J., AND GOVINDAN, R. 2010. Energy-efficient rate-adaptive gps-based positioning for smartphones. In *Proceedings of the 8th Annual International Conference on Mobile Systems, Applications and Services (MobiSys)*. 299–314.
- PALEOLOGO, G. A., BENINI, L., ET AL., A. B., AND MICHELI, G. D. 1998. Policy optimization for dynamic power management. In *Proceedings of the 35th Annual Design Automation Conference (DAC)*. 182–187.
- QUINLAN, J. R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann.
- SCHILIT, B., ADAMS, N., AND WANT, R. 1994. Context-aware computing applications. In *Proceedings of the Workshop on Mobile Computing Systems and Applications*. 85–90.
- SIMUNIC, T., BENINI, L., GLYNN, P., AND MICHELI, G. D. 2000. Dynamic power management for portable systems. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*. 11–19.
- STEINBERG, D. AND COLLA, P. 1997. Cart-classification and regression trees. In *Salford Systems*.
- WILLIAMUS, H. G. 2002. *Conometric Analysis* 5th Ed. Prentice Hall.
- YOUNIS, O. AND FAHMY, S. 2004. Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Trans. Mobile Comput.* 3, 366–379.
- ZHENG, Y. AND XIE, X. 2011. Learning travel recommendations from user-generated gps traces. *ACM Trans. Intell. Syst. Technol.* 2, 2:1–2:29.
- ZHUANG, Z., KIM, K., AND SINGH, J. P. 2010. Improving energy efficiency of location sensing on smartphones. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys)*. 315–330.

Received February 2011; revised July 2011; accepted October 2011