MakeSense: An IoT Testbed for Social Research of Indoor Activities

JIE JIANG, Southern University of Science and Technology and University of Surrey
RICCARDO POZZA and NIGEL GILBERT, University of Surrey
KLAUS MOESSNER, Chemnitz University of Technology and University of Surrey

There has been increasing interest in deploying Internet of Things (IoT) devices to study human behavior in locations such as homes and offices. Such devices can be deployed in a laboratory or “in the wild” in natural environments. The latter allows one to collect behavioral data that is not contaminated by the artificiality of a laboratory experiment. Using IoT devices in ordinary environments also brings the benefits of reduced cost, as compared with lab experiments, and less disturbance to the participants’ daily routines, which in turn helps with recruiting them into the research. However, in this case, it is essential to have an IoT infrastructure that can be easily and swiftly installed and from which real-time data can be securely and straightforwardly collected. In this article, we present MakeSense, an IoT testbed that enables real-world experimentation for large-scale social research on indoor activities through real-time monitoring and/or situation-aware applications. The testbed features quick setup, flexibility in deployment, the integration of a range of IoT devices, resilience, and scalability. We also present two case studies to demonstrate the use of the testbed: one in homes and one in offices.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Applied computing** → **Sociology**; • **Computer systems organization** → **Sensor networks**; • **Hardware** → **Sensor devices and platforms**;

Additional Key Words and Phrases: Internet of things, testbed, social research, home activities, work environment

ACM Reference format:

Jie Jiang, Riccardo Pozza, Nigel Gilbert, and Klaus Moessner. 2020. MakeSense: An IoT Testbed for Social Research of Indoor Activities. *ACM Trans. Internet Things* 1, 3, Article 17 (May 2020), 25 pages.

<https://doi.org/10.1145/3381914>

The work was carried out as part of the “HomeSense: Digital Sensors for Social Research” project funded by the Economic and Social Research Council (grant ES/N011589/1) through the National Centre for Research Methods.

Authors’ addresses: J. Jiang, Department of Computer Science, Southern University of Science and Technology, Shenzhen, 518055, China, and Centre for Research in Social Simulation, University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom; emails: jiangj@sustech.edu.cn, jie.jiang@surrey.ac.uk; R. Pozza, Institute for Communication Systems, University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom; email: r.pozza@surrey.ac.uk; N. Gilbert, Centre for Research in Social Simulation, University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom; email: n.gilbert@surrey.ac.uk; K. Moessner, Faculty of Electrical Engineering and Information Technology, Chemnitz University of Technology, Chemnitz, 09111, Germany, and Institute for Communication Systems, University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom; email: k.moessner@surrey.ac.uk.

This work is licensed under a Creative Commons Attribution International 4.0 License.

© 2020 Copyright held by the owner/author(s).

2577-6207/2020/05-ART17

<https://doi.org/10.1145/3381914>

1 INTRODUCTION

Internet of Things (IoT) is a paradigm encompassing various application areas, ranging from smart homes and buildings, intelligent transportation systems, and smart cities to industrial automation, smart healthcare, and smart grids [Al-Fuqaha et al. 2015]. In the domain of smart homes and buildings, IoT mainly advocates the need for distributing multi-modal sensing and actuation capabilities into our living environments (e.g., homes, offices) to make the objects we interact with every day smarter and thus improve our lives. In such scenarios, being aware of the contexts or surrounding environments [Perera et al. 2014] and of the daily activities [Wang et al. 2018] carried out by the occupants not only help in designing smart environments and intelligent applications but also allow social researchers to study in a larger scale, such as household or office practices in real-world settings.

Social researchers take a great interest in understanding real-life practices, such as family dynamics and child rearing (e.g., Atkinson and Jacobs [2016]; Ruppanner [2015]), practices around meals [Gattshall et al. 2008], sleep [Williams et al. 2015], assisted living arrangements and mobile health solutions (e.g., LópezGómez and Sánchez-Criado [2015]; Mort et al. [2013]), homeworking [Sullivan and Gershuny 2016], and energy-related practices [Pierce et al. 2010]. In office environments, monitoring the indoor environment quality and the productivity of the occupants is another area of research [Al Horr et al. 2016] where matters such as personal thermal comfort, levels of ambient lighting, noise, and air quality are some of the main properties requiring monitoring and control to ensure the well-being of office occupants.

Existing social research methods are both qualitative and quantitative, and often some combination of the two is used for pragmatic and inference purposes [Shannon-Baker 2016]. With the advance of IoT, sensor-generated data are becoming widely available, and the use of such data for social studies of indoor activities has thrived in recent years with applications in areas such as smart homes and offices, as well as assisted living. For example, Williams et al. [2015] discuss the use of accelerometers to study people's sleep patterns. Amft and Tröster [2008] study people's dietary behavior by using inertial sensors to recognize movements, a sensor collar for recognizing swallowing, and an ear microphone for recognizing chewing. Wang et al. [2014] help in detecting elderly accidental falls by employing accelerometers and cardiometers. Microsoft used a multi-modal system [Oliver and Horvitz 2005] for office activity recognition, recognizing user state from cameras, audio sensing, and computer keyboard and mouse interactions.

In the literature, there are in general two types of IoT infrastructure that have been used by social researchers to supplement surveys, questionnaires, interviews, time-use diaries, and so forth. The first type of IoT infrastructure is embedded in the concept of living labs, which are an initial attempt to carry out social studies in realistic environments. The Philips HomeLab [de Ruyter et al. 2005] and the PlaceLab [Intille et al. 2006] are two examples of such living labs that provide a living space where the participants agree to stay for a period of time and have their activities and social interactions monitored. These living labs enable researchers to study constructed social environments, but require people to temporarily leave their original real-world settings (e.g., homes and offices) and behave as if they were in real-world habitats. A drawback of the living lab method is that the experimental settings might not be able to account for the diversity of the environments and behaviors existing in the real world. Moreover, this method cannot scale due to the cost of the infrastructure, and it is also difficult to recruit participants if the experiments require a long period of continuous participation (e.g., several months). The second type of IoT infrastructure is the smart home testbeds developed for example in the CASAS project [Cook et al. 2013], CARE project [Kröse et al. 2008], and Sphere project [Zhu et al. 2015]. These projects were early efforts to design sensing and actuating infrastructures that can be deployed into real homes for monitoring

people's daily living for research purposes such as health monitoring. Nevertheless, these testbeds focus more on the analytical aspects of sensor-generated data rather than provide an easy-to-use IoT infrastructure for social researchers to build their own applications for studying different aspects of indoor activities. Moreover, the design of these testbeds is mostly tailored to showing the viability of using sensor-generated data in specific research domains and thus provide limited support for sensor integration, privacy control, connectivity, and data visualization, among others.

For social studies of indoor activities to be carried out smoothly, it is essential that an IoT testbed meets the following requirements:

- The IoT testbed should be able to be tailored to diverse experiment settings.
- The IoT devices should be non-intrusive, cost efficient, and easy to deploy.
- The IoT devices should not interfere with or disturb the participants' daily routines.
- The privacy of the participants should be guarded at a high standard during and after the data collection.
- The experiment coordinators should be able to monitor the data collection.

To these ends, in this article we introduce MakeSense, an IoT testbed that aims to facilitate the experimentation of social research in real-world indoor environments such as homes and offices. The features of MakeSense can be summarized as follows:

- The testbed supports extensible APIs for integrating additional self-developed and commercial sensors.
- The non-intrusive design of the testbed facilitates user acceptability by excluding image/video-based IoT devices and minimizes the disturbance to the participants' daily routines.
- The IoT devices are composed of commodity sensor modules to minimize the material cost.
- The testbed provides pre-built Docker images and scripts¹ to automate the setup and deployment process.
- The onboard WiFi and Bluetooth enable quick and flexible deployment of the devices in complex environments.
- End-to-end data encryption and pseudonymization are employed to preserve the security and privacy of the participants.
- The testbed supports both real-time data transfer and local storage to enable real-time monitoring and ensure the quality of data collection.

The rest of the article is organized as follows. In Section 2, we discuss related work. In Section 3, we introduce the IoT testbed, namely MakeSense, and illustrate its sensor network, mechanism of data collection, data management and monitoring, data visualization, and analysis. In Section 4, we present two use cases showing how MakeSense was used to assist two research projects regarding home and office monitoring. Thereafter, in Section 6, we share some lessons learned from the two projects. Finally, we conclude our work with some possible extensions in Section 7.

