Smart*: An Open Data Set and Tools for Enabling Research in Sustainable Homes

Sean Barker, Aditya Mishra, David Irwin, Emmanuel Cecchet, and Prashant Shenoy
University of Massachusetts Amherst
{sbarker,adityam,irwin,cecchet,shenoy}@cs.umass.edu

Jeannie Albrecht
Williams College
jeannie@cs.williams.edu

ABSTRACT

The goal of the Smart* project is to optimize home energy consumption. As part of the project, we have designed and deployed a "live" system that continuously gathers a wide variety of environmental and operational data in three real homes. In contrast to prior work, our focus has been on sensing depth, i.e., collecting as much data as possible from each home, rather than breadth, i.e., collecting data from as many homes as possible. Our data captures many important aspects of the home environment, including average household electricity usage every second, as well as usage at every circuit and nearly every plug load, electricity generation data from on-site solar panels and wind turbines, outdoor weather data, temperature and humidity data in indoor rooms, and, finally, data for a range of important binary events, e.g., at wall switches, the HVAC system, doors, and from motion sensors. We also have electricity usage data every minute from 400 anonymous homes. This data corpus has served as the foundation for much of our recent research. In this paper, we describe our data sets as well as basic software tools we have developed to facilitate their collection. We are releasing both the data and tools publicly to the research community to foster future research on designing sustainable homes.

1. INTRODUCTION

The rise in energy prices over the last decade combined with growing fears over the impact of climate change has motivated recent research in the design of sustainable buildings and homes. Much of this research, including our own work in the Smart* (pronounced smart-star) project, is grounded in data gathered from the real world. As sensor networking researchers are well aware, deploying long-lived sensing systems poses a significant challenge. In particular, recent work highlights the unique and often overlooked challenges of designing *in situ* residential sensing deployments, which must blend into the home without compromising household aesthetics [4]. As a result, in many cases, researchers collect only the data they require for a specific project using a temporary, short-lived deployment. A disadvantage of the approach is that it may fail to capture aspects of the home that only reveal themselves over long periods. Further, researchers may be less likely to invest in scaling up project-specific deployments, since collecting data may have little value after a project's conclusion.

In our own research over the past three years, we have taken a different approach. Rather than deploying sensors and gathering data with a specific purpose in mind, we have instead concentrated on deploying a long-lived system for gathering a wide array of home data. This data has served as the foundation for many research projects [2, 7, 11, 12, 16, 17]. While our prior work summarizes aspects of our system and the data it collects, we have not yet provided a detailed description of either. Further, since we continuously work to improve our system's operation and data fidelity, including upgrading to better sensors, deploying additional sensors, and designing more efficient ways to query sensors, the descriptions that appear in much of our prior work is out-of-date. Since we have received an increasing number of requests for data, rather than continue to respond to each request individually, we have decided to create and maintain a public data repository. The lack of detailed public data sets has recently been cited as an impediment to academic research [10]. Our repository will include the data sets from this paper, as well as open-source Linux-based software we have developed for communicating with commercial power meters.

This paper's goal is to provide a detailed overview of our data, outline our choice of sensors and their capabilities, and describe the basic software tools we have developed to gather data. We also discuss some of the experiences and pitfalls in developing our system over the past three years. Since our deployment uses only commercially-available hardware, other researchers should be able to replicate our deployment with relative ease. The release of our data and software tools is inspired, in part, by the the Reference Energy Disaggregation Data Set (REDD), which is being widely used by researchers to validate and compare new disaggregation algorithms [9]. While our data may also prove useful for disaggregation research, both our goals and our data differ from REDD in important ways, as described below.

Heterogeneity. We collect data from a variety of different sources, including, but not limited to, electricity usage at the mains panel, each circuit, and nearly every plug load. We believe that correlating data from multiple sensors will prove useful for researchers. To this end, we also gather data from multiple weather, motion, door, wall switch, and thermostat sensors, as well as electricity generation data from solar panels and wind turbines. As one example of using multiple sensors, in recent work, we analyzed a refrigerator and used both its internal temperature and its average real power each second to quantify the cooling rate of the compressor [2].

