Providing households with real-time feedback from the monitoring of energy consumption and generation

John Pearce

This report is submitted as partial fulfilment of the requirements for the Higher Degree by Research Preliminary Programme of the School of Electrical, Electronic and Computer Engineering, The University of Western Australia, 2015

Abstract

Reducing energy consumption and the use of renewable energy sources have become increasingly important. While the public have been taking steps to reduce energy consumption and switch over to more energy efficient appliances, there is a limit to the possible level of energy reduction using this approach. Making more effective use of the variable generation of renewable energy sources requires a shift in patterns of consumption to more closely follow generation. However, typically occupants are unaware of their energy usage until the end of each billing period.

There are a number of commonly used appliances that contribute significantly to household energy consumption and for which time of use could be shifted without significant inconvenience, such as washing-machines, dishwashers and tumble-dryers. A system capable of monitoring and processing data for household energy consumption, generation and storage in real-time could provide useful feedback to occupants leading to a reduction in both overall energy consumption and that from the power grid.

This exploratory study focuses the development of a system to monitor energy consumption and generation for households with roof-top solar photovoltaic (PV) systems and to use that data to drive shifts in energy consumption to better match generation. We present the development of a system to ingest, process and visualise energy consumption and generation data. Using this system we implement a processor to predict, in the short-term, periods of lower than usual energy generation and generate notifications for household occupants to suggest deferring using elastic-load appliances. The model of a battery energy storage system created is presented and used to analyse the viability of incorporating battery energy storage into household solar PV systems.

Keywords: energy monitoring, domestic, prediction, solar photovoltaic, visualisation, notification

Acknowledgements

I would like to formally acknowledge the following people, who have made this thesis and project possible.

Prof. Thomas Bräunl, for the opportunity by providing this project and his guidance throughout.

Peter Bulanyi and Dennis Darcy, of Si Clean Energy, for their support with access and integration of the AllSolus Access platform and the UWA Future Farm.

Contents

Abstract **ii**

Acknowledgements **iii**

1 Introduction **1**

1.1 Background 2

1.1.1 Overview 2

1.1.2 Solar Photovoltaic Generation 3

1.1.3 Energy Display & Feedback 4

1.1.4 Predicting Energy Consumption 4

1.1.5 Battery Energy Storage Systems 5

1.1.6 Energy Disaggregation 5

1.1.7 Tariffs 7

1.2 Monitoring Locations 8

1.2.1 Off-grid Solar PV with storage 8

1.2.2 Grid-tied Solar PV 9

2 Aims **10**

2.1 Problem Statement and Motivation 10

3 System Design **11**

3.1 Requirements Analysis 11

3.1.1 Functional 11

3.1.2 Non-functional 12

3.2 System Components 12

3.2.1 Platform 13

3.2.2 Task Controller 13

3.2.3 Ingest and Processing Nodes 14
3.2.4 Message Broker 15
3.2.5 Storage 15
3.2.6 Dashboard 16
3.2.7 Notifications 17
3.3 Metrics Calculations 18
3.3.1 Normalising Periodic Measurements 18
3.3.2 Energy From Power 19
3.3.3 Always-on Consumption 19
3.3.4 Clouded Irradiance 20

4 Battery Model 22

4.1 Design 22
4.1.1 Assumptions 23
4.1.2 Parameters 23
4.2 Simulation 24

5 Results & Analysis 27

5.1 Energy Usage 27
5.2 Energy Prediction 29
5.3 Battery Storage 30
5.4 Costs 32
5.5 Energy Disaggregation 34
5.6 Limitations 35

6 Conclusion 37

6.1 Future Work 37
6.2 Conclusion 37

A System Design Details 39

B Household Appliance Usage 42

C InfluxDB Time-series **44**

D Battery Model Code Listing **45**

List of Tables

1.1 Tariff period rates 7

3.1 Time and observed cloud cover 21

4.1 Battery parameters used in Figure 4.1 26

5.1 Battery parameters used for simulations 27
5.2 Total energy for 2015/05/14 to 2015/06/05 inclusive 29
5.3 Battery full-cycles for 2015/05/14 to 2015/06/05 inclusive 32
5.4 Total costs for 2015/05/14 to 2015/06/05 inclusive 33
5.5 Yearly savings compared to household without generation (PS1 tariff) 33

A.1 Docker containers deployed on each host 39
A.2 Efergy Engage ingest 40
A.3 Allsolus Access ingest 40
A.4 UWA Solar ingest 40
A.5 Forecast.io ingest 41
A.6 RabbitMQ Routes 41

B.1 Appliance usage with power and energy consumption 42

List of Figures

1.1 Tariffs for energy time of use 8

3.1 Major system components and data flows 13
3.2 Time-series naming convention 16
3.3 End-user dashboard 17
3.4 Mobile device notification 18
3.5 Power consumption in a household over the period of a day 20
3.6 Daily minimum power consumption in a household 20
3.7 Solar power generation and estimated clear-sky radiation 21

4.1 Battery energy over a charge-discharge cycle 25

5.1 Daily energy consumption and generation 28
5.2 Single day power consumption and generation 28
5.3 Single hour power consumption and generation 29
5.4 Scatter plot of hourly energy consumption for household 1 and time of day 30
5.5 Scatter plot of hourly energy consumption for household 1 and temperature 30
5.6 Average hourly energy consumption for household 1 separated into weekdays and weekends 30
5.7 Comparison of power and energy over a day, with and without battery storage 31
5.8 Impact of battery capacity on average energy import per 15min . 31
5.9 Average daily costs with the addition of battery capacity for 2015/05/14 and 2015/06/05 inclusive 33
5.10 Average weekday and weekend energy and tariff costs for household 1 34

5.11 Average weekday and weekend energy and tariff costs for household 2 34

5.12 Average weekday and weekend energy and tariff costs for household 1 with 2kWh battery storage 34

5.13 Average weekday and weekend energy and tariff costs for household 2 with 2kWh battery storage 34

5.14 Average weekday and weekend energy and tariff costs for household 1 with 7kWh battery storage 34

5.15 Average weekday and weekend energy and tariff costs for household 2 with 7kWh battery storage 34

A.1 Example of schedule dictionary 39

A.2 Example of push notification POST request 41

B.1 Tumble dryer power consumption 42

B.2 Dishwasher power consumption 43

B.3 Washing machine power consumption 43

CHAPTER 1

Introduction

This exploratory study focuses the development of a system to monitor energy consumption and generation for households with roof-top solar photovoltaic (PV) systems and to use that data to drive shifts in energy consumption to better match generation. To maximise the value of energy generated by the solar PV system occupants must closely regulate their power usage throughout the day to match that of the solar PV systems generation. An ideal outcome would be a system that could predict and recommend the best times for occupants to use elastic-loads, such as charging electric vehicles (EV), running dishwashers, tumble-dryers or washing-machines.