2 RELATED WORK

Recent technological advances have resulted in cost reduction and increased availability of hardware needed for conducting real-life experiments at a large scale, which drives the need for IoT testbeds to enable faster experimentation with various sizes, hardware, topologies, and degrees of flexibility [Tonneau et al. 2015]. In the literature, many testbeds were developed to facilitate wireless sensor network experimentation. For example, one of the most comprehensive IoT

¹<https://github.com/jiejiang-jojo/make-sense>.

testbeds—SmartSantander [Sanchez et al. 2014]—provides an experimental research facility that could be used to test smart city applications and services. It offers thousands of fixed and mobile sensors (environment, parking, transportation, etc.). Another large-scale testbed—FIT IoT-LAB [Adjih et al. 2015]—provides 2,728 low-power wireless nodes and 117 mobile robots for experimenting with large-scale wireless IoT technologies, ranging from low-level protocols to advanced Internet services. Users can manage the nodes remotely through convenient tools to customize the firmware, launch repeatable experiments, and gather results. Some other examples of testbeds targeting smart city applications include SoundCity [Issarny et al. 2016], FINE [FINE 2019; Sánchez et al. 2018], among others. Although these testbeds are powerful in aspects such as scale, heterogeneity, and mobility, they are often complex to set up and require a lot of technological training. In contrast to the large-scale IoT testbeds aforementioned, OpenTestBed [Munoz et al. 2019] provides a minimalistic design by building a testbed from off-the-shelf components such as Raspberry Pi single-board computers, OpenMote B low-power wireless devices that are fitted into glass domes known as OtBoxes. Each OtBox connects to an MQTT broker over WiFi without the need for dedicated wiring or back-end. Although OpenTestBed is a simple and scalable testbed for testing IoT solutions, there is no direct support of integrating sensing devices and sensor data management. Another group of testbeds similar to ours focuses on the experimentation of smart building applications. For example, the SmartCampus testbed [Nati et al. 2013] provides a facility for IoT experimentation using a variety of sensor devices deployed within buildings that could measure temperature, humidity, illuminance, noise, air quality, occupancy, and energy consumption. Similarly, the KETI testbed aims to collect sensing data from offices and parking lots by deploying sensors that could measure indoor climate, energy consumption of office utilities, people's presence in offices, and parking lot status. These two testbeds are similar to ours in terms of the provided IoT devices and domains of application. However, neither of them offers an end-to-end solution from data collection to data visualization with automated setup and deployment support.

Living labs are another popular way of conducting research of indoor activities in realistic settings, and began to flourish in the past decade. An early establishment is the Philips HomeLab [de Ruyter et al. 2005], which provides behavioral researchers an instrument for studying human behavior in a home setting with cameras and microphones installed throughout the home. Another example is the MIT PlaceLab [Intille et al. 2006], which offers a facility for the study of ubiquitous technologies in home settings via a network of sensors that capture a complete record of audio-visual activity, including information about objects manipulated, environmental conditions, and use of appliances. Some other examples of living labs include the Gator Tech Smart House [Helal et al. 2005] from the University of Florida; the MIT Human Speechome Project [Roy et al. 2006]; the Assisted Living Laboratory [Kleinberger et al. 2007] at the Fraunhofer IESE in Kaiserslautern; the Georgia Tech Aware Home [Kientz et al. 2008]; the TigerPlace [Skubic et al. 2009] in Columbia, Missouri; and, more recently, the Halmstad Intelligent Home [Lundström et al. 2016] at Halmstad University and the UJAmI [Espinilla et al. 2018] at the University of Jaén, Spain. However, most of these labs have the drawback of using intrusive sensors such as cameras. Even though in some cases the cameras were used to capture just silhouettes, privacy issues still stand since the data can be used to extrapolate contextual information about occupants. Another drawback of the living lab approach is that the experimental settings might not be able to account for the diversity of the environments and behaviors existing in the real world. Moreover, this method cannot scale due to the cost of the infrastructure, and it is also difficult to recruit participants if the experiments require a long period of continuous participation (e.g., several months).

There is another relevant group of works focusing on home monitoring and activity recognition. For example, the Ambient Assisted Living Context Aware Residence for Elderly (CARE) project [Kröse et al. 2008] proposed a non-intrusive sensing platform by leveraging proximity and

switch-like sensors to recognize activities from the elderly living alone at home. The Washington State University CASAS project [Cook et al. 2013] tried to move away from living laboratories with easy-to-install battery-operated devices deployed in homes such as temperature sensors, door sensors, and infrared motion/light sensors. The SPHERE project [Zhu et al. 2015], a platform based on wearables, video cameras, and environment sensors for healthcare in residential environments, was deployed in 100 households in Bristol, UK. Targeting general-purpose sensing, the Synthetic Sensors [Laput et al. 2017] developed by the researchers from Carnegie Mellon University integrate a set of non-intrusive sensor modules and proposed to apply machine learning methods to recognize meaningful events at different abstraction levels from raw sensor data. Although these works all provide a solution for home monitoring with different sets of sensors, their focus is either on the sensing technology itself and/or on researching methods for activity recognition rather than developing an IoT infrastructure that could be easily adjusted and deployed for social research of indoor activities in different settings such as homes or offices.

3 MAKESENSE

MakeSense has been designed to provide an infrastructure that enables real-world experimentation for large-scale social research of indoor activities through real-time monitoring and/or situation-aware applications. The use of cloud infrastructure for data monitoring and storage, as well as for data analysis and data visualization, introduces benefits for scalability, ease of extension, reliability, and manageability. The design of the self-manufactured IoTEgg device, developed for MakeSense, leverages commodity ambient sensors and moves away from intrusive IoT devices, such as cameras. Furthermore, due to its multi-radio setup with Bluetooth Low Energy (BLE) and WiFi, MakeSense allows ad hoc self-manufactured or commercial sensors to be added and to be interfaced with the IoTEggs acting as relays. This can be achieved easily by exploiting MakeSense's standard APIs and providing access to sensors and actuators via its IoT protocol—the Open Mobile Alliance (OMA) lightweight M2M (LWM2M) protocol [OMA LWM2M 2018], which allows users to interact and remotely manage resource-constrained IoT devices in a secure and efficient way.

3.1 Testbed Architecture

MakeSense adopts a client-server-like paradigm. All of the sensing devices deployed in indoor environments such as homes or offices, as shown in the top row of Figure 1, serve as clients that collect physical environment data that are sent directly to the server back-ends via LWM2M through Broadband Internet. The server back-ends can be hosted on a private cloud or an on-premise dedicated server with enough capacity (as shown in the center of Figure 1). Besides receiving data from sensors and storing them on persistent storage, the server back-ends also provide support for administration and visualization through web applications. In the rest of this section, we will elaborate on the technical design choices of MakeSense.

MakeSense features both commercial off-the-shelf (COTS) IoT devices (i.e., energy monitors and BLE wristbands) and self-manufactured IoT devices (i.e., the IoTEggs) (see Sections 3.2.1 and 3.2.2 for more details). These IoT devices can be directly connected to WiFi access points or benefit from the capability of the IoTEggs' WiFi modules to operate in both station and access point modes, as well as from their onboard BLE module. Both static and mobile (e.g., wristbands) COTS and self-manufactured BLE- and WiFi-based sensors/actuators can thus be utilized by the IoTEggs, augmenting their onboard default set of sensors/actuators and enabling opportunistic sensing interactions. Data captured by the IoT devices are collected from houses or buildings relying on readily available WiFi networks and Internet access, thus benefiting from higher reliability, lower interference, and higher throughput as compared with other radio technologies that would need to be deployed (i.e., other lower-power radios). Concerning power supply, today most houses and

Fig. 1. MakeSense network layout.

buildings have sufficiently distributed power sockets, which makes the powering of IoT devices such as the IoTEggs easy. Moreover, to reach locations not easily accessible or subject to mobility, battery-operated BLE sensors may also be used, but they have to be charged periodically.

MakeSense supports data transmission and device management over LWM2M, which is used together with the Constrained Application Protocol (CoAP) [CoAP 2018], an application layer protocol designed for resource-constrained IoT devices. By using WiFi and CoAP via a UDP protocol, high temporal resolution measurements can be collected, limited only by the sampling requirements of the physical sensors. Furthermore, MakeSense implements security at the communication level with Datagram Transport Layer Security (DTLS) and pre-shared key authentication, and all of the data are encrypted with Advanced Encryption Standard (AES) 128 bit before transmission.

The server side is deployed on a private cloud that supports persistent storage of sensor-generated data via databases deployed on virtual machines (VMs). The infrastructure is based on OpenStack, with storage distributed identically between disk/volume-based (Cinder) and document-based object (Swift) storage services. Moreover, front-end interfaces for monitoring, administration, and visualization are deployed inside VMs that can be easily accessed anywhere through the Internet via web browsers. The use of a private-based cloud solution was chosen because it enables scalable and efficient handling of large-scale experiments, thus reaching a large number of users.

MakeSense introduces a reliable storage solution by providing data backup and replication. On the remote cloud infrastructure, data storage serves two purposes: data post-processing and real-time data visualization/analysis. In the former case, the open source databases such as PostgreSQL [PostgreSQL 2018] and MongoDB [MongoDB 2018] are used. In the latter case, the open source search engine Elasticsearch [ElasticSearch 2018] is used together with its dedicated visualization platform Kibana [Kibana 2018]. MakeSense opts out the use of Logstash [Logstash 2018], a data loading service commonly used together with Elasticsearch, and instead employs a lightweight replacement based on Python. Users may adjust the choice of these database and visualization/analysis modules according to their needs.

Fig. 2. MakeSense software architecture.

Figure 2 shows the software architecture of MakeSense. Since LWM2M adopts a client-server architecture, we developed an LWM2M client running on the IoTegg, interfacing with the embedded sensors and actuators, and communicating via CoAP with an LWM2M server. To provide a model for interoperability between devices and applications/services deployed on the cloud side, the sensors and actuators were developed as IPSO Smart Objects [Jimenez et al. 2016].