Scalability. Our system's aim is to collect high-resolution data at scale. For electrical loads, we gather average real power each second for entire homes and each circuit, and average real power from almost every individual plug load every few seconds. We also record on/off/dim events from nearly every wall switch in one home. Additionally, our sensors at the mains panel record apparent power for the home and each circuit, as well as the voltage and frequency on both phases of the home's split-leg input power. We believe collecting data from *every* load will enable new research. For instance, we are not aware of any prior work on non-intrusive load monitoring (NILM) that focuses on large scale scenarios—greater than 100 loads—with many relatively low-power loads, e.g., less than 50 watts (W), which is a common characteristic of modern

homes. The lack of research may be due, in part, to the difficulty in providing ground truth data across many loads. As prior research on NILM has shown, reactive power is often useful in disaggregation [3]. While past work has asserted that distributed sensing at every load is prohibitively expensive [8], our deployment demonstrates that the cost is well within the bounds of a modest research budget, e.g., a few thousand U.S. dollars.

Redundancy. Instrumenting the same loads multiple times at different levels of the electrical wiring tree—the entire home, each circuit, and each wall switch and outlet—reveals important information about the relative accuracy of the sensors. In our own experience, we have found that sensors may exhibit errors that are difficult to detect, but have a significant impact on the conclusions drawn from fine-grained data. For example, we discovered that the widely-used Energy Detective (TED) power meter [18] for monitoring a home's electricity at the mains panel sometimes experiences communication problems while sending data over the powerline. The meter uses an unreliable X10-like protocol that is highly sensitive to noise on the powerline. While the display blinks orange when the problems occur, the data masks the problem by always recording the last power reading as the current power reading if it does not receive a new reading. We only discovered the errors when correlating our readings with data from meters at outlets and wall switches. As we indicate in prior work [7], the hidden communication problems complicate disaggregation.

2. SMART* PROJECT OVERVIEW

As part of the Smart* project, we have built a data collection infrastructure that records data from a variety of sensors deployed in real homes. Our infrastructure supports both pulling data by querying individual sensors, and pushing data from sensors to a gateway server, which runs our software tools. The infrastructure has support for tracking (i) average real and apparent power every second for the home and each circuit at the mains panel, (ii) real power usage every few seconds from nearly all of the home's plug loads, (iii) on-off-dim events at nearly all of the home's wall switches, (iv) average electricity generation from solar panels and micro wind turbines every five seconds, (v) a variety of events related to energy consumption, including motion sensing, door/trigger sensing, and thermostat sensors, and (vi) environmental data every minute via weather sensors both inside and outside the home.

Our data collection infrastructure provides a web interface to configure devices in each home and control the data gathering process. Importantly, our infrastructure is designed as a "live" system that operates continuously for an indefinite period. Thus, we expect to release periodic snapshots of our data in the future, starting with an initial release in August 2012. We are targeting a release every 6 months of the previous 6 months worth of data. The long-term nature of the deployment should provide researchers a window into how electricity usage changes season-to-season and year-to-year.

3. SMART* OPEN DATA SETS

Our initial release consists of two data sets: (i) a high-resolution data set from three homes and (ii) a lower resolution data set from 400 homes. We refer to the former as the *UMass Smart* Home Data Set* and the latter as the *UMass Smart* Microgrid Data Set*, and request that researchers cite these names in their work.

3.1 Smart* Home Dataset

While other researchers have targeted breadth in collecting data, e.g., gathering household electricity from many more homes [1, 9], our deployments target depth by gathering a multitude of data from many sensors in three real homes. We first briefly describe each home before describing the various types of data we collect. Since each home includes a different mix of sensors, we also outline which sensors are deployed in which homes. While we do not reveal the exact location of the homes, they are all in Western Massachusetts.

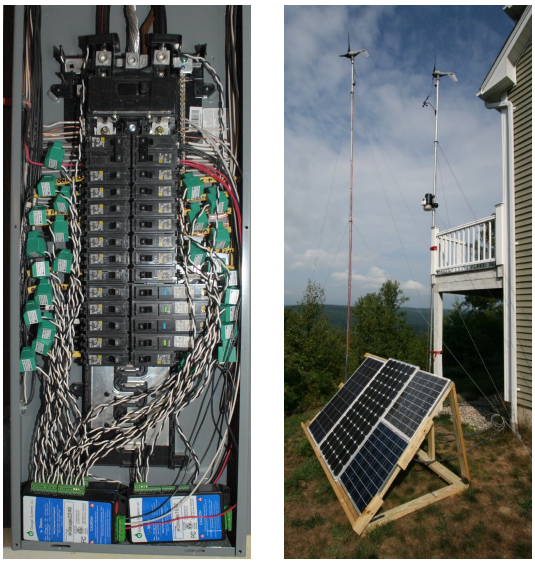
3.1.1 Deployment

Home A is a two-story, 1700 square foot home with three full-time occupants. The home has a total of eight rooms including its basement. The main level has a living room, bedroom, kitchen, and bathroom, while the second story has two bedrooms and a bathroom. The home does not have central air conditioning (A/C). In the summer, the occupants use three window A/C units: one in the living room and one in each of the upstairs bedrooms. The home's heating system uses natural gas. Other major appliances include an electric dryer and washing machine, heat recovery ventilation (HRV) unit, dishwasher, refrigerator, and freezer. The home has 35 wall switches, which primarily control room and closet lighting; switches also control an exhaust fan in each bathroom and the garbage disposal. The electrical panel has 26 individual circuits.

