Sensus: A Cross-Platform, General-Purpose System for Mobile Crowdsensing in Human-Subject Studies

Haoyi Xiong, Yu Huang, Laura E. Barnes, and Matthew S. Gerber

{hx6d,yh3cf,lb3dp,msg8u}@virginia.edu

Department of Systems and Information Engineering, University of Virginia

Charlottesville, Virginia, USA

ABSTRACT

The burden of entry into mobile crowdsensing (MCS) is prohibitively high for human-subject researchers who lack a technical orientation. As a result, the benefits of MCS remain beyond the reach of research communities (e.g., psychologists) whose expertise in the study of human behavior might advance applications and understanding of MCS systems. This paper presents Sensus, a new MCS system for human-subject studies that bridges the gap between human-subject researchers and MCS methods. Sensus alleviates technical burdens with on-device, GUI-based design of sensing plans, simple and efficient distribution of sensing plans to study participants, and uniform participant experience across iOS and Android devices. Sensing plans support many hardware and software sensors, automatic deployment of sensor-triggered surveys, and double-blind assignment of participants within randomized controlled trials. Sensus offers these features to study designers without requiring knowledge of markup and programming languages. We demonstrate the feasibility of using Sensus within two human-subject studies, one in psychology and one in engineering. Feedback from non-technical users indicates that Sensus is an effective and low-burden system for MCS-based data collection and analysis.

Author Keywords

Crowdsensing; Participatory Sensing; Programmable Platform; Human Factors

ACM Classification Keywords

C.2.4 Distributed Systems: Distributed Applications

INTRODUCTION

Human-centric, participatory sensing [3] and mobile crowdsensing (MCS) [18] have facilitated participant recruitment and data collection for a range of human-subject studies. Instead of establishing data collection agreements in person with each participant and recording behaviors manually, MCS

researchers aim to recruit participants online and track behavioral and cognitive data automatically and remotely through apps running on mobile devices in the field.

Eagle and Pentland *et al.* [11, 13, 12] were among the first to use MCS to study human subjects. The authors developed the Reality mobile app to sense social interactions and mobility, and they recruited participants to install and run Reality on their mobile devices. By analyzing the collected data, the authors were able to investigate relationships between the structure of friendship networks and mobility patterns. Pentland's group extended this work to personality [7] and social ties [9], and others have used a similar approach to study mental and physical health relationships [29], connections between mental health and academic performance [39, 38], and human mobility and depression [4]. These and other studies indicate that MCS is a feasible paradigm for recruitment and data collection in human-subject research [10, 31].

The MCS paradigm has many potential benefits; however, heterogeneity of the mobile infrastructure and lack of general-purpose MCS tools present significant barriers to entry for human-subject researchers who lack a technical orientation. Existing MCS systems are predominantly designed for a single purpose, target only one of the major mobile platforms (Android or iOS), and require knowledge of markup and programming languages. This paper presents Sensus, a cross-platform, general-purpose system for MCS-based human-subject studies. Sensus lowers the barrier to entry into MCS-based research and increases the likelihood that MCS will be applied in a diverse set of fields. The remainder of this paper motivates Sensus, reviews related systems and research, presents technical details of Sensus, and presents two Sensus-based studies that demonstrate the flexibility and effectiveness of Sensus.

OPEN ISSUES FOR MCS IN HUMAN-SUBJECT STUDIES

Consider the following normative scenario, which motivates the use of MCS in human-subject research and highlights several open issues that we address in this paper.

Normative Scenario *A group of psychologists aims to better understand the relationship between students' social anxiety levels and telephone-based communication with family members and friends. The researchers have designed a survey to measure anxiety, and they would like to tag survey responses with location identifiers to help eliminate potential confounds*

Figure 1: Traditional participant recruitment flyers for psychology studies.

due to the physical setting. Ideally, the survey would be administered immediately after phone calls, and the study population would comprise geographically dispersed students from multiple campuses across the nation. In addition to survey and location data, the researchers would also like to provide participants with a wearable device that tracks physical activity levels, which might also correlate with social anxiety.

Our normative scenario highlights several challenges. First is the need to recruit and enroll a geographically dispersed study population (students across multiple campuses). Second is the need to administer traditional surveys in a context-sensitive manner (after telephone calls). Third is the need to integrate traditional survey data with location and activity data, which are difficult to elicit via self-report. In the following sections we argue that smartphone-based MCS could potentially address these issues for human-subject researchers if the current state of MCS technology were significantly advanced.

Participant Recruitment and Enrollment

Figure 1 shows human-subject research advertisements that are representative of those used in many psychological studies. The advertisements describe the proposed research, characteristics of desired participants, incentives, and contacts for the researchers. Researchers typically post advertisements like these in conspicuous places offline (e.g., outside of clinics or schools) and online (e.g., web forums). Participants must connect with the researchers via mail, telephone, or email to enroll in the study, which usually involves a person-to-person meeting between the participant and the researcher.

Researchers in our normative scenario would likely consider creating and distributing such advertisements; however, the level of participant effort required for enrollment as well as the costs associated with managing the enrollment process would be significant. An MCS-based alternative would allow individuals to enroll and participate in the study from their mobile devices without the need for human intervention on

the part on the researchers, thus expediting the study's start-up time, reducing enrollment effort, and removing barriers to participation due to physical proximity. However, these potential benefits must be balanced with the cost of developing the MCS system (e.g., a smartphone app) for multiple hardware and software platforms (e.g., iOS and various Android-based devices). Since socioeconomic and behavioral attributes correlate with the underlying platform [33], our hypothetical researchers would require an MCS system that targets the heterogeneous mobile infrastructure and nevertheless operates in a uniform way so as to remove confounds arising from participants' experience with the MCS. Such an MCS system does not currently exist, and developing one for each mobile platform would be an expensive endeavor.

Context-Sensitive Survey Administration

A traditional approach to our normative scenario might collect longitudinal data by asking participants to periodically complete online or offline surveys about social anxiety and other variables described in the scenario. Recall bias is an issue for such surveys, since survey administration is very likely to occur well after the target variable—telephone calls to family members—has been observed. Randomly scheduled experience sampling methods suffer from the same issue. The ideal solution would involve survey administration immediately after observation of the target variable. Our hypothetical researchers are targeting a young population for which smartphone ownership rates are high. The coexistence of telephony capabilities and flexible programming interfaces on these devices presents an opportunity for MCS-based technology to meet the researchers' needs; however, they will encounter the same barriers described above: Such an MCS app does not currently exist and would be expensive to develop.

Data Collection and Integration

Our normative scenario targets data that are much richer than traditional survey constructs. The experimental approach should accurately integrate survey response data with GPS-quality location data and physiological readings from consumer-grade wearable sensors. To date, each of these data collection requirements has been addressed separately. There are many free, web-based survey sites; all new smartphone devices have small, efficient GPS sensors; and wearable sensing technology for physiological monitoring has become commonplace. Effective as these individual solutions are, the costs associated with collection and integration would be significant for our hypothetical researchers, who would need to access, download, and integrate data from multiple vendors, each with its own proprietary set of commitments to data formatting, access, and anonymization.