By leveraging LWM2M and IPSO Smart Objects, M2M applications are deployed on the cloud and operated using standard APIs. LWM2M clients maintain their registration and thus their current availability, as well as the list of their objects and resources (i.e., sensors and actuators), via an LWM2M server registry, accessible via HTTP requests. For MakeSense, as shown in Figure 2, we designed two M2M applications interfacing with the LWM2M server via HTTP: one for building images and downloading firmware over-the-air (FOTA) to the IoTeggs and the other for handling data collection from the IoTeggs, configuring the notification interval and initiating requests for data collection (if the device or sensor is not blacklisted) via the standard LWM2M Observation/Notification interface upon device registration.

The LWM2M server publishes via Advanced Message Queuing Protocol (AMQP) to a RabbitMQ message queue broker. It publishes both control messages (e.g., LWM2M client registrations) on one queue exchange and live sensor measurements on another queue exchange. The use of a message queue is for asynchronous communication, increased reliability and scalability, and decoupling between the data collection and data storage/visualization applications. In addition to the M2M application for data collection that subscribes to the control message queue exchange, we designed another two applications to collect sensor measurements from the live data message queue exchange and store them into historical data stores (e.g., PostgreSQL and MongoDB) and into Elasticsearch for subsequent visualization and analysis with Kibana.

3.2 Sensor Network

MakeSense supports a range of connectivity options enabling virtually any sensor capable of interfacing with BLE and WiFi to be used in conjunction with the IoTegg acting as a gateway. Given both the non-intrusive design of the IoTegg and the secured communication and storage, the security and privacy of the participants of the study is preserved. Furthermore, due to its scalable design in terms of both sensor variety and ease of extension in the cloud, MakeSense enables research experiments to be carried out in a wide range of real residential and commercial indoor environments.

3.2.1 Self-Manufactured Sensor Suites. In the market, there are many electronic modules that can be used to build sensor suites for different measurements. One example of such development is the IoTegg designed and developed at the 5G Innovation Centre (5GIC) at the University of Surrey. The IoTegg is encapsulated in a box, as shown in Figure 3(a), and is coordinated by a Seeeduno

Fig. 3. IoTEgg consisting of a suite of sensor modules.

Arch-Pro development board [Seeeduno 2018], based on an NXP LPC1768 microcontroller unit (MCU). A temperature and humidity sensor board HTU21D [HTU21D 2018] is managed via an I2C interface and sampled periodically by the client application deployed on the ARM core. An Avago ADPS-9960 light sensor [APDS-9960 2018], also managed via an I2C interface, is used to sample ambient light measured in $\frac{\mu W}{cm^2}$. The GP2Y0A60SZ ranging sensor from Sharp [GP2Y0A60SZLF 2018] included in the design is an analog sensor with a wide detection range of 10 to 150 cm, which is sampled via a 12-bit ADC and converted through the manufacturer’s calibration table. Finally, a MEMS microphone breakout board INMP401 [INMP401 2018] is used to sample noise levels in the environment on another ADC channel and the values are converted to sound-pressure-level decibels. The sampling rate of the IoTEgg can be customized for different use cases via a configuration file.

The IoTEgg has been designed, prototyped, and manufactured internally in our center as a means for both capturing environmental changes in physical measures and acting as a hub for relaying data from and to additional wireless sensors. The IoTEgg has been equipped with a BLE (Bluegiga BLE112) module that allows additional COTS or self-manufactured sensor suites to be added and connected to the IoTEgg, so to enable further data collection. Furthermore, the WiFi module (ESP8266) allows both operation either as a WiFi access point or as a WiFi station, or both together at the same time, thus enabling also additional WiFi-based sensors to be connected to the IoTEgg for relaying data.

This setup has the clear advantages of not requiring much effort for installation; the infrastructure can be deployed in minutes; and it allows practical extensibility with COTS or ad hoc battery-operated devices connected to the IoTEgg working as a gateway, as well as allows local storage in the IoTEgg. For example, static wireless devices can be attached to fixed real-world objects (e.g., motion sensors to drawers, doors) or mobile real-world objects (e.g., wristbands and smartphones to people), thus capturing broader dynamics.

One may assemble a combination of different electronic modules to build sensor suites according to their needs. The advantages of self-manufactured sensor suites are as follows: (1) there is a wide selection of modules to choose from, (2) the data collection of each module can be adjusted for specific uses, and (3) the cost of the sensors can be controlled by choosing products of varying degrees of sensitivity. However, there are several disadvantages of self-manufacturing sensor suites: (1) manufacturing requires solid engineering background to be able to design low-cost and high-quality sensors; (2) there is usually a fairly long development and test cycle; and (3) the cost of sensors might be higher than mass production, and for larger volumes, costs can be reduced.

3.2.2 COTS Sensor Suites. When building one’s own sensor suite is not a choice, there is also a wide selection of commercial products ranging from elementary ones like temperature and humidity or accelerometer sensor suites (e.g., Ti Sensor Tags [Texas Instruments 2018]) to more sophisticated ones like energy monitors, pet tags, smart locks, and wearable trackers. Figure 4 shows two examples of commercial products that have been used in a number of studies.

Fig. 4. Examples of commercial sensor suites.

Smart wristbands have been widely used by researchers in healthcare applications, such as activity recognition [Añazco et al. 2018], fall detection [Khojasteh et al. 2018], and gait analysis for Parkinson’s disease [Mazilu et al. 2015]. Figure 4(a) shows a BLE wristband called *Mi Band* [Xiaomi 2018], which is mainly used by people to track their levels of exercise, such as how many steps a person has achieved per day, which can be converted to calories.

However, as for many other commercial products, there are no open APIs available to download data for further analysis, which makes it more challenging for researchers to harvest the data for research purposes. In our case, instead of using the wristband for its original purpose, we exploited the BLE module built in both the wristband and the IoTEgg to capture the received signal strength indication (RSSI) distribution over time, so as to have a rough localization method. This allowed us to estimate how close the wristband is to the IoTEggs placed in different locations of a building based on the RSSI measured by each IoTEgg. The combined use of the IoTEgg and the wristband is an example of how other sensors might be integrated to the sensing capabilities of the IoTEgg.

Another example of commercial sensor suite is the electricity monitoring kit from CurrentCost [CurrentCost 2018] as shown in Figure 4(b), which has been used in several studies (e.g., Kelly and Knottenbelt [2015]; Zhu et al. [2015]). It features a current transformer clamp, a transmitter, and a number of individual appliance monitors (IAMs) to measure the energy consumption of the whole house and individual appliances, and a monitor hub to collect and display the sensor data. The measurements are in units of Watts, and the maximum sampling rate is every 6 seconds. The users can direct the data stream from the electricity monitors to their own devices for post-processing via the standard data port and specification provided by CurrentCost.

The Mi Band and the CurrentCost energy monitor presented in this article are only two examples of integrating commercial sensors with MakeSense. There is a broad range of commercial products that could be explored. For example, Bluetooth-enabled activity trackers like Mi Band are widely available in the market, such as Fitbit [Fitbit 2019] and Garmin [Garmin 2019]. Energy-monitoring products are also gaining popularity for both home and business applications. Among the many energy monitors in the market, EmonPi [EmonPi 2019] provides an open-platform energy-monitoring solution. It supports data access via the open source web app Emoncms that could be run locally on the EmonPi itself or the user’s own server. Another energy monitor product that provides data access is Neurio [Neurio 2019]. Users can direct the data to their own cloud infrastructure via Neurio software and Cloud API. Another type of off-the-shelf device that could be easily integrated into MakeSense is the Flume water sensor [Flume 2019b]. The Flume water sensor straps around the existing water meters and sends the real-time water usage data to the Flume Bridge, which transmits the data up to the Flume Cloud via the home WiFi. By querying the Flume open API [Flume 2019a], MakeSense can collect the real-time water usage data of a household via the Raspberry Pi component. Adding the Flume sensors only involves small changes to the current energy data source. With air pollution being a common problem around the globe, air quality monitors are also becoming popular. An example of such a product is Kaiterra

[Kaiterra 2019], which measures PM2.5 and TVOC. It supports data access via download from the Kaiterra app.

With the advance of IoT, more commercial sensing products are becoming available in the market. The main advantages of using commercial products are that (1) typically less development work is required; (2) the sensors are normally more robust, being thoroughly tested; and (3) they sometimes rely on ready-to-use platforms for data monitoring. However, there are several disadvantages: (1) rigid and limited control for data collection, (2) the combination of sensor modules cannot be easily changed or extended to support new sensors, (3) cost might be higher for deployments in large volumes, and (4) the products may run out of stock or even be out of production.

3.2.3 Limitations. Currently, MakeSense does not support recording audio or video information and mainly incorporates low-energy IoT devices (low data rate sensors). For capturing audio information, the microphone module on the IoTEggs could be used. As for capturing video information, an easy-to-implement approach is to connect the off-the-shelf video cameras to Raspberry Pis, by which the video data could be collected. Since MakeSense relies on the home WiFi, it would be too much overhead to transmit raw audio or video data to the remote data server. Therefore, an alternative is to store the raw audio and video data locally while sending the real-time summary data to the data server. We are planning to extend MakeSense to cater to such needs in the future.