Home A is our most deeply instrumented home. Using sensors installed in the mains panel, we collect electricity data every second for the entire home, as well as each circuit. We have replaced 30 of the home's 35 wall switches with units that transmit on-off-dim events for the switches over the powerline to a gateway server. We were unable to replace the remaining five switches for various reasons: three basement switches do not have neutral wires in the switch box, the garbage disposal's power exceeded the rating of the programmatic switches, and an exact replacement for one kitchen switch is not available. We are able to derive the power usage from the uninstrumented switches via the circuit data: the basements switches are on dedicated circuits, the garbage disposal is on a circuit with only the dishwasher (which has a dramatically different power profile), and the kitchen switch is on a circuit dedicated to kitchen lights, which has only one other already-instrumeted load. The home's electrical wiring also aids our data collection. Each circuit is dedicated to either lighting (monitored at wall switches), outlets (monitored by plug meters), or individual large appliances (monitored at the mains panel). Since our wall switches report on-off-dim events, rather than raw power, having the lighting on separate circuits makes it simple to correlate lighting events with power usage using the circuit data.

In addition to monitoring characteristics of electricity usage, Home A also includes a variety of other sensors. The home's heating system has three zones controlled by three thermostats—-one in the living room and one in each upstairs bedroom. We have installed thermostats that transmit information about the heating system to our gateway server, including when the furnace turns on or off, when the setpoint rises or falls, e.g., from an occupant changing it manually, and when the temperature changes. We have also deployed motion sensors in all eight rooms that signal when motion is detected and when it is no longer detected, i.e., if two minutes of idle time occurs from the last motion. We have also deployed door sensors that report open and close events. Our initial door sensor deployment is small: we have attached two sensors to the refrigerator (for the refrigerator and freezer compartments) and one door sensor to the basement freezer. Finally, we have deployed a weather station that collects both outdoor—temperature, humidity, pressure, wind speed, rainfall, solar intensity, etc.—and indoor weather statistics. We have deployed temperature/humidity sensors in all eight rooms, as well as inside the refrigerator.

Home B is similar to Home A in size, at roughly 1700 square feet



(a) (b)

Figure 1: CT installation at the mains panel in Home A (a); solar panel and micro wind turbines at Home C (b).

across two stories with eight rooms and four full-time occupants. The primary level includes a living room, kitchen, and dining room, while the second story includes two bedrooms. The home also has a finished basement and two bathrooms. Unlike Home A, Home B has central A/C, in addition to a gas-powered heating system. Similar to Home A, we have installed sensors in Home B's mains panel, which records electricity usage every second for the entire home and for all 21 circuits. Home B also has a weather station for gathering outdoor weather statistics, although we do not currently monitor indoor weather. Home B's heating and cooling system uses a single, centrally located thermostat. Rather than deploy the same thermostat as in Home A, we have deployed the recently released NEST thermostat in the home. The NEST makes the same basic information available for logging as Home A's thermostat, but uses WiFi for communication (rather than a custom 900Mhz wireless protocol) and includes logic for learning behavioral patterns from a built-in motion sensor and autonomously altering the setpoint.

Home C is much larger than Home's A and B, at roughly 3500 square feet across two stories. Due to Home C's size, it requires two separate electrical panels with a total of 60 circuits. We currently monitor electricity usage for the entire home, as well as 21 of its circuits, in addition to outdoor weather statistics via a weather station. Unlike Home's A and B, Home C has both solar panels and (until recently) two micro wind turbines. We are currently redeploying the turbines to a new location. We record current from the three solar panels and two micro wind turbines, at five second intervals (averaged from samples every second), along with the battery voltage. The generation uses micro-inverters at each panel to record individual current, convert the power to AC, synchronize it with the grid, and feed it into the home's grid supply. The home net meters its power onto the grid via an electrical meter that reverses direction when the home's generation exceeds its consumption. The micro-inverters are designed to disengage power if grid power goes out, to prevent backfeeding onto dead power lines.