Summary of Open Issues and Present Objectives

Current smartphone sensors, computational power, and market penetration are suited to human-subject investigations, but the practical reality is that only those with deep technical orientations or significant financial resources are currently able to leverage these tools effectively. This paper presents our effort to extend MCS methods to new disciplines, researchers, and study populations. Our objectives are to (1) target the heterogeneous mobile infrastructure while maintaining a uniform user

experience, (2) support a wide range of MCS-based studies that leverage sensing and interaction capabilities of modern smartphones, (3) eliminate the need for researchers to understand markup and programming languages when developing their MCS studies, and (4) implement the MCS system such that it relies exclusively on readily available mobile devices and cloud storage. To the best of our knowledge, this is the first such effort to be described.

RELATED WORK

This section reviews work bearing on the open issues described above. To preview our conclusion, we find that the MCS community has established many key design principles and technological components; however, these advances only apply to theoretical mobile systems, particular segments of the existing mobile infrastructure (iOS or Android but not both), and researchers with technical orientations.

MCS Tasks

Researchers have studied a variety of MCS tasks including participant recruitment [32, 5, 35, 42]; context and activity recognition [24]; modeling of location characteristics and linking such characteristics to user profiles [6, 20]; mapping network cells to geographic locations [16]; social interaction and collective behavior [30]; mobile object discovery in urban areas [40]; public transportation [25]; and energy and network consumption [23, 37, 36]. Sensus does not advance the state-of-the-art for these tasks; however, it does provide an open, cross-platform, researcher friendly test bed where these advanced techniques could be fielded for evaluation.

MCS Frameworks

Researchers have developed many MCS frameworks to support the tasks described above. Medusa supports prototyping for general-purpose MCS applications [28, 41], and [21, 14] support the implementation of context-awareness in MCS applications. ResearchKit for iOS[1] and ResearchStack for Android[2] facilitate the development of smartphone-based sensing applications. These frameworks lower the complexity of MCS application development; however, they require expertise with markup and programming languages (HTML, XML, JSON, Java, and Objective-C), and the resulting applications target iOS or Android but not both. Cross-platform solutions require manual conversion between frameworks or wholesale reimplementation. SensingKit addresses hardware-based sensors for Android and iOS [22]; however, it requires programming knowledge for each platform. These issues limit adoption by researchers and study participants. In contrast, Sensus achieves cross-platform operation without requiring markup or programming language knowledge.

MCS Applications

Many MCS applications target a particular research question rather than general usage. For example, StudentLife collects information about university students' mental health, academic performance, and behavioral trends [38]. In contrast, MyExperience was one of the first general-purpose MCS applications;

however, it runs on Windows Mobile devices (1% of current market share)[3] and requires knowledge of an XML-like scripting language for configuration [17]. The sensing capabilities of AndWellness are limited to location and activity type, and the app is confined to Android and requires XML knowledge for configuration [19]. Survalitics aims to be a general-purpose survey instrument, but it is confined to Android and requires manual modification of Java code [27]. Reporter is a commercial app ($3.99 per installation) for iOS that deploys scheduled surveys.[4] AWARE is similar to Sensus in many respects, but the iOS version currently requires manual compilation and deployment of source code [15]. Funf offers a comprehensive set of sensors and non-technical configuration; however, it lacks scheduled and sensor-triggered surveys, and it is only available for Android [1]. Paco allows researchers to design mobile sensing plans for iOS and Android devices without programming knowledge;[5] however, sensing in Paco is limited primarily to app- and call-based events, and this limitation carries over to surveys, which can only be triggered on the basis of these events.

The MCS applications above are useful for various purposes, but each one falls short in a significant way for human-subject researchers who are not technical experts. The following section presents our design requirements for Sensus in the context of these limitations.

SENSUS DESIGN REQUIREMENTS

We decompose the Sensus design requirements by stakeholder, considering the perspectives of human-subject researchers, study participants, and software developers.

Human-Subject Researchers

We engaged a group of University of Virginia psychologists when developing our researcher requirements. We observed that they require an MCS system that is generalizable to new research questions, has low cost, is compatible with mobile platforms that dominate the current market, implements a comprehensive set of configurable hardware and software sensors, supports scheduled and sensor-triggered surveys, does not require markup or programming language knowledge, supports simple study advertisement and enrollment via mobile-accessible distribution channels, secures and anonymizes participant data in compliance with Institutional Review Board regulations, and integrates with data analysis environments like R, MATLAB, and Tableau.

Study Participants

We engaged the University of Virginia Center for Survey Research[6] and the survey design guidelines described by Dillman *et al.* [8] when developing our participant requirements. We observed that study participants require an MCS system that eases enrollment in the study, protects participants from disclosure of personal information during the enrollment process, provides security and anonymity, and conserves power and network usage on mobile devices.

---

[1] http://www.apple.com/researchkit
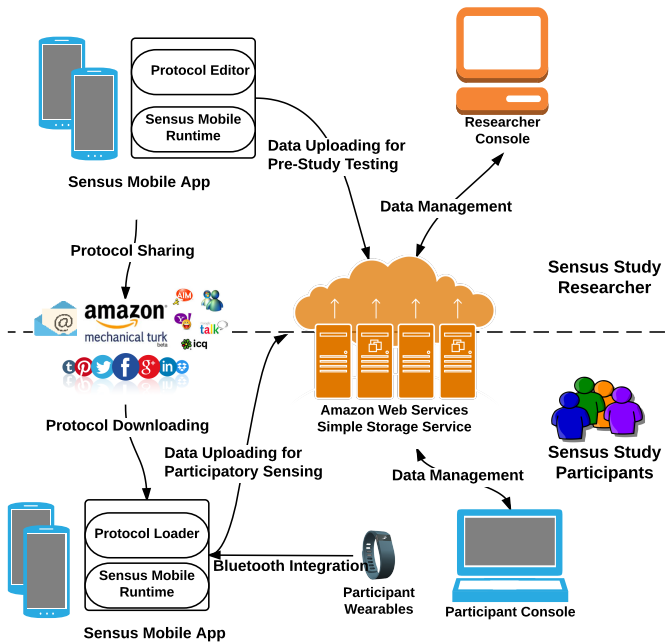
[2] http://researchstack.org

[3] As of May 2016: http://www.gartner.com/newsroom/id/3323017

[4] http://www.reporter-app.com

[5] https://www.pacoapp.com

[6] http://surveys.virginia.edu

Figure 2: High-level design of Sensus.

Figure 3: The Sensus mobile runtime.

Software Developers

Today's heterogeneous mobile infrastructure complicates the development of an MCS system that meets the above requirements. The mobile market is dominated by iOS and Android operating systems, which have different programming languages, development kits, and orientations toward security and application capabilities. These divergences require redundant design and implementation efforts targeting the same functional requirements. In order to provide value to human-subject researchers, MCS system developers require implementation convergence (e.g., shared source code) across mobile platforms.