3.3 Data Collection

Today, most houses and buildings are covered by WiFi access points that can be used without the need to install dedicated gateways or alternative solutions. As such, MakeSense aims to facilitate data collection through easy to set up, fast, and reliable WiFi networks. Furthermore, the ease of adding other types of Bluetooth- or WiFi-enabled sensors and actuators is supported by the IoTEgg, which can be used to relay measurements or control messages back and forth from the cloud infrastructure.

For MakeSense to provide most of the desirable features of an IoT testbed (remote management, reprogramming, customizable data collection, live query for location and connectivity, security, etc.), we leverage LWM2M over CoAP and encryption with DTLS and pre-shared keys authentication. Moreover, it is important that all of the data are pseudonymized from the time they are generated, making it difficult to trace the participants. MakeSense implements data pseudonymization by assigning random identifiers for each participant, and the links between the identifiers and the participants are stored separately.

Since the IoTEgg is an IoT device with limited resources, we ported the Eclipse Wakaama [Wakaama 2018] C library (designed to be built into Linux targets) on our resource-constrained target (i.e., the Seeeduino Arch-Pro), which runs no operating system and just some libraries as a hardware abstraction layer to interface with peripherals. The library adopts a CoAP C implementation that is an adaptation of the Contiki's Erbium CoAP library.

A full-fledged LWM2M client has been developed, featuring the following LWM2M Objects (with standard object numbers): LWM2M Security, LWM2M Server, Access Control, Device, Connectivity Monitoring, Firmware, Location, and Connectivity Statistics. For each of these objects' resources, API and drivers have been implemented in C++ and used by Wakaama to be provided as callable endpoints from any LWM2M server. This allows for configuring the device remotely and providing features such as setting security mode (in our case AES128 was used, provided by the tinyDTLS C library), default device lifetime, notification period, static GPS coordinates, time synchronization, factory reset, and remote reboot. Furthermore, network and connectivity parameters and statistics can be inquired from remote such as WiFi RSSI, IP addresses, and configuration.

Fig. 5. OMA LWM2M server objects in MakeSense.

Finally, APIs for FOTA updates have been customized to allow reliable download of new software updates for the LWM2M client via an ad hoc secondary boot loader (SBL) that exploits self-programming via an external flash memory in the IoTegg.

On the server side, as shown in Figure 5, the LWM2M server (running the Eclipse Leshan [Leshan 2018] Java library) is able to access not only the standard LWM2M objects, but also the additional IPSO Smart Objects (with standard object numbers), representing ambient light luminosity (3301) in microwatts per square centimeter ($\mu W/cm^2$), temperature (3303) in degrees Celsius ($^{\circ}C$), humidity (3304) in percentage (%), RGB led control (3311), loudness (3324) in sound-pressure level decibels (dB SPL), dust concentration (3325) in milligram per cubic millimeter (mg/mm^3), proximity distance (3330) in centimeters, buzzer control (3338), and gesture recognition for which we adopted the multi-state detector (3348) where each integer from 0 to 6 represents directions for the gestures.

Some of the advantages of this design are that (1) users of MakeSense could easily design M2M applications themselves to collect data in various data formats and time intervals or act upon the IoTegg actuators in real time, (2) updates can be sent over-the-air with minimal disruption, (3) monitoring can also be done remotely to identify issues, (4) the LWM2M standard allows for ease of extension and promotes interoperability due to the use of standard IPSO Smart Objects (see LWM2M specifications for more details), and (5) both data pseudonymization and encryption are implemented to provide efficient data security.

3.4 Data Management and Monitoring

All of the cloud data management is handled via a RabbitMQ message queue broker, deployed on a VM and secured with access control and authentication mechanisms. The broker has three

main subscribers listening for data: the data collector service, the data visualization service, and the historical data store service. The data published to the broker comes from two main sources: the control messages (e.g., LWM2M registration updates) and the live data stream from the sensors.

The data collector M2M application, running on a VM as a reliable and always-on Linux service, has been developed in Python. This application listens for the LWM2M client registrations to the LWM2M server via the RabbitMQ, where the server publishes registration messages. The application is thus aware of when devices come online and when data collection from the IPSO Smart Objects can be started by sending requests to the LWM2M server via the HTTP interface. Furthermore, this application helps the IoTeggs synchronize the time with that of the VM.

The data visualization application is also running on VMs where Elasticsearch and Kibana have been deployed. This has been designed as a reliable and always-on service in Python, running as multiple instances in parallel. Live sensor data coming from the RabbitMQ is stored in disk drives mounted as OpenStack volumes via the Cinder service. Concerning the historical data store application, a similar service was designed with storage based on databases such as MongoDB and PostgreSQL.

An M2M application for build generation and FOTA download has also been designed. The GNU ARM Embedded toolchain has been customized to generate both the SBL and the main LWM2M client application allocated in two separate memory intervals in the MCU internal flash memory. The M2M application thus generates the build image, uploads it to the reliable OpenStack Swift object storage service, injects a magic number (i.e., a unique identifier that will be cross verified when the build image is downloaded), and initiates a reboot on the corresponding LWM2M client(s) to prepare for a reliable software update via the SBL at the next reboot.

MakeSense integrates the data storage module (e.g., PostgreSQL) and the data visualization module (ElasticSearch and Kibana) by providing pre-built Docker [Docker 2018] images and scripts that automate the server deployment process.² However, to be able to use PostgreSQL to make data queries or use Kibana to create data visualizations, one needs to be familiar with the basics of SQL and Kibana visualizations.

Some of the advantages of this architecture are that (1) users of MakeSense can easily subscribe to the broker and add their workers to implement live data processing and analysis, (2) MakeSense provides both historical data storage and short-term storage for live visualization, and (3) remote control of the IoT devices makes it easy for software updates and thus facilitates infrastructure maintenance.

3.5 Data Visualization and Analysis

MakeSense relies on the open source search engine Elasticsearch [ElasticSearch 2018] and its dedicated analytics and visualization platform Kibana [Kibana 2018] for data visualization and analysis. Elasticsearch supports scalable and real-time search with an HTTP web interface and schema-free JSON documents. Kibana provides visualization capabilities on top of the large volumes of data indexed by an Elasticsearch cluster. Users can create different kinds of visualizations such as line, bar, and scatter plots, pie charts, and maps with the indexed data. Furthermore, users can combine different visualizations to create customized dashboards, as shown in Figure 6, which can be shared via web or integrated as HTML frames into existing web applications. An extended view of the lower two quadrants in Figure 6 will be shown later in Figure 8 and explained in the associated text.

²<https://github.com/jiejiang-jojo/make-sense>.

Fig. 6. Kibana dashboard.

Table 1. MakeSense Features

	Scale	Environment type	Heterogeneity		Mobility	Con- currency	Primary Use Case
			Node Type	Protocol			
Make-Sense	>2,000 sensors	Real homes and offices	Yes	Yes	Yes (wristband based)	Multi-user	Home and office monitoring

As for data analysis, in addition to simple statistics (max, min, standard deviation, etc.), users may carry out more advanced data analysis exploiting Kibana’s plugins, such as X-Pack [X-Pack 2018], which enables the analysis of time-series data by creating models of normal behaviors in the data and identifying anomalous patterns in that data.

3.6 Feature Summary

In line with the evaluation of the existing IoT testbeds, Table 1 gives an overview of MakeSense’s capabilities relating to the criteria used in Chernyshev et al. [2018]. MakeSense has been demonstrated to support more than 20 home deployments and a multi-layer office building deployment that contains more than 2,000 sensor nodes. The types of environment that MakeSense targets are mainly indoor environments such as homes and offices. Many types of sensor nodes are supported, including the self-developed IoTeggs, off-the-shelf products such as the CurrentCost energy monitors, and the wristband Mi Band. The sensor networks of MakeSense are connected to each other either via WiFi or Bluetooth. Moreover, MakeSense adopts the LWM2M protocol for remote management of sensing devices and related services. Multiple distinct experiments/evaluations can be performed on MakeSense in parallel.

Fig. 7. Example of sensor suites installation at home.

4 SHOW CASES

4.1 Home Monitoring

A recent application of MakeSense was in the HomeSense project [HomeSense 2018], which aimed at studying how digital sensors can be used by social science researchers to facilitate their understanding of home life (i.e., what do people do at home, when, where, and alone or with others). For example, it is expected that when people cook hot meals, there are corresponding changes in signals such as temperature, humidity, electricity usage, and so forth that can be captured by sensing devices. In the rest of this section, we illustrate how MakeSense was used in the HomeSense project with respect to IoT device deployment and data analysis.

4.1.1 Installing Sensor Suites in Homes. The HomeSense project recruited 20 households where sensor suites were installed to monitor the occupants' home activities. The suites of sensors used in HomeSense consist of the IoTegg, the Mi Band wristband, and the CurrentCost energy monitor illustrated in Section 3.2.2.