3.1.2 Data Types

Below, we provide detailed information about our sensors and the data they collect. We use a simple nomenclature for sensor IDs, that incorporates their location (by room and, if applicable, by circuit) and, in some cases, their functionality. For example, we pre-pend a unique number to the name of each circuit we monitor. We then pre-pend the appropriate circuit number to each individual load we monitor (at wall switches and plugs) to indicate which circuit the load is attached to. Similarly, we use standard names for rooms within each home, and include the name in a data item's ID. Tables 1 and 2 summarize the data we collect. Each row in the tables represents a distinct file storing time-series data. We expect to release all data as a set of simple space-delimited text files (one per day) storing the time-series with Unix (UTC) timestamps.

Name	*Timestamp*	*Data Fields*
Circuit ID	Unix	Real Power (W), Apparent Power (VA)
Phase ID	Unix	Frequency (Hz), Voltage (V)
Panel ID	Unix	Current (Amps)
Turbine ID	Unix	Real Power (W)
Battery	Unix	Voltage (V)

Table 1: We collect electricity usage every second at the mains panel and electricity generation data every five seconds from solar panels and wind turbines. Each row represents the row format for time-series data in a distinct data file.

Name	*Timestamp*	*Data Fields*
Meter ID	Unix	Real Power (W), Circuit, Room
Switch ID	Unix	MaxPower (W), Dim%, Circuit, Room
Thermostat ID	Unix	On/Off (1 | 0), Temp (F), Setpoint (F)
Motion ID	Unix	Yes/No (1 | 0), Room
Door ID	Unix	Open/Close (1 | 0)

Table 2: We collect data from plug meters, wall switches, motion sensors, door sensors, and thermostats. Each row represents the row format for time-series data in a distinct data file.

Electricity at the Mains Panel. Numerous commercial power meters are available, such as the TED 5000, BrulTech ECM-1240, Current Cost Envi, and eGauge. These meters sense electricity usage using current transducers (CTs) wrapped around each leg of a home's split-phase input power. Our system is compatible with any of these meters, which generally make the data available via the web. Since these meters typically have slots for a limited number of CTs, e.g., the TED supports six CTs per gateway, we use multiple units in each home's panel to cover as many circuits as possible. Figure 1(a) shows Home A's installation, with three units in the bottom of the panel. We use 100A CTs for each leg of power, and 20A CTs for each circuit, matching the breaker ratings. The CTs are rated to have less than 1% error for current and voltage. Redundant monitoring of both the home's aggregate data and every circuit allows us to determine the relative error of the sensors, by comparing the aggregate usage with the sum of all circuits' usage. Figure 2 demonstrates that over 90% of the per-second readings for the entire home and the sum of the circuits is within 2% of each other, while over 99% of readings are within 4% of each other.

For transmitting data to our gateway server, our unit uses the HomePlug Ethernet-over-Powerline protocol, which is designed for high-bandwidth applications like HDTV. For each CT, we record both real and apparent power. Additionally, the unit is able to record voltage and frequency every second on both phases. Onboard flash in each unit stores the last ten minutes of per-second data, requiring our system to only query the device once every 10 minutes. As mentioned above, we pre-pend to each circuit a unique number that we use across devices. For example, the electric dryer's dedicated circuit in Home A is "02:Dryer". The first two rows of Table 1 summarize the data we collect at the mains panel. The first row's data includes real and apparent power for all circuits (in Home's A and B) and the entire home. The second row's data includes frequency and voltage for both electrical phases.

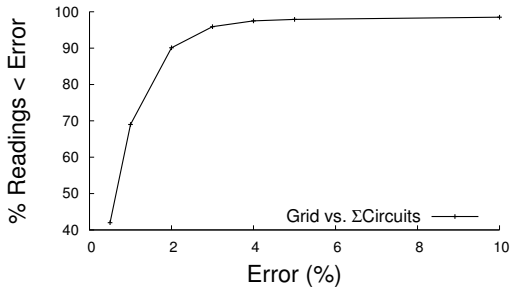


Figure 2: Error between Home A's aggregate electricity data and the sum of all the individual circuits

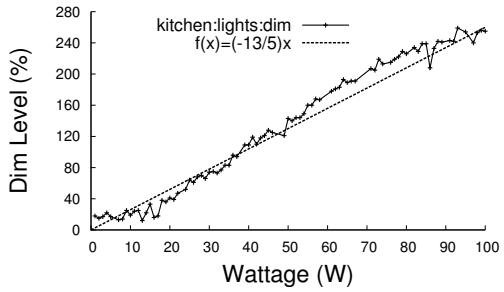


Figure 3: Power usage is a linear function of a light's dim level.