The researcher, participant, and developer requirements described above apply broadly to those involved in MCS-based human-subject research. As far as we know, Sensus is the only MCS system that satisfies all of these requirements.

SENSUS DESIGN AND IMPLEMENTATION

Figure 2 presents the high-level design of Sensus from the perspectives of the researcher (top half) and the study participant (bottom half). These two perspectives coincide in the sharing of sensing plans—called *protocols*—and the storage of study data within Amazon Web Services (AWS) Simple Storage Service (S3). The end-to-end progression is summarized as follows: Researchers use the Sensus mobile app to configure a protocol, which is then disseminated to study participants as an encrypted JSON file. Each study participant receives the protocol file as an email attachment or URL, decrypts the protocol, and loads it into the Sensus mobile app on his or her device. Upon execution the protocol directs the collection and transmission of participant data, which are submitted to AWS S3 for retrieval and analysis by researchers. Additional detail
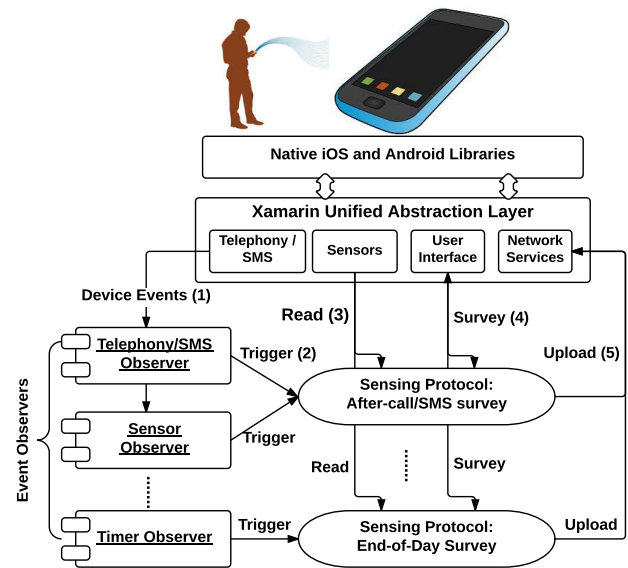
for each of these components is presented in the following sections.

Sensus Mobile App

The Sensus mobile app comprises the Sensus mobile runtime, the protocol editor, and the protocol loader.

Sensus Mobile Runtime

The Sensus mobile runtime (Figure 3) controls the underlying device according to the study protocol(s) that the participant has loaded into the Sensus mobile app. First, the runtime manages data probing. Table 1 presents the various data that can be probed within Android and iOS. Probes for most hardware-based sensors (e.g., altitude and compass heading) follow the observer pattern and are activated upon value changes. Probes have configurable data storage rates and several offer continuous (higher power consumption) and periodic (lower power consumption) sensing modes.

Second, the runtime administers researcher-designed surveys, which are formatted similarly to those found in online services such as SurveyMonkey.[7] The distinguishing feature of these surveys is that they can be deployed in response to any probed datum. Surveys can also be triggered during researcher-specified time intervals.

Third, the runtime anonymizes data collected from the probes and surveys. Each datum has fields that record the sensed data (e.g., degrees latitude/longitude for GPS data), time of sensing, and the protocol and device that generated the datum. Each field can be anonymized to varying degrees. The runtime supports field omission, one-way cryptographic hashing of text fields, placement of timestamp fields on random-anchored timelines, and rounding of floating-point fields. For example, the researcher might choose to anonymize all GPS data by anchoring the GPS timestamps to an arbitrary date and time in

---

[7]https://www.surveymonkey.com

	Probe	Android	iOS
Movement	3-axis acceleration	✓	✓
	Altitude	✓	✓
	Compass heading	✓	✓
	GPS coordinates	✓	✓
	POI proximity	✓	✓
	Horizontal speed	✓	✓
Social Media	Facebook profile	✓	✓
Communication	Phone call meta-data	✓	✓
	SMS messages	✓	
Environment	Light level	✓	
	Sound level	✓	✓
	External temperature	✓	
Device/Network	Battery level	✓	✓
	Bluetooth encounters	✓	
	Cell tower binding	✓	
	Screen on/off	✓	
	WLAN BSSID	✓	✓
User	HealthKit		✓
	Survey	✓	✓

Table 1: Sensus probes implemented for Android and iOS. POI refers to researcher-designated points of interest, HealthKit refers to the iOS data tracking backend, and Survey refers to surveys that are triggered on the basis of other probed data or researcher-defined schedules.

the past and rounding the coordinates to tenths, thus reducing spatial fidelity by approximately four miles.

Fourth, the runtime handles data storage. All probed data are temporarily stored locally either in RAM or in persistent storage. The choice of media is left to the researcher, as are timing options governing how long data are kept locally before being pushed to a remote data store. Upon activation, the remote data store collects all local data into a JSON file, securely transmits the JSON file to AWS S3, and purges the local data store. Energy- and data-efficient transmission strategies are supported by limiting remote data storage to occasions when the device is plugged in and/or connected to a Wi-Fi network.

To demonstrate the above operations, consider again the Sensus mobile runtime shown in Figure 3. The telephony system might emit an event indicating that a call has just ended (1). The telephony probe within the Sensus mobile runtime picks up this event, and the probe triggers an after-call survey designed by the researcher (2). The runtime reads the GPS sensor to geo-tag the triggered survey (3), the runtime displays the survey to the participant (4), and the participant's anonymized responses are uploaded to AWS S3 (5).

Protocol Editor
The Sensus mobile app serves dual purposes with respect to protocols. First, it manages the execution of protocols on participant devices as described above. Second, it supports a graphical protocol editor that researchers use to design protocols for distribution to study participants. Figure 4 shows the primary screens provided by the protocol editor. It takes only 14 taps—with no data entry or technical knowledge—to

configure a basic protocol that stores data from all probes to a local file on the device. Configuring remote storage of data in an AWS S3 bucket requires an AWS account and the information shown in Figure 4b.[8] Individual probes can be enabled and disabled with two taps each as shown in Figures 4c and 4d. Figure 4d also shows data rate and anonymization controls. All of these settings are described in a user manual.[9]

Sensus is able to administer surveys triggered by probed data or according to researcher-defined schedules. For example, the researchers in our normative scenario require self-report data on psychological affect (anxiety and positivity) throughout the day. The day is divided into seven time intervals 09:00–10:59, 11:00–13:59, ..., 21:00–22:59 (Figure 5a). Under this schedule the participant will receive the survey at a random time within each interval. The survey can contain a variety of inputs across multiple survey steps: number entries, pickers, sliders, text fields, and automatically recognized speech inputs (Figure 5b). Once selected for inclusion, each input is configured by the researcher (Figure 5c). When triggered, the survey is rendered for user interaction (Figure 5d). The full sequence of survey design operations shown in Figure 5 required 14 taps and four text entries. Each additional survey input would require a minimum of three taps and one text entry.