With each household having a different configuration of physical spaces, household members, and internet connectivity, we had to plan the locations of deployments for the sensor suites such that the installation could be easily adjusted to different households. For example, Figure 7 shows a floor plan of a household with spots marked for IoT device installation. The principle for installation is to cover as much as possible the space of the house such that the environmental changes caused by human activities can be captured. However, there are also cases where the participants asked explicitly to have certain rooms/places excluded due to privacy concerns. The number of devices installed in each participating house ranged from 13 to 22, and their connectivity setup on average took about 10 minutes. MakeSense currently does not provide any mapping tool for drawing floor plans. For the HomeSense project, the researchers manually drafted the floor plan during the first visit to a participating household and marked down potential locations for installing sensors. The floor plans were then cleaned up, finalized, and digitalized after the field deployment was completed.

Fig. 8. Visualization of the data collected from the IoT devices (IoTegg and energy monitors).

WiFi was available in most parts of each household, which makes it possible to transmit the data collected by the IoT devices in real time to the remote data server deployed on a cloud infrastructure managed by OpenStack. The data collected (every 3 seconds) from each IoTegg are directly sent to the remote data server. The data collected (every 6 seconds) from the energy monitors are first directed to a Raspberry Pi via a USB to RS232/TTL PL2303HX cable and then sent to the remote data server while a local copy is also saved. Concerning the wristbands, household members were asked to wear them when at home so that all of the IoTeggs installed in the house were able to detect the presence of each household member by means of the Bluetooth signal strength. Moreover, all of the data are encrypted with AES (128 bits), and the household-specific information is Pseudonymized.

4.1.2 Recognizing Home Activities from Sensor-Generated Data. In HomeSense, it was expected that the data collected from all of the sensors deployed in different households could be used to help researchers better understand the activities occurring in the house and how they were assembled into household routines. To do so, it is essential that the researchers are capable of seeing the data in action and in a systematic way (i.e., how the data can be visualized is important). To assist social researchers in easily visualizing and analyzing the sensor-generated data (in real time), MakeSense provides templates for visualizing time series. For example, Figure 8 shows a screenshot of the sensor-generated data collected in one participating household via the visualization module of MakeSense.

The visualization of the data collected from the IoTegg, as shown in the upper plot of Figure 8, reveals the changes of signals in the kitchen of the household by measurements of range, temperature, humidity, wristband signal strength, brightness, dust density, and noise level (indicated by line plots of different colors). We can see that the kitchen has been used several times during the day based on the observation that the ranging sensor detected moving objects in proximity (lower values indicating objects moving closer to the ranging sensor) and the signal strength of

Fig. 9. Visualization of the time-use diary.

the wristband worn by the occupant has increased (higher signal strength indicating that the person wearing the wristband is closer to the IoTEgg). Accordingly, we can see that the noise level also increased. The changes in the humidity level may be caused by drink or food preparation. Notice that the ranging sensor outputs a maximum value of 80 cm due to the narrow space in the kitchen and the location where the IoTEgg was placed. Moreover, the values of the wristband signal strength, brightness, and dust density are all re-scaled using Kibana's scripted fields to integrate all of the sensor information in one plot.

The visualization of the data collected from the energy monitors, as shown in the lower plot of Figure 8, indicates how much electricity the whole house has been using, as well as the electricity consumption of individual appliances, including the iron/vacuum, TV, electronics, kitchen appliances, and washing machine via area plots of different colors. In relation to the visualization of the data from the IoTEgg, we can see that the electricity usage of the kitchen appliances increased when there were movements detected by the IoTEgg in the kitchen. Moreover, there is also correlation between the electricity usage of the washing machine and the presence in the kitchen. This is because the washing machine is located in the kitchen.

To evaluate whether our observations from the sensor-generated data are correlated to what happened in the house, the HomeSense project also collected a time-use diary from each household giving detailed information of what activities occurred where, when, by whom, and with what types of appliances with an interval of every 10 minutes. Figure 9 shows an example of the occurrences of four types of activity recorded in a time-use diary, including cooking, watching TV, dining, and doing laundry, indicated by dots of different values and colors. For example, we can see that the occurrences of laundry activity (colored red with value 4) have a strong correlation with the electricity usage of the washing machine.

Comparisons of the recordings obtained from the time-use diaries and the knowledge extracted from the sensor-generated data not only help the researchers gain an in-depth understanding of the underlying environments where different activities were carried out but also help them validate their reasoning of whether certain activities occurred and/or how the activities occurred. Please refer to Jiang et al. [2017] for a more detailed illustration of home activity inference based on sensor-generated data.

With MakeSense, the social researchers were able to deploy a rich set of non-intrusive sensor suites in 20 real households. The data collected from the sensors provided the researchers with extensive information of the physical environments where the participants' daily routines are

Fig. 10. 5GIC IoTegg desk deployment.

carried out. The visualization of such data further helps the researchers understand the flow of household activities and validates their reasoning about the household practices, together with the information obtained from other methods such as time-use diaries.

4.2 Office Monitoring

Office environments can be monitored to ensure that the indoor environmental quality and productivity standards are met [Al Horr et al. 2016]. Thermal comfort, ambient light levels, noise levels, indoor air quality and ventilation are some of the fundamental physical variables to measure to ensure the well-being of office occupants. In this section, we report on the work carried out with MakeSense in a real office setting, demonstrating how social scientists can benefit from the testbed to set up experiments to monitor and study human behavior in an interactive way.

4.2.1 Deploying Sensor Suites in Offices. Figure 10 shows the deployment of 100 IoTeggs at the 5GIC at University of Surrey, with each IoTegg placed on a numbered desk. The link between the IoTegg LWM2M client ID and the desk number is randomized, thus making it difficult to trace the desk user.

Each numbered desk is assigned one IoTegg that is placed on top of it, facing the user, and plugged into a nearby power socket and connected to the WiFi access points available in the 5GIC. The sensors measurements are collected every second from each IoTegg by the M2M application running on a VM deployed on the in-house OpenStack installation.

4.2.2 Indoor Environmental Quality Monitoring and Feedback. The office monitoring system aims at studying how certain environmental variables influence the level of comfort and working conditions of the office occupants and provides an M2M application that is capable of providing feedback to users (e.g., via a buzzer actuator and an email notification service). This is achieved by leveraging the IoTegg proximity sensor for estimating desk occupancy and the IoTegg temperature, humidity, light, and dust sensors for calculating the difference between the real-time sensor measurements and the user-specified preferences. Moreover, users can configure their desk location (number) and a preferred email address for getting notifications about when the desired standards of comfort are violated so that they could take actions to make improvements.

Figure 11 shows the interface via which users can specify their environmental preferences and monitor the changes of the environmental conditions. Six text boxes are provided for the users to

Fig. 11. 5GIC multi-modal sensing office monitoring live data.

configure their desk locations and desired values of environmental variables. A button allows to start/stop the monitoring for the configured desk. The peaks in the plots of the figure showcase the changes of the environmental conditions.

When the monitoring starts, a background service verifies whether a user is in close proximity (i.e., by polling the proximity sensor for measurements less than 75 cm, a value that was determined empirically). If the user is in close proximity and the sensor values exceed the user preferences, notifications with customized messages will be sent to the user. In the meantime, the IoTEgg buzzer is actuated once, suggesting that the user take certain actions to improve his or her working environment. For example, suggestions for actions to be taken as a consequence of violating the desired working conditions can be provided as feedback emailed to the desk user to improve the thermal comfort, light intensity level, and air quality. Depending on the building type, suggestions could be opening the windows, adjusting the air conditioner or heating, asking for additional table lamps, or operating the blinds when available.

In this use case, we demonstrate how an M2M application and a front-end in Node.JS/React.JS can be integrated with MakeSense, benefiting from the LWM2M APIs and the IoTEgg. LWM2M allows users to manage data collection remotely and write applications in the cloud for monitoring and controlling purposes. Thus, by exploiting MakeSense, social researchers can both study people's behavior and provide context-aware feedback that people can act upon.

5 SETUP AND DEPLOYMENT

In this section, we show the steps that a social researcher could follow to set up a home monitoring application using MakeSense with IoTEggs:

- (1) Step 1. Server Deployment
 - Time: <1 hour

- Components: Data Collection Server, Data Storage Back-Ends, Data Visualization Server
 - Action: Running the script for deploying servers
 - Required Technical Skills: Linux command line (shell) operations
 - MakeSense Provides: Docker-based server images, automated deploying script³
- (2) Step 2. IoTEgg Deployment
- (a) Step 2.1. IoTEgg Configuration
- Time: <5 minutes
 - Action: Configuring IoTEgg ID, data server address, sampling rate, WiFi SSID, and password; compiling and uploading firmware to IoTEgg
 - Required Technical Skills: Using the Mbed Online Compiler or other text editors to edit the configuration file, compile, and upload the firmware to IoTEgg
 - MakeSense Provides: Configuration file template
- (b) Step 2.2. Quality Test
- Time: ~24 hours
 - Action: Deploying a set of IoTEggs in a controlled environment and comparing the data collected against each other
 - Required Technical Skills: Using Kibana to create indices and visualizations
 - MakeSense Provides: Kibana
- (c) Step 2.3. Field Deployment
- Time: <10 minutes
 - Action: Connecting IoTEgg to WiFi and checking data collection
 - Required Technical Skills: Using Kibana to create indices and visualizations
 - MakeSense Provides: Kibana

MakeSense aims at reducing the burden on social researchers of setting up the servers for receiving, storing, and visualizing data. It uses Docker technology [Docker 2018] to pack all of the components into docker images. This reduces the prerequisites of using MakeSense. Without Docker, researchers who want to use MakeSense have to install all of the dependencies, such as Python, Flask, and PostgreSQL, themselves and make sure they are configured correctly. Such effort may take a day or even longer depending on their experience. The Docker technology employed by MakeSense only requires to be installed, and all of the dependencies and configurations are pre-built into the images that can be deployed with a simple script within 30 minutes. This also reduces the amount of knowledge required for deployment to the simpler commands of the Linux Shell.