Renewable Generation. Figure 1(b) shows our solar panel and wind turbine deployment at Home C. As stated earlier, we recently took down our wind turbines for relocation. We use a HOBOlink data logger [5] to record the average current from the panels and turbines every five seconds, as well as the attached battery's voltage. Air-X manufactures our wind turbine, which rates its maximum power output as 400 watts in 28 mile per hour winds. The last three rows of Table 1 describe the generation data. Similar to the electricity usage data, we expect to release generation data as simple space-delimited text files with each entry containing a timestamp and the current or voltage over the last five seconds.

Electricity at Outlets. Numerous commercial plug meters are now available. We use two different meters in our deployment: the Insteon iMeter Solo [6] and the Z-Wave Smart Energy Switch [20] from Aeon Labs. The iMeter Solo uses the Insteon protocol to transmit readings to our gateway server via an Insteon Powerline Modem (PLM). The Insteon protocol simulcasts readings wirelessly and over the powerline. While the iMeter Solo's data packets are undocumented, the protocol's simplicity allowed us to reverse engineer it. One disadvantage of the Insteon protocol is its extreme bandwidth limitations, which permits a maximum of one iMeter Solo power query across all devices each second. In practice, since we employ other Insteon sensors, we do not query iMeters at the maximum rate; instead, we query an iMeter only if we detect that power has changed on its circuit. The approach works well for stable loads that rarely change their power usage, e.g., digital clocks, lamps, etc. We currently have 34 iMeter Solos in Home A.

iMeters are not appropriate for loads with highly variable power usage, such as some electronic, e.g., TVs, computers, etc., or inductive, e.g. A/Cs, refrigerators, vacuums, etc., loads. We use Z-Wave wireless Smart Energy Switches to monitor power for these loads. While Z-Wave is a proprietary protocol, the OpenZWave library for Linux provides rudimentary APIs for many Z-Wave devices. Since the project only provides libraries, device-specific code for gathering data or controlling devices must be custom written. As we discuss in the next section, we have written a simple driver for the Smart Energy Switch to query the switches (specified in a configuration file) in serial order for their power usage. We currently have 21 Z-Wave switches in Home A, which record real power from each switch on average every 2.5 seconds.

Our system covers nearly all plug loads in Home A. However, we currently do not monitor a few small loads, our measurement equipment, or transient loads not permanently connected to fixed outlets. The only significant loads we do not monitor individually in Home A are wired to gas furnace. In particular, an exhaust fan for the boiler and multiple recirculator pumps for the baseboard heating system. As we discuss below, we are exploring how to disaggregate these loads using data from Home A's thermostats. The first row of Table 2 shows the data format for the plug meters.

Wall Switch Events. In Home A, as discussed earlier, we have replaced 30 of the 35 mechanical wall switches with Insteon-enabled switches [6]. Specifically, we use Insteon SwitchLinc Relays for non-dimmable lights and Insteon SwitchLinc Dimmers for dimmable lights. The wall switches transmit discrete on/off/dim events for each switch when physically pressed to the gateway via the Insteon PLM. The dim level is reported as a percentage between 0 and 100, with 100% being fully on. As Figure 3 shows, lighting power is a linear function of reported dim level.

Since lighting in Home A is on separate circuits from outlets and other appliances, it is simple to correlate wall switch events with power usage collected by circuit meters. However, if we ever detect a change on a lighting circuit meter without having detected a wall switch event, our gateway is also able to poll the switches for their on-off-dim status. Importantly, the Insteon wall switches look and behave like normal switches, so they are not obtrusive to the homeowners. In Home A, the new switches look exactly like the old switches, and have been in use for over a year. The second row of Table 2 describes our wall switch data.

Thermostat Events. We use two different types of thermostats for monitoring home heating and cooling systems. In Home A, we use three Insteon-enabled Ventstar thermostats [6] that wirelessly transmit data via the Insteon PLM. The thermostat sends messages to our gateway server whenever someone manually changes the setpoint temperature, the temperature changes, or the furnace turns on or off. The thermostat's mode and setpoint is also configurable from the gateway. Since the furnace turning on or off correlates with the operation of the furnace's hot water recirculator pumps, we are optimistic that the furnace data will aid us in disaggregating the furnace loads that we are unable to individually meter. In Home B, we use the NEST thermostat, which has similar monitoring and control functions as the Ventstar but is WiFi-enabled and capable of autonomously controlling the setpoint based on occupancy patterns it learns over time using a built-in motion sensor. The third row of Table 2 describes our thermostat data.