The protocol editor does not require any markup or programming knowledge; however, the researcher must make decisions regarding, for example, survey design, data anonymization, and data collection and storage rates. We document the options and their tradeoffs in the Sensus manual and provide graphical access to these settings in the protocol editor.

Protocol Loader
When the researcher is satisfied with the protocol, he or she locks it with a password to prevent modification by participants. The protocol is serialized to an encrypted JSON file that can only be opened within the Sensus app. This JSON file can be distributed to participant devices via email attachment, URL, or any other means of file transmission. The participant need only access the file from his or her mobile device and tap the file to copy it into the Sensus mobile runtime for decryption. A consent form is then displayed (Figure 6), and if the user consents the study begins immediately.

Given the range of open questions within MCS-based human-subject studies, researchers require a mechanism for conducting randomized controlled trials (RCTs) of different MCS configurations. Sensus supports RCTs by allowing researchers to bundle multiple protocols together into a single protocol file. Each protocol in the bundle may contain systematic variations in probe and survey configurations. These variations can be designed to test hypotheses, for example, regarding participant response rates to different survey schedules. Upon receipt of a bundled protocol, the consenting participant is double-blindly assigned to a random protocol variation (treatment group) and the study begins immediately.

---

[8]Sensus currently supports AWS S3 as the sole remote data store, and we provide a script that automates the configuration of the S3 bucket for use within Sensus.
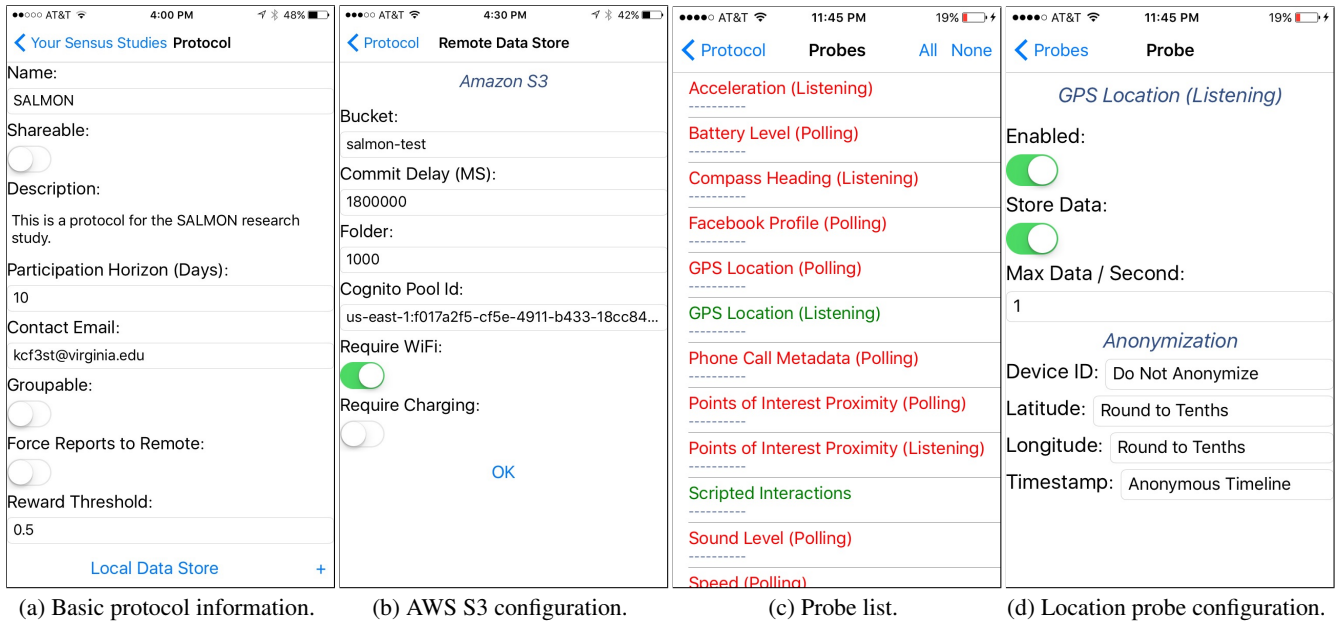
[9]https://github.com/predictive-technology-laboratory/sensus/wiki

(a) Basic protocol information.   (b) AWS S3 configuration.   (c) Probe list.   (d) Location probe configuration.

Figure 4: Protocol editor screens within Sensus. The depicted interaction sequence proceeds as follows: 4a, 4b, 4a, 4c, 4d.

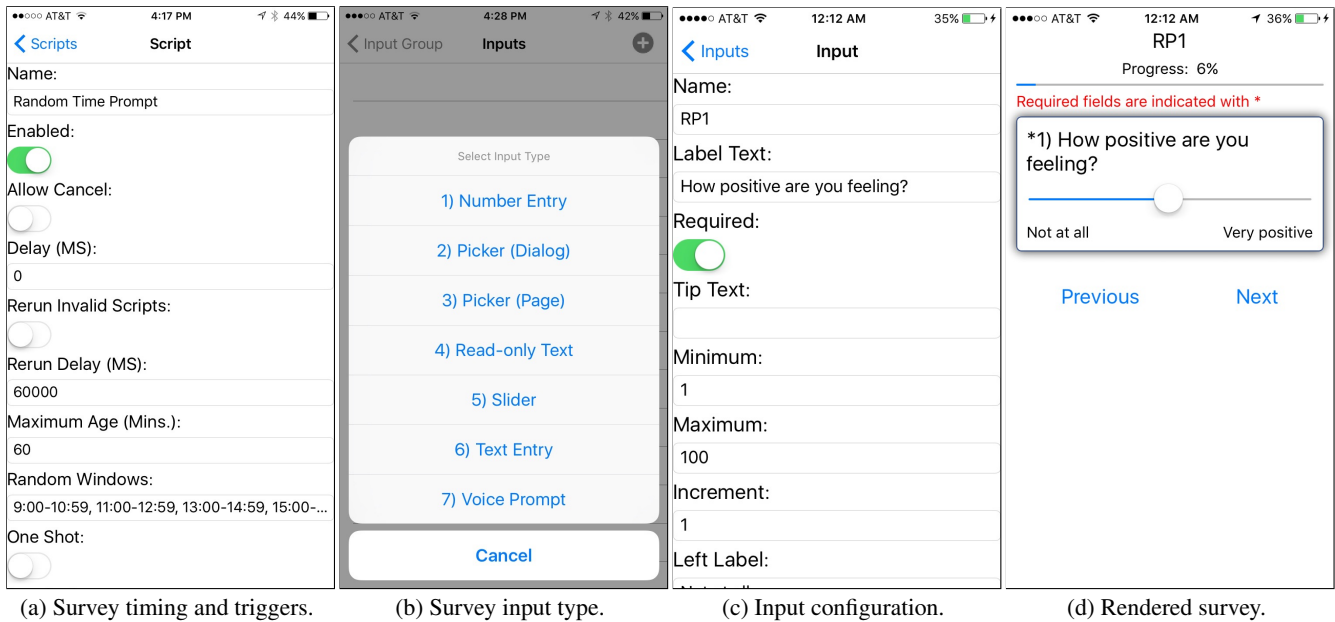(a) Survey timing and triggers.   (b) Survey input type.   (c) Input configuration.   (d) Rendered survey.