In this paper we present the development of a system to ingest, process and visualise energy consumption and generation data. Using this system we implement a processor to predict, in the short-term, periods of lower than usual energy generation and generate notifications to suggest deferring using elastic-loads. A model of a battery energy storage system was created for use in simulation. Results from the data gathered by the monitoring system are used to show the potential of a low-capacity energy storage device and the reliability of energy predictions and disaggregation investigated.

To provide context a background overviewing the influences and challenges to solar PV energy generation, energy monitoring and energy prediction is given. After the aims of the project are outlined with a problem statement, a requirements analysis leads into the details of the system design and how it has been implemented. Followed by the design and simulation of a battery energy storage model. Finally an analysis of the data from the monitoring system is performed alongside the battery model created and the results generated conclude the paper.

1.1 Background

1.1.1 Overview

The latest figures from the Australian Bureau of Statistics [1] find that solar photovoltaic (PV) systems are now installed on the rooftops of one-fifth of Australian households, reaching a total of over 4GW installed solar capacity [2]. The majority of these systems are grid-tied, installed on domestic rooftops and have generation capacities ranging from 1.5 - 5kW [3]. These systems can optionally be installed as a hybrid with battery energy storage to help negate the intermittent power caused by changes in solar radiation, cloud conditions, temperature and v-i operating characteristics [4]. Grid-isolated systems require the battery capacity be sized to provide continual power during PV output fluctuations and throughout the night, however grid-tied systems can rely on the power grid for consistent power during these periods. Free of this constraint, grid-tied systems can balance the cost of battery capacity with sizing to maximise the value of energy generated by storing it to be used within the household rather than exporting it to the grid at a reduced feed-in tariff [5]. Changes in power consumption patterns within the household, even for a given total energy, can allow for more optimal utilisation of the power generated or the use of a smaller, more affordable system.

To meet the goals of a more sustainable future an increased proportion of energy from renewable sources combined with a reduction in consumption will be key. The most significant variances in domestic energy consumption have been attributed to occupant behaviour [6]. Typically occupants are unaware of their energy usage until the end of each billing period but by providing direct feedback in the form of a clearly understandable, real-time usage display energy savings in the range of 5-10% can be expected [7]. Further studies have expanded this principle by presenting the occupants with indirect feedback produced by processing raw usage data [8, 9], while others have tried incorporating social media [10] or provide tailored feedback to individual occupants [11]. The outcomes when using indirect feedback have been varied, depending largely on the quality of information presented to the occupants [7]. Disaggregated appliance usage feedback has, in the past, been expensive to implement and difficult to study, so limited data for outcomes exists, but theory suggests the increased insight for should prove valuable [7]. There are a large number of energy monitoring devices and platforms currently available, however as found in a study by Banerjee [12] the majority provide nothing further than access to fine-grained energy data, leaving the task to the user to interpret and monitor these results and decide what decisions to make to reduce energy consumption.

Since Hart [13] pioneered the approach of non-intrusive appliance load monitoring (NALM) to disaggregate individual appliance energy usage there have been many [14, 15, 16, 17, 18, 19] approaches to sensing and processing algorithms. The most widely adopted sensing being whole-house level power sensors and smart meter integrations [20]. Algorithms have been driven by the data features made visible with the sampling frequency and power-level resolution available [20]. While high frequency (>1MHz) and resolution sensors allow for algorithms to disaggregate a larger number of appliances [19], low frequency (1Hz) data from whole-house CT sensors can still produce useful disaggregation results into groups of appliance types [14].

1.1.2 Solar Photovoltaic Generation

The primary factor in solar photovoltaic system design is the ability to collect solar radiation. The amount of energy produced is a function of the solar radiation incident on the panel surface. This is dependant on a number of factors, Duffie and Beckman [21] found the main factors as being global radiation, ground reflectance, time and day of year, panel tilt angle, altitude and climate influences. Energy generation from a solar PV system is also dependant on other characteristics of the system and site. Iyengar et al. [22] found these factors to be the characteristics of;

- **panel** size, type, age and number,
- **site** placement, tilt and orientation,
- **surrounding area** proximity to trees or buildings that could cast shadows over the panels,
- **season**, determining the length of day and solar intensity,
- **weather** factors of temperature and cloud cover.

Due to these factor, close-by objects casting shadows can make solar generation prediction difficult, as there are dynamic changes in shadows throughout the day and year. Iyengar et al. [22] found that most prediction models had focused on only large solar farms, where these factors cause less of an impact given the typical, more open siting. A method for calculating cloudless irradiance by multiplying all weather parameters proved to be useful in making predictions in a study by Sharma et al. [23].

1.1.3 Energy Display & Feedback

Many studies [24, 25] have found that energy consumption is essentially invisible to household occupants and thus difficult to connect to specific behaviours. A qualitative field study by Hargreaves, Nye and Burgess[26] of households given visual energy displays, found that the monitors needed to look visually attractive and provide clear information that is easily contextualised. Typical in-home displays give a simple factual feedback of the current power usage, and historical comparative feedback using graphs comparing the energy consumed in periods of time [9]. After evaluating the current state of energy consumption visualisation, Masoodian et al. [27] found most to use either a time-series or pie chart, they introduce a new visualisation, the *time-pie*, combining elements of both and allowing for the inclusion of contextual information. Moving further away from simplistic, raw data presentation, Holmes [28] propose *eco-visualisations* as a method of combining data-driven computer animations and artistic forms to create ambient energy visualisations and potentially act as a non-monetary incentive to increase conservation.

The impact of feedback on behaviour has been extensively studied, a review by Abrahamse et al. [29] found that feedback is more effective when provided in real-time and frequently. A study by Darby [7] into the effectiveness of feedback on reducing energy consumption found that it is predominately unknown by many users and that immediate, real-time feedback could be extremely valuable if presented in a clearly understandable manner. Fischer [30] identifies additional features for successful feedback as involving appliance-specific breakdowns and interaction or choices. Implementations of useful feedback have failed to keep up with current knowledge, often due to the technical requirements of the required monitoring system [30].

1.1.4 Predicting Energy Consumption

Most past research has focussed on predicting energy consumption from the perspective of load-prediction for use by electricity generators at utility scale. When applied to only a single household energy consumption prediction allows for the intelligent use of energy storage systems and load-shifting. Hobbs [31] groups energy prediction into three range categories, short, medium and long-term.

- **Short-term** predictions being between one-hour to one-week,
- **medium-term** between one-week to one-year,
- **long-term** as more than a year in advance.

Mishra et al. [32] studied short-term prediction in several households over a three month period. Using various sensors they were able to predict energy consumption for five periods of a day, in some cases reaching with an error of less than 5%. Using the available REDD dataset for analysis Viet et al. [33] studied the relationship between historical and future energy consumption. Errors for their predictions ranged between 5 and 150%.