Motion Events. We use Insteon Skylink motion sensors [6] to monitor occupancy in six rooms of Home A. The motion sensors send our gateway server information when (a) activity is detected in a previously dormant room, or (b) activity has not been detected for two minutes in a previously active room. We record the information for each room, which includes a timestamp when motion is detected (or not), as well as the new occupancy status of the room. The fourth row of Table 2 describes our motion sensor data.

Door Events. We use Insteon TriggerLincs [6] to monitor when doors open and close. As with the motion sensors, we record the time of the opening and closing. Thus far, we have only deployed door sensors on the kitchen refrigerator, its freezer compartment, and the basement freezer. However, even this small amount of data has proven useful, since we are able to track how often the freezer and refrigerator open, which has a significant effect on the interior temperature and the resulting compressor cycle. Further, opening the door triggers an interior light, which we correlate with 120W

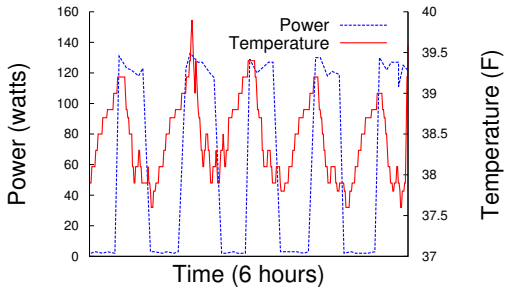


Figure 4: Power and temperature data demonstrating the 'guardband' behavior of a refrigerator. Data from [2].

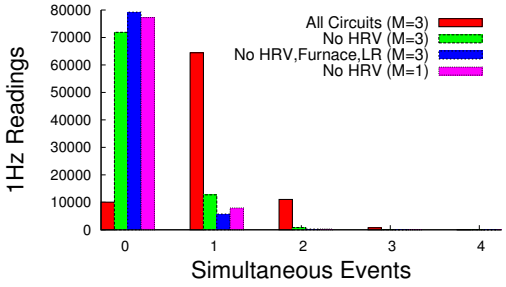


Figure 5: Histogram of concurrent power events in Home A.

power spikes in the refrigerator and freezer's circuit data. We plan on deploying additional sensors at interior and exterior doors in the near future. The fifth row of Table 2 describes our door sensor data.

Weather Station Data. Finally, we use an Oregon Scientific WMR 200A professional weather station [14] to monitor indoor and outdoor weather data. We deploy the weather station's rain gauge, anemometer, and temperature/humidity sensors on a pole mounted in the rear of Homes A, B, and C. In Home A, we also use eight additional temperature/humidity sensors to monitor the status of the three rooms on the main level, the two upstairs bedrooms, the basement, and the interior of the refrigerator. We record the average of the temperature, humidity, wind, and rainfall metrics every minute.

3.2 Smart* Microgrid Data Set

In addition to our live dataset from the three homes we are monitoring, we also plan to release a second data set from 400 homes, which includes average real power usage for each home at one minute granularity for an entire day. For privacy reasons, the data source and the homes are kept anonymous. This data is well-suited for emulating microgrids or examining the grid-scale effects of various optimizations, such as the use of energy storage [11]. We expect to include data for longer periods in a future release.

3.3 Potential Uses

Below, we outline past and potential future uses of our data set.

Cost Optimization. SmartCharge uses energy storage to cut electric bills when using market-based electricity pricing plans [11]. In this work, we used Home A's aggregate electricity data to quantify the potential for savings using today's market-based pricing plans and batteries. We also developed a machine learning-based model to predict aggregate consumption for Home A using multiple features, including the weather data we collect. Finally, we used our microgrid data to quantify the effect of energy storage at grid-scale.

Demand Flattening. SmartCap flattens electricity demand by shifting electricity usage for background loads without impacting their objective, e.g., to maintain an environmental setpoint or complete a task [2]. We designed a Least Slack First (LSF) scheduling policy, which schedules loads in ascending order of their remaining slack—the time which they may remain off without affecting their objective. We used our home electricity data, plug load and circuit data for eight background loads, and our temperature and humidity data from our weather sensors to evaluate LSF's potential for demand flattening. For example, Figure 4 is a graph from our work [2] that shows how the refrigerator's interior temperature correlates with its power usage. In this case, the work exploited slack in the compressor cycle, e.g., when the temperature was between its maximum and minimum point, to advance or defer the compressor.