Figure 5: Survey editor being used to create a randomly timed survey of psychological affect measured with a slider.

Figure 6: Upon receipt of a protocol file, the participant copies it into the Sensus mobile runtime and proceeds through a consent screen showing a data collection schedule.

Sensus Implementation

Sensus is built on top of the Xamarin platform.[10] Conceptually, Xamarin creates an abstraction layer on top of the native iOS and Android operating systems, exposing a unified programming interface for most app functionality including file system interaction, user interface design and rendering, network communication, and many on-board sensors (e.g., GPS). The developer targets this unified programming interface with C# code, which runs within the Xamarin runtime on the device. The Xamarin runtime maps C# code to native Objective-C (for iOS) and Java (for Android) libraries. As a result, approximately 80% (20,000 lines) of code within Sensus, including the entire user interface, is shared between iOS and Android. The remaining functionality (8% iOS and 12% Android) comprises C# calls into non-shared iOS and Android libraries. The overall economy of our approach is significant, as most enhancements and fixes reside within the shared code base. All code is executed natively and the user interface is rendered with native elements, producing a look and experience that is familiar to iOS and Android users.

Analysis

Sensus data accumulate either locally on participants' devices or remotely within AWS S3. To support the researcher, we have implemented and released the SensusR package for the R Statistical Computing Environment.[11] This package provides methods for downloading Sensus data from AWS S3, parsing the JSON into R's native data frame format, and plotting time series graphs for each probe used in the study. Once in R's data frame format, the researcher can leverage the other packages in R's ecosystem to build statistical models of the data and test their hypotheses. Alternatively, the researcher can export Sensus data from R to comma-separated value (CSV) files for import into other environments like MATLAB, Tableau, and Microsoft Excel.

STUDY I: SOCIAL ANXIETY

We have deployed Sensus in two human-subject studies. The first involved a team of clinical psychologists, and the second involved a team of engineers. To preview our findings vis-à-vis Sensus, we observed that the psychologists were able to design effective Sensus protocols without training in markup languages and mobile development. Participants were successfully recruited and enrolled via email and online advertisements, and the teams were able to test hypotheses using the collected data. We provide additional detail on researchers' experience with Sensus at the end of each study description below.

The first study focused on social anxiety in a university student population. Social anxiety is the most common anxiety disorder among young adults [2], and little is known about how this population's social behaviors and networks contribute to the disorder's etiology and change following treatment. Prior studies have relied on retrospective reports of social functioning that are prone to bias in cases of prolonged recall periods [34]. A team of clinical psychologists used Sensus to study the effect of real-world social interactions on social anxiety. The team comprised one lead faculty member, one post-doctoral researcher, one Ph.D. student, and several undergraduate research assistants. The lead researcher had minimal technical and programming experience and had not previously conducted an MCS-based study. The team had no mobile development experience, although one individual had previously used the Palm Zire Personal Digital Assistant for experience sampling. Prior to the study, a subset of the students took a course entitled Big Data in Mental Health Research, which provides practical understanding of data science with R as it applies to mental health.

The researchers hypothesized that high (vs. low) social anxiety would be related to (1) more frequent and longer communication with close social contacts; (2) less frequent communication with non-family contacts; and (3) a higher ratio of outgoing calls to incoming calls. Further, the researchers hypothesized that high (vs. low) social anxiety would predict (4) relatively more negative and less positive emotions in social contexts, especially outside of the family; and (5) less social communication outside of the family following a

---

[10] https://www.xamarin.com

[11] https://github.com/predictive-technology-laboratory/sensus/wiki/SensusR

negative mood, perhaps suggesting decreased use of social support to regulate mood. The researchers recruited participants from undergraduate psychology courses and provided course credit as compensation. These students generally share many characteristics including life phase and psychosocial stressors, thereby eliminating many potential confounds. The Institutional Review Board at our university approved the deployment of Sensus for this study.

Study and Protocol Design
Researchers first used the Sensus protocol editor as described below.

Social Interaction Anxiety Scale Assessment
Baseline social anxiety levels of each participant were assessed via the Social Interaction Anxiety Scale (SIAS) [26] as a prerequisite for participating in the study and receiving compensation. People who suffer from social anxiety disorders typically have higher SIAS scores. The researchers implemented this survey using the survey editor (Figure 5), configuring the survey to deploy immediately following the consent screen (Figure 6).

After-Call Ecological Momentary Assessment (EMA)
The researchers also used the survey editor to create an EMA assessing participants' mood at the end of each telephone call. A sample of four key questions from this EMA are listed below.

Q1 How positive did you feel just before you received the call, on a scale of 0 (not at all positive) to 100 (very positive)?

Q2 How negative did you feel just before you received the call, on a scale of 0 (not at all negative) to 100 (very negative)?

Q3 How positive did you feel after the call, on a scale of 0 (not at all positive) to 100 (very positive)?

Q4 How negative did you feel after the call, on a scale of 0 (not at all negative) to 100 (very negative)?

This EMA was configured to deploy only when the participant finished an incoming or outgoing call. Participants had the option to skip this EMA without impacting their compensation.

Periodic GPS Location Tracking
In order to provide contextual information for the EMAs above, researchers configured the protocol to poll each participant's location every 5 minutes using GPS. This required negligible power, as the GPS sensor was not active during the inter-poll times. Sensus also supports continuous probe listening with and without listening while the device is asleep; however, the additional power consumption of these modes was considered too costly for this study.

Data Storage
In addition to configuring surveys, EMAs, and probes within the protocol, researchers configured an AWS S3 bucket for centralized storage of study data, specifying energy-saving upload strategies within the protocol editor (upload every 6 hours when connected to Wi-Fi).

Prior to the official start of the study, the researchers completed an internal test of Sensus and the above protocol among 10
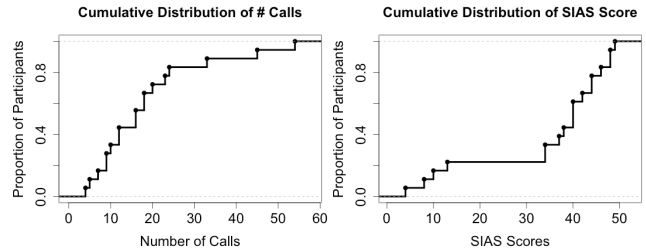

Figure 7: Cumulative distribution of participants' SIAS scores and total number of calls placed during Study I.

Measures	SIAS Pearson Correlation	p-Value
Number of calls	0.26	0.30
Q1 mean	-0.23	0.36
Q2 mean	-0.19	0.45
Q3 mean	0.13	0.60
Q4 mean	-0.40	0.10
Q1 standard deviation	0.08	0.76
Q2 standard deviation	-0.03	0.91
Q3 standard deviation	0.28	0.26
Q4 standard deviation	0.18	0.47

Table 2: Correlation and significance analysis for SIAS scores and after-call surveys.