1.1.5 Battery Energy Storage Systems

Early studies [34, 35] began with a focus on system design for grid-tied solar PV energy smoothing to meet utility connection requirements rather than to store energy for an isolated system using available components. At the time of the Hund, Gonzalez and Barret [34] study, valve-regulated lead-acid (VRLA) batteries were the most economical choice but was noted that an their smoothing algorithm would be damaging to battery longevity. The report notes that the system was not suited to providing smoothing for short periods of cloud cover but rather smoothing a solar irradiance curve modelled from the previous day and spreading the peak of the power generation curve. The study determined that smoothing could be achieved using a relatively small battery capacity, but had only modelled their irradiance averaging algorithm. Zahedi [35] reviewed the current role of energy storage in grid-tied solar PV systems and identified the benefits to the power grid of a wider deployment. The paper modeled a super capacitor as the ideal storage solution. Li, Hui and Lai [36] further improved over the smoothing system from Hund, Gonzalez and Barret [34] and produced a model for a real-time smoothing system able to smooth fluctuations of 10% per 15 minutes and had shifted from VRLA batteries to better suited $LiFePO_4$ batteries. Chersin, Ongsakul and Mitra [4] approached the design of a PV smoothing battery energy storage system (BESS) with a model that better suited a time-of-day tariff and optimised local consumption. This study modelled a BESS able to provide a constant power output during daytime and smoothing battery charging from the grid during off-peak hours. The constant power output by the system is better matched to the stand-by consumption of a household, so is consumed locally rather than exported to the grid at a lesser value.

1.1.6 Energy Disaggregation

Nonintrusive Appliance Load Monitoring (NALM) is built upon the research by Hart [13] into determining the consumption of individual appliances by analysing the total current and voltage, this was in contrast the the techniques previously used requiring sensors to be placed at each appliance to measure individual power consumption. This method proved to be very convenient and effective compared to using individual sensors. Two algorithms were presented in the paper MS-NALM and AS-NALM, manual and automated initial setup variants, which are combined with appliance models and signatures to determine which appliance is undergoing a state change. At this point NALM was not reliable with appliances using <150W or those using a continuously variable amount of power. This paper lays out the theory for NALM, which later algorithms and models expand on.

Since its introduction, research into NALM has covered a range of sampling frequencies, from hourly [37] to into the mega-hertz range [38]. Most studies have involved either low frequency power measurements or kilo-hertz range AC current measurements. The use of high frequency (kHz) data samples allowed Laughman et al. [39] to identify appliance signatures using higher harmonics and transient characteristics of the power signal and Patel[40] to observe identifiable differences in the noise created by the switch of an appliance. Systems requiring a high sampling frequency are held back from broader deployment due to the costs. Using a lower frequency of sampling (1Hz) for pattern matching was studied by Berges et al.[41] with the constraint of disaggregating to only a small set of typical appliances. Baranski and Voss [42] use statistical methods to build appliance pattern databases, which Bergman et al.[43] later use to apply dynamic programming to increase the accuracy of the database by allowing for variations between different appliances of the same type, however results are only accurate with kilo-watt range loads.

More recent research by Kolter and Johnson [44] applied Hidden Markov Models (HMM) in appliance modelling, Kim et al. [15] extended the probability appliance model factoring in typical usage durations, time of day and past correlations between appliance usages. While Vogiatzis, Kalogridis and Denic [45] have shown success in applying rule-based filtering and Fourier analysis to optimise the the finite state machine appliance model when using power measurements sampled only every 20 seconds.

One more novel approach has been to extend the disaggregation from the appliance level into the energy used by a particular occupant for that appliance. In Lee et al. [11], a system to personalise appliance disaggregation is proposed based on tracking the households occupants, at the room level, and linking this with the given appliance room location.

The application of NALM to real-time feed is limited as most algorithms are intended for off-line processing and are computationally expensive [46], [15], [44]. These studies have taken the approach of modelling appliances into transitions between a small number of steady operating states. To achieve reasonable levels of accuracy only the energy consumption of a small number of devices, with only on/off states, could be disaggregated, typically appliances using significant energy for heating or cooling and simple thermostat control. A review of NALM approaches by Barker et al. [47] found state-based appliance model approaches are not suitable for online processing to provide real-time results.

1.1.7 Tariffs

Energy retailers apply a charge per unit of energy imported-to or exported-from the grid. A number of tariffs are currently available, in the case of Synergy [48], there are three options of residential import tariff and one export tariff. These are

- **Home Plan** (A1), a flat-rate unit charge
- **SmartPower** (SM1) a variable-rate unit charge with weekday morning and afternoon peak rates with seasonal changes to time periods, and
- **PowerShift** (PS1) a variable-rate unit charge with weekday afternoon peak rates, in addition to
- **Renewable Energy** (RE) a fixed-rate unit charge for energy exported.

To encourage roof-top solar PV uptake previous government feed-in subsidies resulted in exporting energy to the grid being cost-effective than self-consumption. These subsidies are no longer available for new installations in Western Australia and as they effectively negate the economic argument for self-consumption and use of battery storage they are not discussed further in this study. The costs of the tariffs studied are given in Table 1.1, and their hourly variance over a winter week is shown in Figure 1.1.

Cents per unit	**A1**	**SM1**	**PS1**	**RE**
Constant	24.5961			7.1350
Peak		47.4099	22.6326	
Off peak		12.7360	11.7193	
Weekend shoulder		19.9873		
Weekday shoulder		24.1195		
Super peak			41.1872	

Table 1.1: Tariff period rates

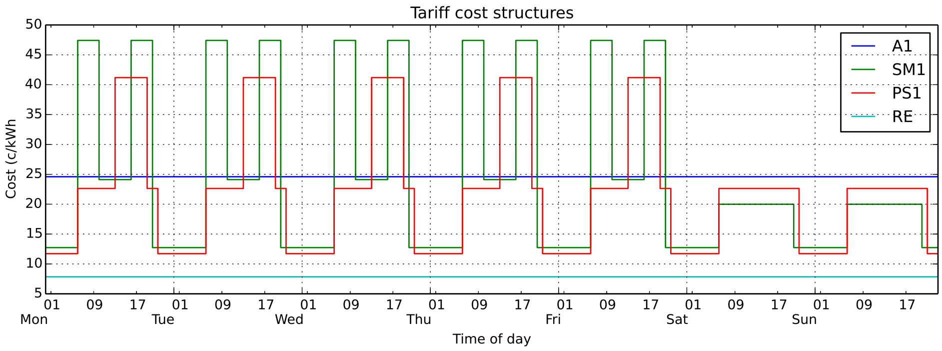

Figure 1.1: Tariffs for energy time of use

1.2 Monitoring Locations

A number of locations, with differing generation and monitoring equipment were used in this study. The system was designed with a consideration of ingesting data from a variety of sources, those used are detailed in this section.