Load Monitoring. Home automation (HA) protocols, such as Insteon, are designed to provide low-cost remote actuation capabilities for loads. Unfortunately, their low bandwidth has precluded their use in load monitoring. We design AutoMeter to disaggregate a home's electricity usage each second using low resolution data from Insteon wall switch events and iMeter plug loads [7]. The approach endows low-cost HA systems originally designed for actuation with new sophisticated load monitoring capabilities.

Renewable Prediction. We have designed multiple models for predicting future renewable generation using weather forecasts from the National Weather Service [17, 16, 19], and used them to improve the performance of a variety of systems. Our first model used solar radiation and wind speed from our weather station to predict solar and wind generation, respectively. We then developed more sophisticated models using machine learning techniques that included a variety of other forecast features.

Privacy. Finally, we have studied how our homes' aggregate electricity leaks information about the activities of its occupants, and defined privacy-preserving protocols that enable utilities to bill for usage without revealing occupant behavior [12].

NILM. Similar to REDD, our data might also be useful in developing and evaluating new disaggregation algorithms for electricity data. For example, Figure 5 shows a histogram of the number of per-second readings that fall within a concurrent power event across all circuits in Home A on a single day. One challenge with NILM is disaggregating loads when they simultaneously change power. In this case, we define an event as a change in power greater than 10W, with an associated margin M of either 0 seconds or 3 seconds that determines the duration (or 'length') of an event. If a change occurs at time T and the margin is M seconds, then any other change in power is concurrent if it occurs between $T + M$ and $T - M$. Figure 5 shows that the vast majority of per-second readings, e.g., $x = 0$ or $x = 1$, are not part of concurrent events (with a threshold of 10W and a margin of 3 seconds).

While the number of readings that fall within concurrent events ($x \geq 2$) is approximately 10,000 for $M = 3$, most of these are caused by a small number of highly variable circuits. If we remove the HRV circuit, the number of readings falls by more than 10x to approximately 900, while removing the furnace and living room outlets results in an additional 3x reduction to roughly 300 readings. This data suggests that either a few strategically placed power meters can simplify NILM, or that pure NILM algorithms might benefit from focusing on these few highly variable devices. Additionally, we can demonstrate the benefits of using higher-precision sensors by decreasing the margin – for example, by removing only the HRV circuit and setting $M = 1$ (as shown in Figure 5), we can achieve under 200 concurrent readings.

4. SMART* SOFTWARE TOOLS

Since our goal is to collect data in support of research in sustainable buildings, rather than researching how to build new types of sensor systems, we use only commodity, off-the-shelf equipment in our deployment. As a result, researchers are able to replicate

our deployment without building custom hardware. We encourage new deployments and data releases, since it is infeasible for us to deeply instrument a large number of houses on our own. In this section, we provide a high-level overview of our system, as well as describe some Linux-based software tools we have developed to interact with sensors. We include the software tools as part of our public release to facilitate researchers in replicating our system.

System Overview. Smart* employs a simple architecture centered around a gateway server in each home. We use the embedded DreamPlug server as our gateway. The DreamPlug has a low, inconspicuous profile, runs a standard Linux distribution, and has numerous USB ports for sensors: the Insteon PLM, Z-Wave Z-Stick2 receiver, and WMR 200A Weather Station console all connect via USB ports. Our system is device agnostic: it supports the sensors we currently use, as well as those we have used in the past, e.g., the TED, DavisPro, Tweet-a-Watt. We plan to support more devices as necessary. In many cases, interacting with sensors is straightforward. Power meters that install in the electrical panel generally make the data available via the web, and do not require specialized software. The Linux software `wview` supports a wide variety of weather stations, including the Oregon Scientific and DavisPro models. Unfortunately, Insteon and Z-Wave do not have mature open-source software tools available. Below, we describe the tools we have developed to interact with these sensors. In the future, we may also release our data collection engine, which uses these basic tools as building blocks to collect our data in real-time.

Insteon Software Tools. Communication with Insteon devices is accomplished through the use of an Insteon PLM, which connects both to the Insteon network over the powerline and to our gateway over a USB serial connection. The most robust existing toolset for communicating with the PLM in Linux is the open-source `plmtools` project [15], which listens on the PLM's USB serial connection to send and receive binary data from the Insteon network. Our fork of `plmtools`, which we call `plmtools-imeter`, is updated and extended in several ways to make it more useful for large-scale sensing deployments. Our most significant addition is support for the Insteon iMeter Solo. The iMeter Solo's commercial software communicates only with proprietary, Windows-only software using an undocumented protocol, which we have reverse engineered for `plmtools-imeter`.