Sensus development team members over a two-week period. This test revealed problems with the protocol loader on certain Android devices, and the problems were fixed before starting the study.

Participant Recruitment
The researchers recruited 10 undergraduate students from undergraduate psychology classes who received course credit for study participation. Another 10 participants were recruited from other departments via email and received monetary compensation. All participants downloaded Sensus v9.0.0 from the Android and iOS stores and received the study protocol via websites and emails.

Data Collection and Analysis
Data collection for this study lasted 10 days. Figure 7 shows participants' SIAS scores as well as the total number of calls they placed and received during the trial. After downloading the data from AWS S3, the psychologists calculated correlations between the SIAS scores and the after-call EMA data. Table 2 shows the results, where "Q1 mean" refers to the mean of the response to Question 1 and "Q1 standard deviation" refers to the standard deviation.

The researchers also correlated the SIAS EMA results with location data collected by the Sensus protocol. The location data for each participant were mapped to the following Foursquare venue types: Transportation, Religious, Work & Study, and Food & Leisure. Total time at each venue was estimated and correlated with participants' SIAS scores. Table 3 shows the results. The strongest correlation was found between SIAS scores and estimated time spent at religious

Locations	SIAS Pearson Correlation	p-Value
Work & Study	0.03	0.91
Transportation	0.11	0.66
Religious	-0.44	0.07
Food & Leisure	0.12	0.64

Table 3: Correlation and significance analysis for SIAS scores and time spent at different location types.

venues ($r = -0.44, p = 0.07$). Note that the researchers conducted data analysis using R and the SensusR package in this study.

Feedback from the Psychologists
At the conclusion of the study, we met with the psychologists to discuss their experience with Sensus. Overall, their feedback was positive and they indicated an interest in using Sensus for future studies. We summarize their feedback below.

Study Design
The psychologists reported a user-friendly and intuitive experience when designing the EMAs, configuring the location probe, and integrating with AWS S3. The GUI-based construction of EMA items was received positively, and since the survey questions were not very long it was not difficult to enter all survey design information through the mobile device. The psychologists indicated that a desktop or web-based protocol design tool would be useful for the construction of longer, more complex EMAs.

Participant Recruitment
The psychologists recruited participants by sharing the Sensus protocol file through email and via website URLs. The researchers appreciated flexibility in distributing the protocol files and suggested the creation of a web-based portal that participants could use to learn about and enroll in Sensus-based studies. This could improve researchers' ability to recruit participants, and it would lower the technical burden on participants.

Data Collection and Analysis
The researchers analyzed the collected data and obtained the results shown in Tables 2 and 3, all using the SensusR package and other packages in the R environment. The researchers reported that the current design of Sensus and SensusR satisfied their requirements vis-à-vis statistical analysis. They suggested that a real time, web-based dashboard and analysis platform would improve their ability to monitor in-progress studies and address issues earlier (e.g., participant drop-off and probe malfunction).

Intra-Study Modification of Sensus Protocols and Software
Throughout the study, the psychologists required fixes and enhancements of the Sensus protocol as well as the Sensus system itself. The psychologists could simply modify the Sensus protocol on their mobile devices and redistribute the protocol file to study participants. Changes to the Sensus software were more complicated, as they needed to be performed by Sensus developers and released through the Google Play
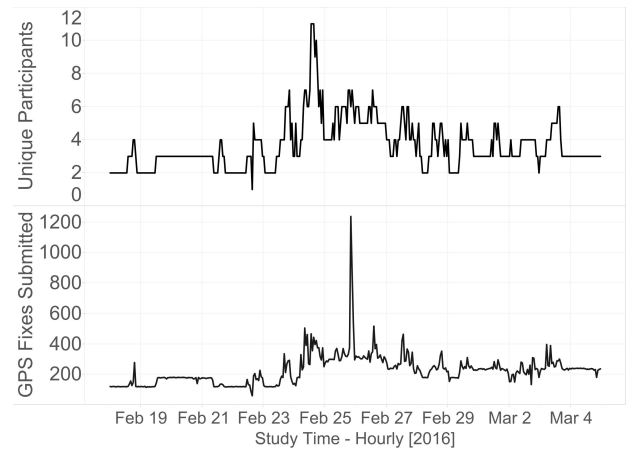

Figure 8: Number of unique participants (top series) and number of GPS location fixes submitted (bottom series) per study day by participants in Study II.

Store (for Android) and the iTunes App Store (for iOS). Android releases usually propagate within a few hours, whereas iOS releases can take a week or more to be approved.

STUDY II: LATE ARRIVALS
Late arrivals and missed appointments cause significant disruptions in many industries. These problems are particularly serious in public health, where patient no-shows lead to intentional overbooking, revenue loss, and wasted resources. In an effort to optimize the functioning of public health clinics, we are studying travel behavior, late arrivals, and missed appointments. We are currently in a pilot phase for this study, and we are focusing on the surrogate problem of analyzing late arrivals of students to their classes. The Institutional Review Board at our university has approved the deployment of Sensus to student devices for the purpose of collecting (1) high-accuracy location traces and (2) class schedule information (for identifying class locations and start times). It took approximately 1 hour to configure the Sensus protocol for this study, which we then distributed to consenting students. Over the course of a 15-day trial run, 22 participating students submitted 87,017 GPS location fixes. Figure 8 shows the per-day volume of unique participants as well as GPS location data submitted by the participants. Figure 9 maps the collected data. We are currently working to build statistical prediction models for the binary response of on-time arrival with predictors including each participant's current location, speed, mode of travel, and historical arrival latency, all of which can be probed within Sensus.

Our own experience with using Sensus in the above study has been positive. Configuration of the Sensus protocol was quick, and integration with AWS S3 is performed using a one-command script provided with Sensus. The most challenging aspect of this study has been participant recruitment. It was not difficult to enroll participants once they committed to the study; however, Sensus does not currently implement pre-study advertisement or in-app incentive mechanisms that might mitigate participant attrition. The latter is demonstrated
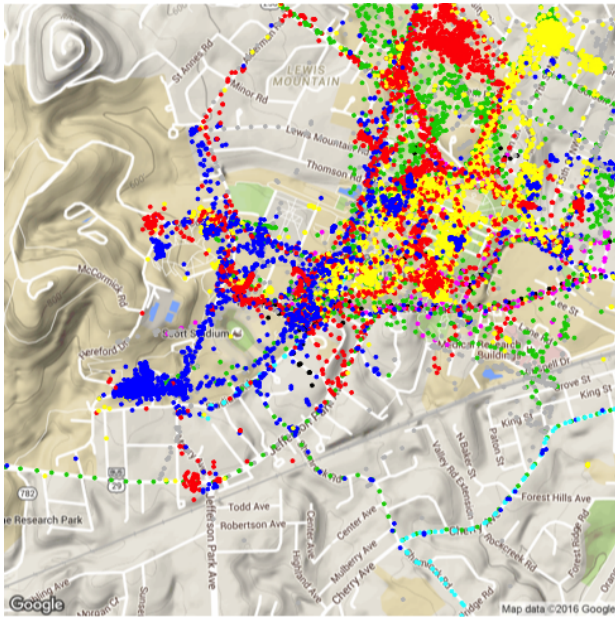
Figure 9: Location data collected in Study II. Each point represents a single GPS reading, and colors indicate the various participants. See the online version of this paper for color.

in Figure 8 with a sharp peak in unique participants followed by a steady decline.