1.2.1 Off-grid Solar PV with storage

The UWA Future Farm 2050, located 150km south-east of Perth, consists of 2 x 5kW ground mounted solar arrays with battery storage to provide up to 3 days redundancy. The initial system was installed in mid-2012, however a manually switchable grid connection was later added. An online platform, AllSolus Access, is integrated with the AllSolus LiveBase for data collection.

- 22 x 235W Q-Cells Q-Pro G2 solar cells
- SMA Sunny Boy 5000TL solar inverter
- 56 x 90W Q-Cells Q-Smart modules solar cells
- SMA SMC 5000A solar inverter
- 24 x Hoppecke 8 OpzV Solar.Power 1000 batteries
- 3 x SMA Sunny Island 5048 battery inverter
- AllSolus LiveBase data-logger

1.2.2 Grid-tied Solar PV

Two home locations are monitored each with separate Efergy CT-clamp transmitters installed on the household-side of the main breaker and on the solar inverter feed-in connection. The Efergy transmitters take a measurement at most every 6s and transmit this wirelessly to an in-home internet-connected hub, which in turn uploads this measurement to the Efergy Engage platform. Data is then accessible via the Efergy Engage platform, with no direct access via the hub possible.

- **Household 1** Single-phase 2.5kW solar
- **Household 2** Three-phase 5kW solar
- **UWA Human Movement** Three-phase 10kW solar

The UWA Human Movement Building has a solar panel installation with a total capacity of 20kW. The installation contains a SMC Sunny WebBox data logger that is accessible within the UWA network. An existing system periodically downloads data from the SMC Sunny WebBox and stores this data in an Internet accessible location.

- 2 x SMA Sunny Boy 10000TL solar inverters
- SMA Sunny WebBox

CHAPTER 2

Aims

2.1 Problem Statement and Motivation

Previous studies have found that feedback to occupants on energy usage can lead to reductions in the order of 5-10% [7, 8, 9]. The degree of change in occupant behaviour to reduce consumption has been linked to the quality and clarity of feedback provided. A significant number of households currently incorporate solar PV systems to generate power [1] and future increases in the adoption of battery energy storage systems (BESS) [4] and electric vehicles (EV) are likely. These conditions combined mean household occupants must track not only energy consumption, but also generation, diversion to storage, and their temporal relationship, to minimise energy consumed from the power grid and optimise economic benefits given the feed-in tariffs available.

A system capable of monitoring and processing data for household energy consumption, generation and storage in real-time could provide useful feedback to occupants leading to a reduction in energy consumed from the power grid. This project seeks to develop a prototype system that will process energy data from typically available monitoring sources to provide immediate and useful feedback to occupants.

CHAPTER 3

System Design

This chapter states the requirements of the system then details the design of components and platform they are deployed upon. Data flows between these components are shown and a rationale for the selection of tools used given.

3.1 Requirements Analysis

System development, particularly in the software space, begins by undertaking a requirements analysis. Functional requirements specify the required behaviour and processes of the system, while non-functional requirements drive the architecture. Consideration is given to key functional and non-functional requirements for the system developed in this project.

3.1.1 Functional

- **Ingest of energy monitoring data** is the primary requirement of the system, as without this input no processing or output can be produced. This requirement includes ingesting consumption and generation from a number of sources, be that directly from monitoring devices, via gateways or hosted services. This process should be as timely as possible, within the constraints of the source, and not sacrifice measurement resolution. Data should be ingested with the sampling rate faithful to the source, including support for a variable period between samples. The system should have the capacity to import historical data.

- **Ingest of environmental data** to support in data processing and prediction. Historical, present, and forecast environmental data must ingested by the system to be made available for analysis.

- **Historical data** of any ingest to the system is to be stored and accessible.

- **Data processing** to both validate data being ingested and be able to condense the many data streams into useful information. A further requirement of this aspect is for the system to be able to extract key information from the current and historical data, and perform predictions using this data.
- **Feedback** should be provided to the user in the form of simplified energy statistics, visualisations and recommendation notifications. This includes includes historical, present and predictions.

3.1.2 Non-functional

- **Accessibility** is a key non-functional requirement, in that the system be not limited any any one platform for access.
- **Response time** is kept to a minimum when accessing large ranges of data.
- **Reliability** of the correctness in processing, visualisations and usefulness of notifications must be ensured. Notifications are only to be generated on predictions of high confidence.
- **Reproducibility** of the system to allow it to be easily re-built, installed and configured by other researchers.
- **Modularisation** of the system to allow changes in one component without restarting the entire system.

3.2 System Components

An outline of the major components that comprise the system is shown in Figure 3.1. Ingest worker nodes accept tasks, from a messaging queue, to collect data from external monitoring sources. These tasks are periodically scheduled to the a messaging queue by the task controller. Ingest workers place the collected data back onto a different messaging queue, where it is then immediately available for any processing nodes and also stored in a time-series database. Processing nodes act upon the incoming data streams and are also able to extract data from the time-series database. Results from each processing node are placed back onto a messaging queue, for immediate access by other processing nodes, and to be stored in the time-series database. Processing nodes are able to send notifications via an external notification service. Data from the time-series database can be visualised using graphs and metrics on a dashboard, where aggregations and down-sampling are performed using query functions of the time-series database.

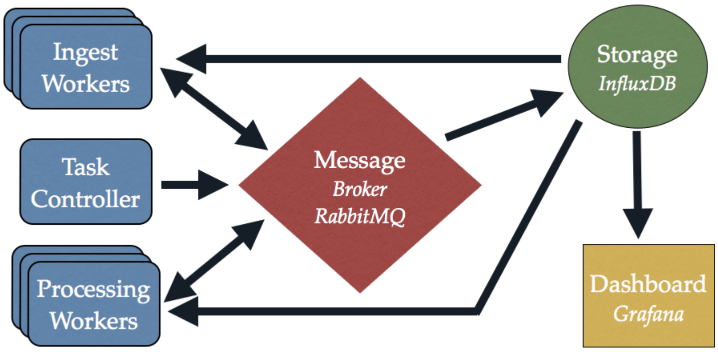

Figure 3.1: Major system components and data flows

3.2.1 Platform

The various components of the system have been compartmentalised using Docker [49] containers and deployed across two DigitalOcean [50] hosts. The allocation of containers to hosts is given in Table A.1. Docker is an open platform that builds upon the use of Linux (LXC) containers. Using Docker is similar to deploying each application in its own virtual machine, the key difference being that each Docker image shares the same Linux kernel already running on the host. This allows for many of the benefits of running and storing applications as self-contained virtual machines but without the same level of overhead. Each Docker container is created using a simple script, known as a *Dockerfile*, that defines exactly how to build up the image, right from the operating system, to any libraries and dependancies, and finally application. A review by Boettiger [51] recognised the applicability of using Docker to build a system that is easily reproducible for use by future researchers.