The setup of IoTEgg for deployment should follow a procedure for quality assurance. It is not uncommon to find some sensors or devices faulty. The individual sensors come with a quality guarantee, and the assembling process also includes a quality test, but faults found in field deployment are very expensive and even more so in the data collection phase. Thus, it is recommended to have a quality test just before the field deployment. Such test will further reduce the possibility of faulty components entering experiments. The recommended way of conducting such tests is to deploy a set of IoTEggs in a controlled environment and compare the data collected from each IoTEgg in a 24-hour window.

There is no need for social researchers to understand how the source code works because the IoTEggs can be configured by editing a file in the source code to change, for example, the sensor ID and server address fields. Such editing can be carried out with the online editor provided by

³<https://github.com/jiejiang-jojo/make-sense>.

MBed, a popular online IDE for IoT devices. With a few clicks, the configuration and firmware can be written to the IoTeggs, which are then ready for field deployment. The sensor ID can carry more information than just a serial number. It is recommended to structure the sensor IDs in a way that can be pattern matched to help identify visualizations later in the experiments.

It is recommended to collect relevant information and prepare the IoTeggs before the deployment. An example is the WiFi configuration details needed for the IoTeggs to access the home Internet and send data back to the servers. Provided that the IoTeggs are pre-configured before the deployment, the field work will be as simple as finding a place for the IoTeggs, plugging them to power sockets, and checking whether the data are coming into the servers via the data visualization platform. This could be done in less than 10 minutes.

The design of MakeSense was carried out by a team of engineers, computer scientists, and social scientists. It aims at providing an automated solution for social researchers to make use of digital sensors to complement the traditional social science methods for studying indoor activities. The consequence of this design is that the deployment can be made without great technical knowledge and can be done by social researchers, as was the case for the HomeSense study. For HomeSense, deployment was undertaken in the course of a home visit that also included a short interview with household members, discussion about and confirmation of “informed consent” with the household, and information about what the sensors such as the IoTeggs and energy monitors did and how the system worked. In this context, deployment of the sensing devices around the home was generally achieved swiftly and easily, and was almost a background activity in comparison with the complications of securing consent and ensuring that the household had an adequate understanding of the research.

6 DISCUSSION

MakeSense was built with the aim of facilitating the use of IoT devices for indoor monitoring for social research. Based on its use in two showcase scenarios relating to home and office monitoring, we share some experience and discuss the limitations of MakeSense.

6.1 Informed Consent

For studies that involve human subjects, we have to ensure adequately informed consent from all types of participants including children and those who participate indirectly (e.g., visitors). The researchers should feel confident that the consent they get is genuinely informed (i.e., participants actually understand what they are signing up for). We found that it is important to include demonstrations as part of the consent process to demystify the technologies.

6.2 Minimum Disturbances

Even with informed consent, for social studies of indoor activities, especially in the setting of homes, it is essential that the deployment of IoT devices does not disturb people’s daily routines. This is not only to reduce participation burden and dropout rate but also minimize the effect introduced by the IoT devices. Based on our field work experiences, there are several things to be noted: (1) non-intrusive IoT devices are preferred, (2) participants need control over the IoT devices (e.g., turning them off when there are visitors who do not want to be monitored), and (3) the IoT devices should be easily integrated to the household infrastructure (e.g., power supply, Internet connection).

6.3 Data Security

When it comes to the monitoring of people’s daily lives, especially in the setting of real homes, people have a strong desire for data security. It is essential that the sensor data is securely

transmitted and stored, and that all of the information relating to the individuals is pseudonymized. Although the participants may not be familiar with the technical details about security, they often require the researchers to explain how their data are collected, saved, and used. For example, in the HomeSense project, when demonstrating how various sensors work and what kinds of information they capture during the first visit to the participating house, some participants were interested to know how their data would be managed and used in the research.

6.4 Integrated Visualization

Social scientists often rely on methods such as questionnaires, surveys, interviews, and time-use diaries to gather information about practices at home or work. However, sensor-generated data are not always as straightforward as the data collected from those methods. To understand the sensor data and use them for social science studies, it is essential that the social researchers can watch the data in action and observe the patterns visually. For example, the social researchers in the HomeSense project found it useful to visually explore the data from various sensor modules in Kibana and cross validate their hypothesis with the activity recordings from the time-use diaries.

6.5 Design Trade-Off

Currently, MakeSense does not support recording audio or video information and mainly incorporates low-energy IoT devices (low data rate sensors). On the one hand, collecting, transmitting, storing, and visualizing audio/video data may pose technical challenges that need more sophisticated infrastructure support. On the other hand, capturing sensitive information such as audio/video in real-life settings such as homes or offices may raise serious privacy concerns and pose barriers for recruiting participants especially when children and guests are involved. Therefore, both technical and ethical issues should be reviewed when sensors such as microphones and cameras are being considered. We are planning to extend MakeSense to cater to such needs in the future.

6.6 Hardware and Software Testing

Some COTS sensors can only function correctly in a limited scenario, which may not be obvious beforehand. For example, having the IoTegg up against a wall or sitting on a table may influence the temperature and proximity sensors due to thermal effects and restricted location of power supply. The proximity sensors we use may also be severely affected by other sources of radiation (e.g., heat). It is important to identify the limitations of the sensors before deploying. Moreover, due to self-manufacturing with COTS components, there might be faulty parts in the hardware. Thus, it requires a test procedure before deployment. A simple procedure can be used to put the devices in a reference environment and compare the generated data with some reference data.

6.7 Software Development

Durability test is also important along with functional testing, because the devices are expected to run without manual intervention for a long period of time, and it would cost a lot of time and effort to repair or replace the device when it is malfunctioning. For example, in the beginning, some IoTeggs were found to be malfunctioning after days of operation without a clear indication about the software and hardware problems. To recover from and to debug such crashes, watchdogs and logging systems were implemented.

6.8 Cloud Operation

Concerning the cloud computing infrastructure, the platform is required to be highly available and reliable, so the design should be fail safe—that is, countermeasures should be in place if services

fail or lag behind. For example, we found out that logging and automatic re-spawning of services and/or message queue workers were very helpful in lowering manual intervention, improving reliability of data collection, and helping to debug issues.

6.9 Cost-Efficient Design

With the rapid development of IoT, a large number of electronic modules are available for researchers to build their own sensor suites. In addition, there are a great variety of COTS-sensing products on the market that provide plug-and-play sensing solutions. However, different electronic modules and commercial products often have a big difference in terms of cost. Even building one's own sensor suites, there are many choices of electronic modules for the same type of sensor with varying sensing capabilities such as accuracy, range, resolution, and response time. It is important to keep a balance between cost and quality. The development of the IoTegg was to provide reasonable sensing capabilities in the scope of indoor environments and in the meantime keep the cost within £50. Moreover, MakeSense tries to take advantage of open source projects, which not only enables researchers to fine tune the design but also reduces the overall cost.

6.10 Automated Setup and Deployment

Due to fieldwork experience of both the HomeSense project and the office monitoring project, a lot of time was devoted in designing measures to automate repetitive operations via scripting. This was useful because the IoTeggs needed to be deployed in large scale. For example, FOTA was provided and can be triggered with a one-line script. This script builds the newly updated software and deploys it on the target IoTeggs over WiFi and then reboots and installs the new software onto the IoTeggs, exploiting the external flash memory. Moreover, to reduce the technical requirements for using MakeSense, pre-built Docker images and scripts were provided to automate the server deployment process with simple commands in a Linux Shell.

7 CONCLUSION

In this article, we presented MakeSense, an IoT testbed to facilitate social studies of indoor activities. Through two use cases, we showed that the testbed is non-intrusive, cost efficient, and easy to deploy, safekeeping the security and privacy of the participants via data encryption and pseudonymization and capable of being adjusted to diverse experimental settings.

MakeSense allows remote monitoring and control, and supports extensible APIs for integrating COTS or self-developed sensor suites. The use of the IoTeggs benefits from widely available commodity sensors and industry standard WiFi/BLE radios, which provides flexibility and ease of setup in deployments. Flexible solutions for storage and visualization enable real-time monitoring and high-quality data collection.

Social research can benefit from the use of MakeSense to provide an additional layer of observation into people's activities in locations such as homes and offices. The data collected from the IoT devices can help the researchers understand the flow of indoor activities and validates their reasoning about the underlying practices, together with the information obtained from other methods such as time-use diaries. With its cost-efficient and flexible design, MakeSense enables researchers to set up experiments at a large scale and can be easily adapted to other indoor settings such as nursing homes, schools, restaurants, and shopping malls.