DISCUSSION AND LIMITATIONS

Based on our experience using Sensus in human-subject studies, we believe that it is well suited to the needs of non-technical researchers as well as study participants and software developers. Using Sensus, the researcher is able to create a new protocol, configure data probes, establish a remote data store within AWS S3, and bind the protocol to the data store to begin uploading data. It is certainly the case that, even with GUI support, probe configuration can be time consuming. Polling rates require careful consideration, and survey construction is a research field in itself; however, Sensus strikes an effective balance in taking care of the technical issues that currently prevent many human-subject researchers from leveraging MCS. Probe tasking and survey construction decisions should be informed by the hypotheses under consideration and knowledge regarding anticipated participants. Sensus gives the researcher freedom to make these decisions while accelerating implementation once decisions have been reached.

Several issues and research questions remain. First, despite Xamarin's effort to provide a uniform experience for participants in Sensus-based studies, divergence between iOS and Android remains a source of variation depending on the probes used. For example, in Android it is possible for Sensus to wake itself up from a background state at a predetermined time, ask a probe to take a reading, and put itself back to sleep. This is desirable from the perspective of maintaining timing and power constraints on data collection. In contrast, iOS only permits such activations for a very limited set of background operations. Thus, the researcher might desire a regularly sched-

uled probe reading, but in many cases the only alternative on iOS is to display a notification to the user and hope that he or she will open the app so that the probe can be activated. This variation between Android and iOS data probing raises important questions. Might the resulting data bias hypothesis tests if the hypothesized variable correlates with user preferences for iOS or Android? How might the Android version of the protocol be modified to eliminate such bias? Second, the foundations of statistical hypothesis testing are rooted in assumptions about random sampling. Traditional surveys and experience sampling methods leverage random sampling of participants and response times to support generalization of conclusions beyond the sample population. As MCS matures and we are able to more precisely administer surveys using contextual sensing, how will the assumptions of our statistical tests be impacted?

In addition to the important questions raised above, the Sensus software would benefit from the addition of several capabilities that have been investigated within the MCS literature: Integrated incentive management, push-style participant tasking from a researcher-maintained node, and on-the-fly updating of participants' protocols are all natural requests of an MCS system, but they have not yet been implemented within Sensus. We are currently working on a web-based protocol editor and a portal for recruiting participants and visualizing study data in real time.

CONCLUSIONS

This paper has presented Sensus, an MCS system for human-subject studies that alleviates technical burdens for researchers who wish to leverage MCS methods. Sensus focuses on GUI-based design of sensing plans, eliminating the need for programming while at the same time targeting iOS and Android with a uniform user experience. Sensus simplifies the recruitment and enrollment of MCS participants by supporting the distribution of sensing plans via traditional means (e.g., email attachments and Internet webpage downloads). Participant enrollment involves minimal effort, and support is provided for conducting double-blind randomized controlled trials of sensing plan variations. We have reported two studies that are using Sensus. Our experience and that of non-technical psychologists indicates that Sensus is a flexible and low-burden means of applying MCS to the study of human-subject research questions. The source code for Sensus is available for free under an Apache v2.0 license, and Sensus is available for free installation from the Android and iOS app stores. More information can be found on our website.[12]

---

[12]https://github.com/predictive-technology-laboratory/sensus/wiki

REFERENCES

1. Nadav Aharony, Wei Pan, Cory Ip, Inas Khayal, and Alex Pentland. 2011. Social fMRI: Investigating and shaping social mechanisms in the real world. *Pervasive and Mobile Computing* 7, 6 (2011), 643–659.

2. American College Health Association and others. 2013. National College Health Assessment II: Reference Group Executive Summary Fall 2012. *Hanover, MD: American College Health Association* (2013).

3. Andrew T Campbell, Shane B Eisenman, Nicholas D Lane, Emiliano Miluzzo, Ronald A Peterson, Hong Lu, Xiao Zheng, Mirco Musolesi, Kristóf Fodor, and Gahng-Seop Ahn. 2008. The rise of people-centric sensing. *Internet Computing, IEEE* 12, 4 (2008), 12–21.

4. Luca Canzian and Mirco Musolesi. 2015. Trajectories of depression: unobtrusive monitoring of depressive states by means of smartphone mobility traces analysis. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 1293–1304.

5. Giuseppe Cardone, Luca Foschini, Paolo Bellavista, Antonio Corradi, Cristian Borcea, Manoop Talasila, and Reza Curtmola. 2013. Fostering participaction in smart cities: a geo-social crowdsensing platform. *Communications Magazine, IEEE* 51, 6 (2013), 112–119.

6. Yohan Chon, Nicholas D. Lane, Fan Li, Hojung Cha, and Feng Zhao. 2012. Automatically Characterizing Places with Opportunistic CrowdSensing using Smartphones. In *Proc. 14th Int. Conf. Ubiquitous Computing ACM*. 481–490.

7. Yves-Alexandre de Montjoye, Jordi Quoidbach, Florent Robic, and Alex Sandy Pentland. 2013. Predicting personality using novel mobile phone-based metrics. In *Social computing, behavioral-cultural modeling and prediction*. Springer, 48–55.

8. Don A Dillman, Jolene D Smyth, and Leah Melani Christian. 2014. *Internet, phone, mail, and mixed-mode surveys: the tailored design method*. John Wiley & Sons.

9. Wen Dong, Bruno Lepri, and Alex Sandy Pentland. 2011. Modeling the co-evolution of behaviors and social relationships using mobile phone data. In *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia*. ACM, 134–143.

10. Nathan Eagle and Alex Pentland. 2005. Social serendipity: Mobilizing social software. *Pervasive Computing, IEEE* 4, 2 (2005), 28–34.

11. Nathan Eagle and Alex Pentland. 2006. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing* 10, 4 (2006), 255–268.

12. Nathan Eagle and Alex Sandy Pentland. 2009. Eigenbehaviors: Identifying structure in routine. *Behavioral Ecology and Sociobiology* 63, 7 (2009), 1057–1066.

13. Nathan Eagle, Alex Sandy Pentland, and David Lazer. 2009. Inferring friendship network structure by using mobile phone data. *Proceedings of the national academy of sciences* 106, 36 (2009), 15274–15278.

14. Salma Elmalaki, Lucas Wanner, and Mani Srivastava. 2015. CAreDroid: Adaptation Framework for Android Context-Aware Applications. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 386–399.