3.2.2 Task Controller

To generate the tasks that need to either periodically or continually execute, to facilitate data ingest or processing, a *Task Controller* has been implemented. Dependant on the particular task the controller is able to assign a continually executing task or periodic tasks at a fixed rate or on a date/time based schedule. The *Task Controller* has been implemented using a Celery [52] beat scheduler. Celery is a Python [53] framework for a distributed task queue. While the primary use case of Celery is for real-time processing it is also able to support scheduled tasks. The scheduled tasks are configured on the *Task Controller* as Python dictionary objects, see Figure A.1, and loaded into the Celery beat scheduler at start-up.

3.2.3 Ingest and Processing Nodes

As both ingest, and processing nodes perform similar functions they have both been implemented using Celery workers. Each node is a Docker container executing the Celery daemon, *celeryd*, which presents that node as a number of concurrent Celery workers assigned to a particular task queue. The task queues are assigned using a routing pattern in the form *tasks.processing.** and *tasks.ingest.**. As tasks are pushed to queues matching these routing patterns, workers accept and execute these tasks. Data ingested or results from processing are then pushed by the worker back to a messaging queue with a routing key set to channel them into storage, as described in Section 3.2.5.

Efergy Engage

Data from the Efergy Engage platform is accessed via a web service as described in Table A.2. As the Efergy Hub only transmits data every 6s, this is the period we use to poll the API. The 6-second resolution data is only available when polled live, historical data is only stored by Efergy in 1-minute intervals. For every household monitored using the Efergy Engage platform a different API key is used, therefore individual polling instances for each household are required. When the connection between the in-home Efergy hub and Engage platform is unavailable data is not buffered by the hub. Due to this we can be certain that data we read from the live API will not later be back-filled with additional data.

Allsolus Access

Data from the Allsolus Access platform is accessed via a web service as described in Table A.3. The Allsolus platform uses an on-site logger, the *LiveBase*, to buffer data to send periodically to the Allsolus Access platform. Measurements on the *LiveBase* logger are typically made every few minutes, but at most frequent are uploaded to the Allsolus Access platform in 10-minute intervals. If the data connection between the LiveBase and Allsolus Access platform is unavailable the LiveBase will buffer data to transmit latter. For this reason we cannot simply poll the Allsolus Access platform for live readings but must first check our database for the last reading and query the Allsolus Access platform for any readings between then and the present time. Only a single measurement channel can be queried in each call, so individual polling instances for each device measurement are required.

UWA Solar

Data from the UWA solar logging platform is accessed via a web service as described in Table A.4. The UWA solar logging platform downloads CSV files, generated by a Sunny Webbox data logger, every 15 minutes and stores these in a location accessible via HTTP. The Sunny Webbox data logger stores measurements at 5-minute intervals and buffers data internally, splitting the CSV file by date. So, as with the Allsolus Access ingest, we first check our database at each poll to determine what period we need to search through the CSV files to ingest. Each row in this CSV has all the measurements made, so only a single polling instance is required.

Forecast.io

Weather data is ingested from the Forecast.io [54] service using a web service as described in Table A.5. The weather data provided by Forecast.io is generated from data aggregated from a wide range of sources and provides both current conditions and hourly forecasts for a given location. The current conditions are polled and ingested directly by a polling instance for each location. Additional polling instances for each location also ingest the current and next day's hourly forecasts every hour. Unlike the rest of the data ingested the forecast data is presented with a routing key that does not result in it being stored in the database.

3.2.4 Message Broker

The open-source message broker application RabbitMQ [55] provides an implementation of the Advanced Message Queuing Protocol (AMQP). A Docker container for RabbitMQ has been created to host the application as part of the system. The Python library Pika [56] is used by the components of this system to interact with RabbitMQ using AMQP. When messages are passed to RabbitMQ they contain a routing key, which determines how they will pass through and to what components they will be made available. The configuration of these routes is summarised in Table A.6.

3.2.5 Storage

InfluxDB [57] an open-source time-series database is used to provide storage for the system. When compared with traditional relational databases, a time-series specific database is optimised to work with time-series data and can provide a significant speed advantage. InfluxDB also provides functions for easily querying to aggregate or down-sample data. Any queries that are commonly used can be converted into continuous queries, so that data is processed as it enters the database rather than when being queried.

RabbitMQ is connected to InfluxDB using the InfluxDB Storage Exchange plugin [58]. This plugin allows an InfluxDB to appear as a RabbitMQ routing exchange. Any messages routed to this exchange in RabbitMQ are then stored into an InfluxDB time-series named with the routing key. The routing keys, and thus time-series names in InfluxDB follow the format given in Figure 3.2, a full listing of the time-series is given in Appendix C.

```
[site].[device].*.*
[site].notification
[lat]_[lng].weather.*
```

Figure 3.2: Time-series naming convention

Each entry in a time-series contains a UTC timestamp, auto-incrementing sequence number and typically only a single value. Multiple values can later be combined from other time-series using either merge or join operations.

3.2.6 Dashboard

The primary end-user interface to the system is a graphical dashboard view. This dashboard is implemented using the Grafana [59] web application. Grafana is open-source dashboard web application built upon the JavaScript framework AngularJS [60] and and plotting library Flot [61]. Being open-source and built with common technologies makes Grafana extensible to adding new graph types or metrics.

Grafana integrates directly with InfluxDB to query any time-series and provides an internal user account database to apply permissions over the data and dashboard options presented. As the user adjusts the time-span being viewed in the dashboard Grafana appends variable down-sampling parameters to the InfluxDB query to limit the data required to best suit the level of zoom.

A dashboard view, as seen in Figure 3.3 has been developed to show the user how their power consumption and generation align over a period of time. The view also shows instantaneous power and trends over time for the amount of energy being generated and consumed. We also show metrics of always-on power and variable energy consumption and generation over a period including the amount of this imported-from or exported-to the grid.

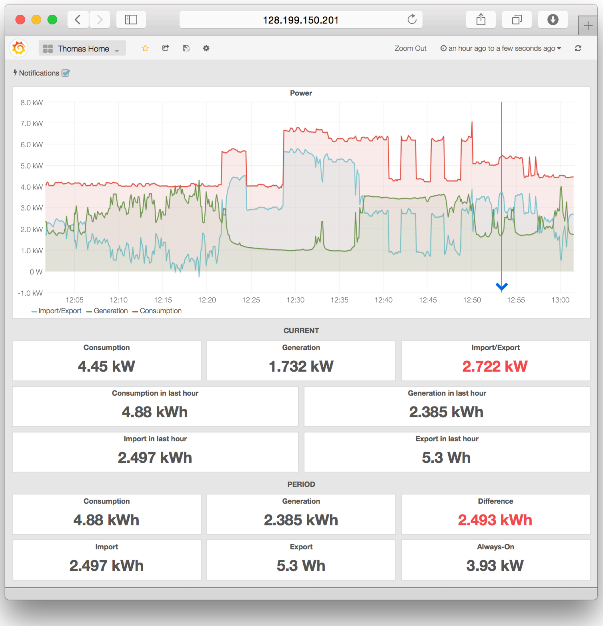

Figure 3.3: End-user dashboard

3.2.7 Notifications

A simple cross-platform method of sending notifications to users was required. To achieve this, the delivery of notifications to end-devices is handled by the service Pushover [62]. Pushover is a service that takes in commands to send push notifications via a HTTP POST request, as given in Figure A.2, and delivers native push notifications to devices using their client on iOS, Android or desktop as shown in Figure 3.4.