For future work, we plan to extend MakeSense with more types of sensors and actuators such that the users can easily adjust the testbed to their needs. We also plan to use MakeSense for other experiments in different settings such as healthy aging, in which we will research how IoT devices can be used to promote personalized interventions.

ACKNOWLEDGMENTS

The authors thank Dr. William Headley for the design and manufacture of the IoTEggs, and Dr. Krístrún Gunnarsdóttir for carrying out the field work for HomeSense and sharing the field work experience.

REFERENCES

- C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, et al. 2015. FIT IoT-LAB: A large scale open experimental IoT testbed. In *Proceedings of the 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT'15)*. 459–464. DOI : <https://doi.org/10.1109/WF-IoT.2015.7389098>
- Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. 2015. Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials* 17, 4 (Fourth quarter 2015), 2347–2376. DOI : <https://doi.org/10.1109/COMST.2015.2444095>
- Yousef Al Horr, Mohammed Arif, Amit Kaushik, Ahmed Mazroei, Martha Katafygiotou, and Esam Elsarraig. 2016. Occupant productivity and office indoor environment quality: A review of the literature. *Building and Environment* 105 (Aug. 2016), 369–389. DOI : <https://doi.org/10.1016/j.buildenv.2016.06.001>
- Oliver Amft and Gerhard Tröster. 2008. Recognition of dietary activity events using on-body sensors. *Artificial Intelligence in Medicine* 42, 2 (Feb. 2008), 121–136. DOI : <https://doi.org/10.1016/j.artmed.2007.11.007>
- Edwin Valarezo Añazco, Patricio Rivera Lopez, Sangmin Lee, Kyungmin Byun, and Tae-Seong Kim. 2018. Smoking activity recognition using a single wrist IMU and deep learning light. In *Proceedings of the 2nd International Conference on Digital Signal Processing (ICDSP'18)*. ACM, New York, NY, 48–51. DOI : <https://doi.org/10.1145/3193025.3193028>
- APDS-9960. 2018. SparkFun RGB and Gesture Sensor Breakout Board. Retrieved October 1, 2018 from <https://www.sparkfun.com/products/12787>.
- Rowland Atkinson and Keith Jacobs. 2016. *House, Home and Society*. Palgrave Macmillan, London, England.
- M. Chernyshev, Z. Baig, O. Bello, and S. Zeadally. 2018. Internet of Things (IoT): Research, simulators, and testbeds. *IEEE Internet of Things Journal* 5, 3 (June 2018), 1637–1647. DOI : <https://doi.org/10.1109/JIOT.2017.2786639>
- CoAP. 2018. The Constrained Application Protocol (CoAP). Retrieved October 1, 2018 from <https://tools.ietf.org/html/rfc7252>.
- Diane J. Cook, Aaron S. Crandall, Brian L. Thomas, and Narayanan C. Krishnan. 2013. CASAS: A smart home in a box. *Computer* 46 (July 2013), 62–69. DOI : <https://doi.org/10.1109/MC.2012.328>
- CurrentCost. 2018. Energy Monitor. Retrieved October 1, 2018 from <http://www.currentcost.com/products.html>.
- B. de Ruyter, E. Aarts, P. Markopoulos, and W. Ijsselstein. 2005. Ambient intelligence research in HomeLab: Engineering the user experience. In *Ambient Intelligence*, W. Weber, J. M. Rabaey, and E. Aarts (Eds.). Springer, Berlin, Germany, 49–61. DOI : https://doi.org/10.1007/3-540-27139-2_4
- Docker. 2018. Securely Build, Share and Run Any Application, Anywhere. Retrieved October 1, 2018 from <https://www.docker.com>.
- ElasticSearch. 2018. ElasticSearch Distributed, RESTful Search and Analytics Engine. Retrieved October 1, 2018 from <https://www.elastic.co/products/elasticsearch>.
- EmonPi. 2019. Open-Hardware Raspberry Pi-Based Energy Monitor. Retrieved May 1, 2019 from <https://wiki.openenergymonitor.org/index.php/EmonPi>.
- M. Espinilla, L. Martínez, J. Medina, and C. Nugent. 2018. The experience of developing the UJAmI smart lab. *IEEE Access* 6 (2018), 34631–34642. DOI : <https://doi.org/10.1109/ACCESS.2018.2849226>
- FINE. 2019. Support Innovative IoT Applications in the Smart City Domain. Retrieved May 1, 2019 from <http://fiesta-iot.eu/index.php/fiesta-testbeds/#5>.
- Fitbit. 2019. Fit for Everybody. Retrieved May 1, 2019 from <https://www.fitbit.com>.
- Flume. 2019a. Flume Data API. Retrieved October 28, 2019 from <https://flumetech.readme.io/docs/querying-samples>.
- Flume. 2019b. Flume Water Sensor. Retrieved October 28, 2019 from <https://flumetech.com>.
- Garmin. 2019. Garmin Wearables. Retrieved May 1, 2019 from <https://www.garmin.com>.
- Michelle L. Gattshall, Jo Ann Shoup, Julie A. Marshall, Lori A. Crane, and Paul A. Estabrooks. 2008. Validation of a survey instrument to assess home environments for physical activity and healthy eating in overweight children. *International Journal of Behavioral Nutrition and Physical Activity* 5, 1 (Jan. 2008), 1–13. DOI : <https://doi.org/10.1186/1479-5868-5-3>
- GP2Y0A60SZLF. 2018. Sharp Analog Distance Sensor (10–150cm). Retrieved October 1, 2018 from <https://www.pololu.com/product/2467>.
- Sumi Helal, William Mann, Hicham El-Zabadani, Jeffrey King, Youssef Kaddoura, and Erwin Jansen. 2005. The Gator tech smart house: A programmable pervasive space. *Computer* 38, 3 (March 2005), 50–60. DOI : <https://doi.org/10.1109/MC.2005.107>.
- HomeSense. 2018. Digital Sensors in Social Research. Retrieved October 1, 2018 from <http://sensors-in-social-research.net/>.