15. Denzil Ferreira, Vassilis Kostakos, and Anind K Dey. 2015. AWARE: mobile context instrumentation framework. *Frontiers in ICT* 2 (2015), 6.

16. Michal Ficek, Nathaniel Clark, and Lukáš Kencl. 2012. Can crowdsensing beat dynamic cell-ID?. In *Proceedings of the Third International Workshop on Sensing Applications on Mobile Phones*. ACM, 10.

17. Jon Froehlich, Mike Y Chen, Sunny Consolvo, Beverly Harrison, and James A Landay. 2007. MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones. In *Proceedings of the 5th international conference on Mobile systems, applications and services*. ACM, 57–70.

18. R.K. Ganti, F. Ye, and H. Lei. 2011. Mobile Crowdsensing: Current state and Future Challenges. *IEEE Communications Magazine* 49 (2011), 32–39.

19. John Hicks, Nithya Ramanathan, Donnie Kim, Mohamad Monibi, Joshua Selsky, Mark Hansen, and Deborah Estrin. 2010. AndWellness: an open mobile system for activity and experience sampling. In *Wireless Health 2010*. ACM, 34–43.

20. Sibren Isaacman, Richard Becker, Ramón Cáceres, Stephen Kobourov, Margaret Martonosi, James Rowland, and Alexander Varshavsky. 2011. Identifying important places in people's lives from cellular network data. In *Pervasive Computing*. Springer, 133–151.

21. PP Jayaraman, A Sinha, W Sherchan, S Krishnaswamy, A Zaslavsky, P Delir Haghighi, S Loke, and M Thang Do. 2012. Here-n-Now: A Framework for Context-Aware Mobile Crowdsensing. In *Proceedings of the Tenth International Conference on Pervasive Computing, UK*.

22. Kleomenis Katevas, Hamed Haddadi, and Laurissa Tokarchuk. 2014. Poster: Sensingkit: A multi-platform mobile sensing framework for large-scale experiments. In *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 375–378.

23. Nicholas D Lane, Yohan Chon, Lin Zhou, Yongzhe Zhang, Fan Li, Dongwon Kim, Guanzhong Ding, Feng Zhao, and Hojung Cha. 2013. Piggyback CrowdSensing (PCS): energy efficient crowdsourcing of mobile sensor data by exploiting smartphone app opportunities. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. ACM, 7.

24. Nicholas D Lane, Ye Xu, Hong Lu, Shaohan Hu, Tanzeem Choudhury, Andrew T Campbell, and Feng Zhao. 2011. Enabling large-scale human activity inference on smartphones using community similarity networks (CSN). In *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 355–364.

25. Suhas Mathur, Tong Jin, Nikhil Kasturirangan, Janani Chandrasekaran, Wenzhi Xue, Marco Gruteser, and Wade Trappe. 2010. Parknet: drive-by sensing of road-side parking statistics. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 123–136.

26. Richard P Mattick and J Christopher Clarke. 1998. Development and validation of measures of social phobia scrutiny fear and social interaction anxiety. *Behaviour research and therapy* 36, 4 (1998), 455–470.

27. Vikas O'Reilly-Shah and Sean Mackey. 2016. Survalytics: An Open-Source Cloud-Integrated Experience Sampling, Survey, and Analytics and Metadata Collection Module for Android Operating System Apps. *JMIR mHealth and uHealth* 4, 2 (2016), e46.

28. Moo-Ryong Ra, Bin Liu, Tom F La Porta, and Ramesh Govindan. 2012. Medusa: A programming framework for crowd-sensing applications. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 337–350.

29. Mashfiqui Rabbi, Shahid Ali, Tanzeem Choudhury, and Ethan Berke. 2011. Passive and in-situ assessment of mental and physical well-being using mobile sensors. In *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 385–394.

30. K.K. Rachuri., C. Mascolo, M. Musolesi, and P.J. Rentfrow. 2011. Sociablesense: exploring the trade-offs of adaptive sampling and computation offloading for social sensing. In *Proceedings of MobiCom*. ACM, 73–84.

31. Mika Raento, Antti Oulasvirta, and Nathan Eagle. 2009. Smartphones an emerging tool for social scientists. *Sociological methods & research* 37, 3 (2009), 426–454.

32. Sasank Reddy, Deborah Estrin, and Mani Srivastava. 2010. Recruitment framework for participatory sensing data collections. In *Pervasive Computing*. Springer, 138–155.

33. Joel Ross, Lilly Irani, M Silberman, Andrew Zaldivar, and Bill Tomlinson. 2010. Who are the crowdworkers?: shifting demographics in mechanical turk. In *CHI'10 extended abstracts on Human factors in computing systems*. ACM, 2863–2872.

34. Saul Shiffman, Arthur A Stone, and Michael R Hufford. 2008. Ecological momentary assessment. *Annu. Rev. Clin. Psychol.* 4 (2008), 1–32.

35. Adish Singla and Andreas Krause. 2013. Incentives for privacy tradeoff in community sensing. In *First AAAI Conference on Human Computation and Crowdsourcing*. 165–173.

36. Leye Wang, Daqing Zhang, and Haoyi Xiong. 2013. effSense: energy-efficient and cost-effective data uploading in mobile crowdsensing. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM, 1075–1086.

37. Leye Wang, Daqing Zhang, Haoyi Xiong, John Paul Gibson, Chao Chen, and Bing Xie. 2016. ecoSense: Minimize Participants' Total 3G Data Cost in Mobile Crowdsensing Using Opportunistic Relays. *Systems, Man, and Cybernetics: Systems, IEEE Transactions on* (2016), 1–14.

38. Rui Wang, Fanglin Chen, Zhenyu Chen, Tianxing Li, Gabriella Harari, Stefanie Tignor, Xia Zhou, Dror Ben-Zeev, and Andrew T Campbell. 2014. Studentlife: assessing mental health, academic performance and behavioral trends of college students using smartphones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 3–14.

39. Rui Wang, Gabriella Harari, Peilin Hao, Xia Zhou, and Andrew T Campbell. 2015. SmartGPA: how smartphones can assess and predict academic performance of college students. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 295–306.

40. Harald Weinschrott, Julian Weisser, Frank Durr, and Kurt Rothermel. 2011. Participatory sensing algorithms for mobile object discovery in urban areas. In *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*. IEEE, 128–135.

41. Yu Xiao, Pieter Simoens, Padmanabhan Pillai, Kiryong Ha, and Mahadev Satyanarayanan. 2013. Lowering the barriers to large-scale mobile crowdsensing. In *Mobile Computing Systems and Applications*.

42. Haoyi Xiong, Daqing Zhang, Guanling Chen, Leye Wang, and Vincent Gauthier. 2015. CrowdTasker: Maximizing coverage quality in Piggyback Crowdsensing under budget constraint. In *Pervasive Computing and Communications (PerCom), 2015 IEEE International Conference on*. IEEE, 55–62.