Figure 3.4: Mobile device notification

A notification can be generated by any processing node. When this occurs the command to generate a push notification is immediately sent to Pushover by the worker that generated the notification and the notification put into the messaging queue with a notifications routing key. This allows for any notification to be received for any further processing and to be stored to be used for annotations on the dashboard.

3.3 Metrics Calculations

Energy monitoring devices used in this study provide a periodic measurement of instantaneous power, with the period between measurements not guaranteed to be constant. This section discusses how we calculate a measure of energy and other metrics from these readings.

3.3.1 Normalising Periodic Measurements

As measurements ingested into the system may be taken at varying intervals, the period between measurements must be considered when performing calculations. Given a fixed duration, simply calculating the arithmetic mean, Equation 3.1, for that period results in a value weighted to intervals with more frequent measurements.

$$\bar{p} = \frac{p_1 + p_2 + \cdots + p_n}{n} \tag{3.1}$$

To address this we must calculate a time-interval weighted mean, using Equation 3.2 with weights given by the time-span between measurements. If all weights are equal this simplifies to the arithmetic mean.

$$\bar{p} = \frac{t_1 p_1 + t_2 p_2 + \cdots + t_n p_n}{t_1 + t_2 + \cdots + t_n} \tag{3.2}$$

This is implemented in our system by first up-sampling measurements using a continuous query in InfluxDB, resulting in measurements being converted into fixed intervals in the up-sampled time-series by forward-filling from the last value. This up-sampled time-series is then used to perform calculations.

3.3.2 Energy From Power

We can calculate energy from the power measurements and the time-span between them using Equation 3.3. Energy is calculated for both consumption and generation, as well as import and export from the grid. Energy consumption and generation are directly calculated from the power measurements. For grid imported and exported energy, first the difference between power consumption and generation is derived at each point in the time-series and used for this calculation. Periods with a positive power difference occur when energy is being imported from the grid and negative when it is being exported.

$$E = \sum_{i=1}^{n} t_i p_i \tag{3.3}$$

This can further be simplified, given the normalised time periods when using the up-sampled time-series, we can calculate energy in InfluxDB by summing the power measurements over a time range and scaling this value by the number of hours in the given time range.

$$E = t_{range} \sum_{i=1}^{n} p_i \tag{3.4}$$

3.3.3 Always-on Consumption

The always-on power consumption is a measure of the constant power consumption of a household when occupants are not actively using any appliances. This figure represents the stand-by power used in a household. Power consumption throughout a typical day is shown in Figure 3.5. The always-on power consumption is most clearly noticeable in the early hours of the day, 01:00AM - 05:00AM. To approximate always-on power consumption we take the minimum of the power measurements for a period, the result of this is shown in Figure 3.6.

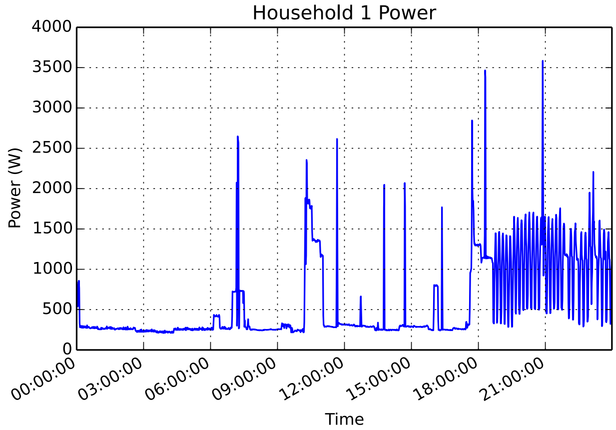

Figure 3.5: Power consumption in a household over the period of a day

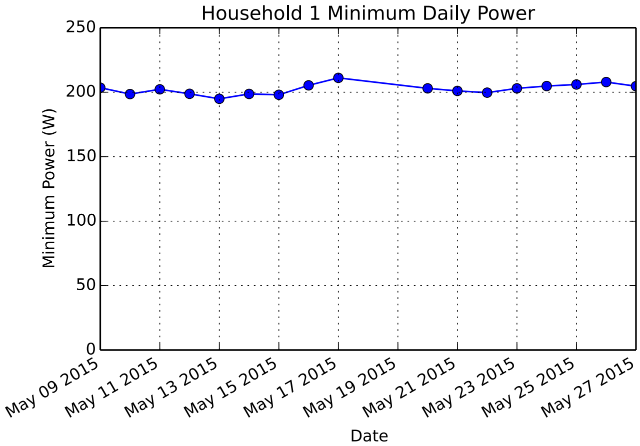

Figure 3.6: Daily minimum power consumption in a household

3.3.4 Clouded Irradiance

The PySolar [63] package was used to calculate an estimated value for clear-sky radiation at each location for a flat surface. In Figure 3.7 the estimated clear-sky radiation is shown against the power generated. When looking at the hourly observations for the day using Forecast.io, given in Table 3.1 we can visually identify a relationship between cloud cover and power generation. A simple metric for this is calculated as the product of estimated clear-sky radiation and clear-sky cover. The clear-sky cover being the remaining percentage of sky after the cloud cover has been subtracted. This metric is then grouped into a mean for morning and afternoon to be used in generating notifications.

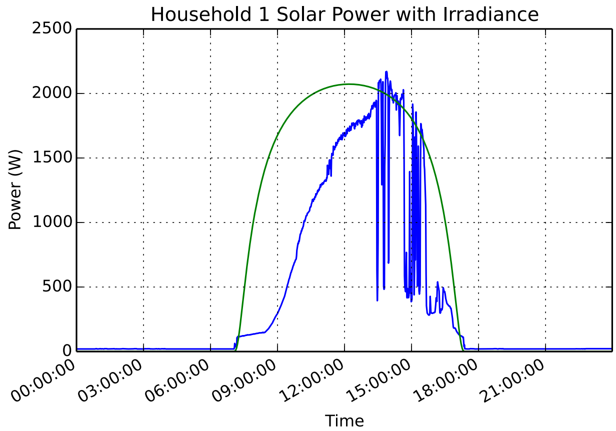

Figure 3.7: Solar power generation and estimated clear-sky radiation

Time	Cloud cover (%)
06:00:00	0
09:00:00	0
11:00:00	0
12:00:00	9
15:00:00	66
18:00:00	79
19:00:00	0

Table 3.1: Time and observed cloud cover

CHAPTER 4

Battery Model

A battery model was required to investigate the impact of incorporating battery energy storage with household roof-top solar PV systems. This chapter details the development of such a model and how it was used in a simulation with data gathered by the energy monitoring system.