- HTU21D. 2018. SparkFun Humidity and Temperature Sensor Breakout Board. Retrieved October 1, 2018 from <https://www.sparkfun.com/products/retired/12064>.
- INMP401. 2018. SparkFun MEMS Microphone Breakout Board. Retrieved October 1, 2018 from <https://www.sparkfun.com/products/9868>.
- Stephen S. Intille, Kent Larson, Emmanuel Munguia Tapia, Jennifer S. Beaudin, Pallavi Kaushik, Jason Nawyn, and Randy Rockinson. 2006. Using a live-in laboratory for ubiquitous computing research. In *Pervasive Computing*, K. P. Fishkin, B. Schiele, P. Nixon, and A. Quigley (Eds.). Springer, Berlin, Germany, 349–365. DOI : https://doi.org/10.1007/11748625_22
- Valerie Issarny, Vivien Mallet, Kinh Nguyen, Pierre-Guillaume Raverdy, Fadwa Rebhi, and Raphael Ventura. 2016. Dos and don'ts in mobile phone sensing middleware: Learning from a large-scale experiment. In *Proceedings of the 2016 International Middleware Conference (ACM/IFIP/USENIX Middleware'16)*. DOI : <https://doi.org/10.1145/2988336.2988353>
- Jie Jiang, Riccardo Pozza, Kristrún Gunnarsdóttir, Nigel Gilbert, and Klaus Moessner. 2017. Using sensors to study home activities. *Journal of Sensor and Actuator Networks* 6, 4 (2017), Article 32. DOI : <https://doi.org/10.3390/jsan6040032>
- Jaime Jimenez, Michael Koster, and Hannes Tschofenig. 2016. IPSO Smart Objects. Retrieved October 1, 2018 from <https://www.iab.org/wp-content/LAB-uploads/2016/03/ips0-paper.pdf>.
- Kaiterra. 2019. Air Pollution Can't Hide. Retrieved May 1, 2019 from <https://www.kaiterra.com>.
- Jack Kelly and William Knottenbelt. 2015. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. *Scientific Data* 2 (March 2015), 1–14. DOI : <https://doi.org/10.1038/sdata.2015.7>
- Samad Barri Khojasteh, José R. Villar, Camelia Chira, Víctor M. González, and Enrique de la Cal. 2018. Improving fall detection using an on-wrist wearable accelerometer. *Sensors* 18, 5 (April 2018), 1–28. DOI : <https://doi.org/10.3390/s18051350>
- Kibana. 2018. Kibana Data Visualisation Engine. Retrieved October 1, 2018 from <https://www.elastic.co/products/kibana>.
- Julie A. Kientz, Shwetak N. Patel, Brian Jones, Ed Price, Elizabeth D. Mynatt, and Gregory D. Abowd. 2008. The Georgia tech aware home. In *CHI'08 Extended Abstracts on Human Factors in Computing Systems (CHI EA'08)*. ACM, New York, NY, 3675–3680. DOI : <https://doi.org/10.1145/1358628.1358911>
- Thomas Kleinberger, Martin Becker, Eric Ras, Andreas Holzinger, and Paul Müller. 2007. Ambient intelligence in assisted living: Enable elderly people to handle future interfaces. In *Universal Access in Human-Computer Interaction: Ambient Interaction*, C. Stephanidis (Ed.). Springer, Berlin, Germany, 103–112. DOI : https://doi.org/10.1007/978-3-540-73281-5_11
- B. Kröse, T. Van Kasteren, C. Gibson, and T. Van Den Dool. 2008. Care: Context awareness in residences for elderly. In *Proceedings of the Conference of the International Society for Gerontechnology*. https://pure.uva.nl/ws/files/4115026/59687_287355.pdf.
- Gierad Laput, Yang Zhang, and Chris Harrison. 2017. Synthetic sensors: Towards general-purpose sensing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI'17)*. ACM, New York, NY, 3986–3999. DOI : <https://doi.org/10.1145/3025453.3025773>
- Leshan. 2018. Eclipse.org Java Library for LWM2M. Retrieved October 1, 2018 from <https://www.eclipse.org/leshan>.
- Logstash. 2018. Centralize, Transform & Stash Your Data. Retrieved October 1, 2018 from <https://www.elastic.co/products/logstash>.
- Daniel López Gómez and Tomás Sánchez-Criado. 2015. Analysing hands-on-tech care work in telecare installations: Frictional encounters with gerontechnological designs. In *Aging and the Digital Life Course*, D. Prendergast and C. Garattini (Eds.). Berghahn Books, Oxford, UK, 179–197. DOI : <https://doi.org/10.2307/j.ctt9qdb6b.16>
- Jens Lundström, Wagner O. De Moraes, Maria Menezes, Celeste Gabrielli, João Bentes, Anita Sant'Anna, Jonathan Synnott, and Chris Nugent. 2016. Halmstad intelligent home: Capabilities and opportunities. In *Internet of Things Technologies for HealthCare*, M. U. Ahmed, S. Begum, and W. Raad (Eds.). Cham, Switzerland, 9–15.
- OMA LWM2M. 2018. Open Mobile Alliance Lightweight M2M Protocol. Retrieved October 1, 2018 from <https://www.omaspecworks.org/what-is-oma-specworks/iot/lightweight-m2m-lwm2m/>.
- Sinziana Mazilu, Ulf Blanke, and Gerhard Tröster. 2015. Gait, wrist, and sensors: Detecting freezing of gait in Parkinson's disease from wrist movement. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops'15)*. IEEE, Los Alamitos, CA, 579–584. DOI : <https://doi.org/10.1109/PERCOMW.2015.7134102>
- MongoDB. 2018. MongoDB, Open-Source Depdocument-Oriented Database Program. Retrieved October 1, 2018 from <https://www.mongodb.com/>.
- Maggie Mort, Celia Roberts, and Blanca Callén. 2013. Ageing with telecare: Care or coercion in austerity? *Sociology of Health and Illness* 35, 6 (July 2013), 799–812. DOI : <https://doi.org/10.1111/j.1467-9566.2012.01530.x>
- Jonathan Munoz, Fabian Rincon, Tengfei Chang, Xavier Vilajosana, Brecht Vermeulen, Thijs Walcarius, Wim van de Meerssche, and Thomas Watteyne. 2019. OpenTestBed: Poor man's IoT testbed. In *Proceedings of the Workshop on Computer and Networking Experimental Research Using Testbeds (IEEE INFOCOM-CNERT'19)*.
- M. Nati, A. Gluhak, H. Abangar, and W. Headley. 2013. SmartCampus: A user-centric testbed for Internet of Things experimentation. In *Proceedings of the 2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC'13)*. 1–6.

- Neurio. 2019. Intelligently Managing the Home's Energy. Retrieved May 1, 2019 from <https://www.neur.io>.
- Nuria Oliver and Eric Horvitz. 2005. S-SEER: Selective perception in a multimodal office activity recognition system. In *Machine Learning for Multimodal Interaction*, S. Bengio and H. Bourlard (Eds.). Springer, Berlin, Germany, 122–135. DOI: https://doi.org/10.1007/978-3-540-30568-2_11
- Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. 2014. Context aware computing for the Internet of Things: A survey. *IEEE Communications Surveys & Tutorials* 16, 1 (First quarter 2014), 414–454. DOI: <https://doi.org/10.1109/SURV.2013.042313.00197>
- James Pierce, Diane J. Schiano, and Eric Paulos. 2010. Home, habits, and energy: Examining domestic interactions and energy consumption. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'10)*. ACM, New York, NY, 1985–1994. DOI: <https://doi.org/10.1145/1753326.1753627>
- PostgreSQL. 2018. PostgreSQL Open Source Object-Relational Database. Retrieved October 1, 2018 from <https://www.postgresql.org/>.
- Deb Roy, Rupal Patel, Philip DeCamp, Rony Kubat, Michael Fleischman, Brandon Roy, Nikolaos Mavridis, et al. 2006. The Human Speechome Project. In *Symbol Grounding and Beyond*, P. Vogt, Y. Sugita, E. Tuci, and C. Nehaniv (Eds.). Germany, 192–196.
- Leah Ruppanner. 2015. *Contemporary Family Issues*. Oxford University Press, Oxford, UK.
- Luis Sánchez, Jorge Lanza, Juan Ramón Santana, Rachit Agarwal, Pierre Guillaume Raverdy, Tarek Elsaleh, Yasmin Fathy, et al. 2018. Federation of Internet of Things testbeds for the realization of a semantically-enabled multi-domain data marketplace. *Sensors* 18, 10 (2018), 3375.
- Luis Sanchez, Luis Munoz, Jose Antonio Galache, Pablo Sotres, Juan R. Santana, Veronica Gutierrez, Rajiv Ramdhany, et al. 2014. SmartSantander: IoT experimentation over a smart city testbed. *Computer Networks* 61 (2014), 217–238. DOI: <https://doi.org/10.1016/j.bjp.2013.12.020>
- Seeeduno. 2018. Arch-Pro Mbed Enabled Development Board. Retrieved October 1, 2018 from <https://developer.mbed.org/platforms/Seeeduno-Arch-Pro>.
- Peggy Shannon-Baker. 2016. Making paradigms meaningful in mixed methods research. *Journal of Mixed Methods Research* 10, 4 (Oct. 2016), 319–334. DOI: <https://doi.org/10.1177/1558689815575861>
- Marjorie Skubic, Gregory Alexander, Mihail Popescu, Marilyn Rantz, and James Keller. 2009. A smart home application to eldercare: Current status and lessons learned. *Technology and Health Care* 17, 3 (Aug. 2009), 183–201. DOI: <https://doi.org/10.3233/THC-2009-0551>
- Oriel Sullivan and Jonathan Gershuny. 2016. Change in spousal human capital and housework: A longitudinal analysis. *European Sociological Review* 32, 6 (Dec. 2016), 864–880. DOI: <https://doi.org/10.1093/esr/jcw043>
- Texas Instruments. 2018. SensorTag. Retrieved October 1, 2018 from http://www.ti.com/ww/en/wireless_connectivity/sensortag/index.html.
- Anne-Sophie Tonneau, Nathalie Mitton, and Julien Vandaele. 2015. How to choose an experimentation platform for wireless sensor networks? A survey on static and mobile wireless sensor network experimentation facilities. *Ad Hoc Networks* 30 (2015), 115–127. DOI: <https://doi.org/10.1016/j.adhoc.2015.03.002>
- Wakaama. 2018. Eclipse.org C Library for LWM2M. Retrieved October 1, 2018 from <https://www.eclipse.org/wakaama>.
- Jin Wang, Zhongqi Zhang, Bin Li, Sungyoung Lee, and R. Simon Sherratt. 2014. An enhanced fall detection system for elderly person monitoring using consumer home networks. *IEEE Transactions on Consumer Electronics* 60, 1 (Feb. 2014), 23–29. DOI: <https://doi.org/10.1109/TCE.2014.6780921>
- Jindong Wang, Yiqiang Chen, Shuji Hao, Xiaohui Peng, and Lisha Hu. 2018. Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters* 119 (2018), 2–11. DOI: <https://doi.org/10.1016/j.patrec.2018.02.010>
- Simon J. Williams, Catherine Coveney, and Robert Meadows. 2015. 'M-apping' sleep? Trends and transformations in the digital age. *Sociology of Health & Illness* 37, 7 (Sept. 2015), 1039–1054. DOI: <https://doi.org/10.1111/1467-9566.12283>
- X-Pack. 2018. Elastic X-Pack Extended Features Package. Retrieved October 1, 2018 from <https://www.elastic.co/guide/en/x-pack/current/xpack-introduction.html>.
- Xiaomi. 2018. Mi Band Wristband. Retrieved October 1, 2018 from <https://www.mi.com/en/miband/>.
- Ni Zhu, Tom Diethe, Massimo Camplani, Lili Tao, Alison Burrows, Niall Twomey, Dritan Kaleshi, Majid Mirmehdi, Peter Flach, and Ian Craddock. 2015. Bridging e-health and the Internet of Things: The SPHERE project. *IEEE Intelligent Systems* 30, 4 (July 2015), 39–46. DOI: <https://doi.org/10.1109/MIS.2015.57>

Received November 2018; revised November 2019; accepted January 2020