Our software turns the iMeter Solo into an easily scriptable meter which can be queried using a simple, one line Linux command. We have also extended `plmtools` in several other ways, such as adding more robust error handling (which is important given the potential for powerline packet collisions), human-readable descriptions of observed packets in real-time, and the decoupling of packet deliveries from receipts. The latter enhancement allows, for example, a single process to receive and process all incoming packets, while other processes asynchronously dispatch commands over the power line. This is useful when simultaneously listening for interrupts (such as from Insteon wall switches or motion sensors) and dispatching commands (such as iMeter Solo queries).

Z-Wave Software Tools. Communication with Z-Wave devices occurs wirelessly using a small USB receiver. Existing Z-Wave support is similar to the iMeter Solo, in that its official software is proprietary. The OpenZWave project [13] is a new project working towards open-source support for Z-Wave devices, which currently includes rudimentary documentation and examples. To use our Z-Wave Smart Energy Switch, we have written a small set of programs to mask the details of OpenZWave and directly query the meter. The program enables us to use Z-Wave meters as drop-in replacements for iMeters, when we require high bandwidth for rapid data collection. Our Z-Wave daemon simply queries every Z-Wave meter in range in round-robin fashion with a user-defined delay between queries, which determines the data collection rate.

5. CONCLUSION AND FUTURE PLANS

This paper describes two datasets—the UMass Smart* Home Data Set and the Smart* Microgrid Data Set—-we are releasing as part of the Smart* project to foster research in designing sustainable homes. We expect to issue periodic releases of the Smart* Home data every six months. Finally, we are continuing to improve our existing deployments, and plan to add additional sensors and data products to future releases.[1]

6. REFERENCES

[1] O. Ardakanian, S. Keshav, and C. Rosenberg. Markovian Models for Home Electricity Consumption. In *GreenNets*, August 2011.

[2] S. Barker, A. Mishra, D. Irwin, P. Shenoy, and J. Albrecht. SmartCap: Flattening Peak Electricity Demand in Smart Homes. In *PerCom*, March 2012.

[3] G. Hart. Residential Energy Monitoring and Computerized Surveillance via Utility Power Flows. *IEEE Technology and Society Magazine*, 8(2), June 1989.

[4] T. Hnat, V. Srinivasan, J. Lu, T. Sookoor, R. Dawson, J. Stankovic, and K. Whitehouse. The Hitchhiker's Guide to Successful Residential Sensing Deployments. In *SenSys*, November 2011.

[5] `https://www.hobolink.com/`, June 2012.

[6] `http://www.insteon.net/`, June 2012.

[7] D. Irwin, A. Wu, S. Barker, A. Mishra, P. Shenoy, and J. Albrecht. Exploiting Home Automation Protocols for Load Monitoring in Smart Buildings. In *BuildSys*, 2011.

[8] Y. Kim, T. Schmid, Z. Charbiwala, and M. Srivastava. Viridiscope: Design and Implementation of a Fine Grained Power Monitoring System for Homes. In *UbiComp*, 2009.

[9] J. Kolter and M. Johnson. REDD: A Public Data Set for Energy Disaggregation Research. In *SustKDD*, August 2011.

[10] J. Markoff. Troves of Personal Data, Forbidden to Researchers. In *New York Times*, May 21st 2012.

[11] A. Mishra, D. Irwin, P. Shenoy, J. Kurose, and T. Zhu. SmartCharge: Cutting the Electricity Bill in Smart Homes with Energy Storage. In *e-Energy*, May 2012.

[12] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin. Private Memoirs of a Smart Meter. In *BuildSys*, November 2010.

[13] `http://www.openzwave.com/`, June 2012.

[14] `http://www.oregonscientific.com/`, June 2012.

[15] `http://plmtools.sourceforge.net/`, June 2011.

[16] N. Sharma, J. Gummeson, D. Irwin, and P. Shenoy. Cloudy Computing: Leveraging Weather Forecasts in Energy Harvesting Sensor Systems. In *SECON*, June 2010.

[17] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy. Predicting Solar Generation from Weather Forecasts Using Machine Learning. In *SmartGridComm*, October 2011.

[18] `http://www.theenergydetective.com/`, June 2012.

[19] T. Zhu, A. Mishra, D. Irwin, N. Sharma, D. Towsley, and P. Sheony. The Case for Efficient Renewable Energy Management for Smart Homes. In *BuildSys*, November 2011.

[20] `http://www.aeon-labs.com/site/products/view/5/`, June 2012.

[1]Research supported by NSF grants CNS-1143655, CNS-0916577, CNS-0855128, CNS-0834243, CNS-0845349