4.1 Design

As periodic consumption and generation power were the only measurements available from the monitoring system a simplified battery model driven by only these measurements was created for use in a discrete-step simulation. The assumptions made in the model closer reflect a battery energy storage system rather than battery cell, in that the voltage-current (V-I) relationship is abstracted by a battery management system (BMS).

The battery model was implemented in Python [53] as a discrete time-step power-energy-power converter incorporating an ideal fixed-capacity energy store. An outline of the algorithm is shown in Algorithm 4.1 and the complete code listing in Appendix D. Once instantiated, the battery is at its lowest depth-of-discharge (DoD), and can be either charged or discharged at a given power level and duration. Returned is either the amount of energy consumed-by or extracted-from the battery for this duration. A maximum continuous power and short-duration peak power limit the energy returned, as does the current amount of energy stored in the battery, or state-of-charge (SoC). Conversion losses are applied as a fraction of this energy at both charge and discharge. A cycle is counted by the battery after the SoC reaches full and then returns to empty.

4.1.1 Assumptions

- Power is constant (either continuous or peak level) given there is available energy capacity.
- Peak charging/discharging power is possible at a fixed multiple of continuous power for a set period. Unless this period is fully depleted a set recovery period is not required.
- Conversion losses are equal for charge and discharge.
- Capacity is unaffected by charge/discharge rate (No Peukert effect) or cycle count.
- A cycle is counted only when SoC meets the extremes of empty-full-empty for the given depth of discharge (DoD) and maximum charge level.
- No temperature dependance.
- Self-discharge is not significant.
- Instant charge/discharge response time.

4.1.2 Parameters

- **Capacity**: Maximum battery capacity (Wh)
- **Depth of discharge**: Minimum level to discharge to (ratio compared to full capacity)
- **Maximum charge**: Maximum level to charge to (ratio compared to full capacity)
- **Discharge rate**: Maximum sustained discharge rate (W)
- **Charge rate**: Maximum sustained charge rate (W)
- **Park charge/discharge rate**: Peak rate (ratio compared to charge/discharge rate)
- **Peak rate period**: Maximum continuous peak period (s)
- **Peak rate recovery period**: Duration to recover from peak rate (ratio compared to peak rate period)

- **Conversion loss**: Loss in energy during conversion to/from storage (ratio lost compared to input amount)

Algorithm 4.1: Battery Model

```
begin
  for energy measurement step period do
    if power > 0 then
        charge(power, period);
      else
        discharge(power, period)
    fi
  where
  proc charge(power, period) ≡
    if peak charge left then
        power := min(power, peak rate)
      else
        power := min(power, continuous rate)
    fi
    energy stored += (power * period) * (1 - losses)
  .
  proc discharge(power, period) ≡
    if peak discharge left then
        power := min(power, peak rate)
      else
        power := min(power, continuous rate)
    fi
    energy stored -= (power * period) * (1 + losses)
  .
end
```

4.2 Simulation

An environment that steped through a time-series of power levels was required to operate the battery model in a simulation. This was achieved using a Pandas dataframe with values for power consumption and generation along a fixed-frequency time-series index, which was then iterated along with the state calculated at each step. To demonstrate the battery model a step function, in three-hour intervals, attempting to charge at 2kW, zero-power, and attempting to draw 2kW is show in Figure 4.1 using the parameters given in Table 4.2.

Data captured by the monitoring system, for both households with roof-top solar PV, between 2015/05/14 and 2015/06/05 inclusive was then used to provide the power level values into the simulation. For these simulations the environment included the grid-connection as a source/sink for infinite energy. Using the fixed interval data as generated in Section 3.3.1 the environment was calculated at each step and the step results saved. This calculation first determines the flow of power at the step, by comparing power consumption and generation to determine whether there is any shortfall or excess and how much energy has been generated and consumed by assuming the power levels are constant for the step duration. If there is an excess of power at this step an attempt is made to use it to charge the battery for the step duration. The battery stores energy from this charge period according to its parameters and the remainder is exported to the grid. If instead there is a shortfall in power, an attempt is made to discharge the battery at this level for the step duration in preference to importing from the grid. This process was repeated for each household and set of battery parameters, as given in Table 5, including a zero-kWh ideal battery to show the comparison to the present situation.

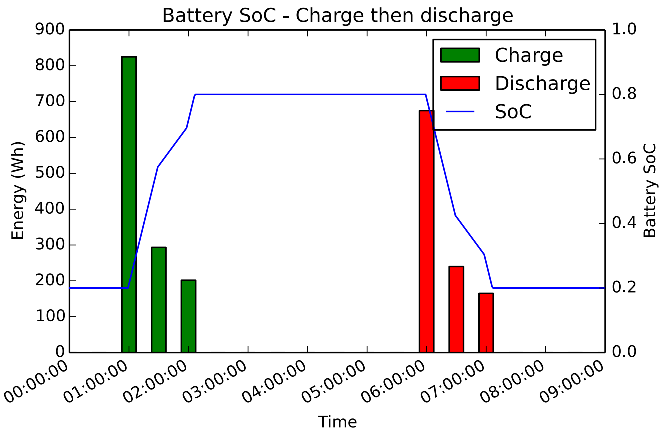

Figure 4.1: Battery energy over a charge-discharge cycle

Capacity	2 kWh
Range	20% - 80%
Continuous Power	500W (0.25C)
Peak Power	1500W (0.75C)
Losses	10%

Table 4.1: Battery parameters used in Figure 4.1

CHAPTER 5

Results & Analysis

Using data ingested and processed by the system some initial data analysis has been performed. In particular these analyses shows the usefulness of the ingest, processing and storage components of the system. The dashboard user interface was used for an initial visual analysis of the data, while the Python library Pandas [64] was used for a deeper analysis. Unless otherwise stated results have been calculated for the period of 2015/05/14 to 2015/06/05 inclusive and for the battery parameters as shown in Table 5.

Capacity (kWh)	0.5, 1, 2, 5, 7, 15
Range	0% - 100%
Continuous Power	0.5C / 0.5E
Peak Power	0.75C / 0.75E
Losses	10%

Table 5.1: Battery parameters used for simulations

5.1 Energy Usage

For the period examined in this analysis daily totals for energy generation and consumption are shown in Figure 5.1 for both households. Over this period it is evident that no amount of battery storage could completely remove the need for grid import as daily generation rarely exceeds consumption. Low energy consumption can be seen in household 1 for May 18th and 19th, this was due to a power monitoring transmitter requiring a battery replacement. The time period for these two days has been excluded from any further calculations.

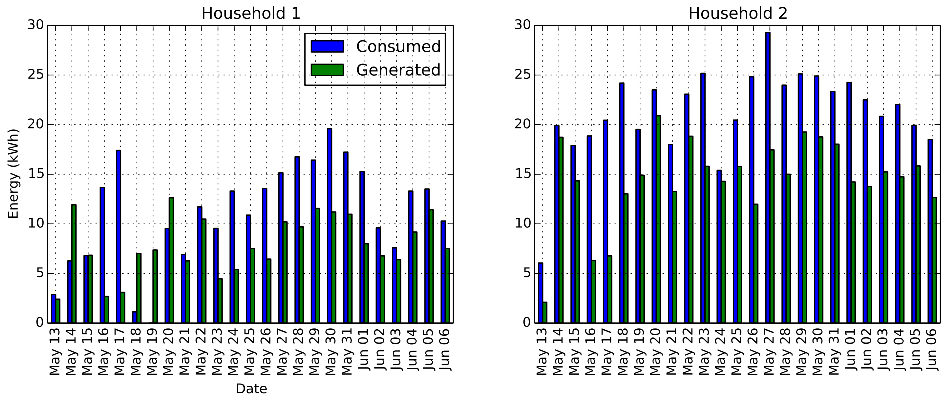

Figure 5.1: Daily energy consumption and generation

Looking at energy generation and consumption at a daily level obscures the true amount of grid interaction as varying proportions of that energy generated and consumed are exported and imported from the grid. In Figure 5.2 a view of the power consumption and generation for both households over a single day, May 26th, shows the variations in alignment and amplitude. These variations can be seen even closer in Figure 5.3 and correspond to a need to either import or export energy to the grid. Short duration, often just minutes, changes in power in the order of 1-2kW can be seen.

The total energy figures for both households have been calculated and are given in Table 5.1. Both households have shown the ability to generate approximately 70% of the energy they consumed, but only 25% and 58%, respectively, of that consumed was from local generation.

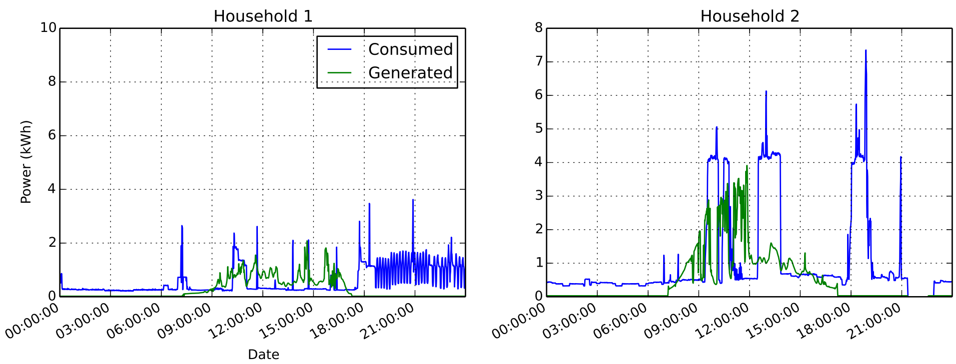

Figure 5.2: Single day power consumption and generation

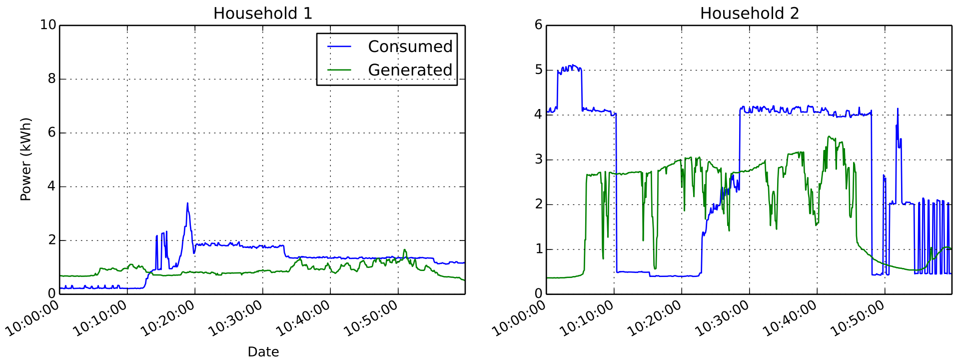

Figure 5.3: Single hour power consumption and generation

Energy (kWh)	**Household 1**	**Household 2**
Energy Exported	122.61	130.68
Energy Imported	200.25	290.91
Energy Generated	187.50	347.14
Energy Consumed	265.13	507.37

Table 5.2: Total energy for 2015/05/14 to 2015/06/05 inclusive

5.2 Energy Prediction

To determine the correlation between energy consumption and time or temperature, scatter plots were created as shown in Figure 5.4 and Figure 5.5. The baseline energy consumption is clear and consistent across both, with the time of day plot showing a trend toward increased power consumption in the evening and night. Given this trend, the temperature plot would need to be controlled for time of day before any further conclusions could be drawn from it.

Focussing on the correlation between energy consumption and time of day, Figure 5.6 shows how average hourly energy consumption compares between weekdays and weekends. It is clearly evident that during the weekday working hours significantly less energy is consumed. Given the limited period of data ingested into the system, and that this data covers only a winter period it is fair to assume this difference is due to heating while the household occupied rather than an opportunity for appliance load-shifting.

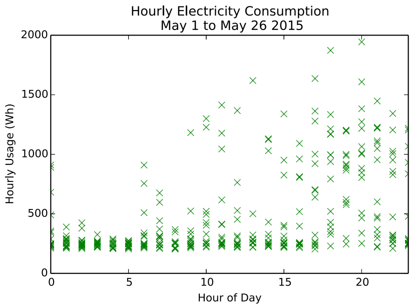

Figure 5.4: Scatter plot of hourly energy consumption for household 1 and time of day

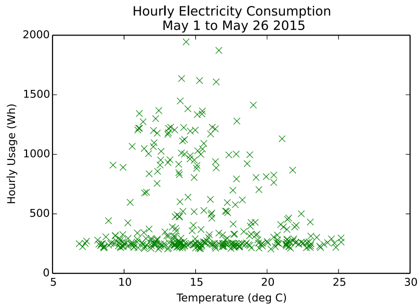

Figure 5.5: Scatter plot of hourly energy consumption for household 1 and temperature

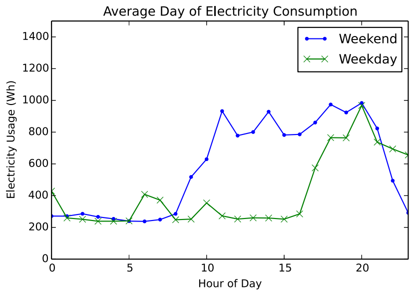

Figure 5.6: Average hourly energy consumption for household 1 separated into weekdays and weekends

5.3 Battery Storage

The impact of the addition of even a small battery (2kWh) can be seen in Figure 5.7, energy is diverted to battery storage then returned throughout the day thus reducing the amount imported from the grid. It can be see how the battery charge increases smoothly and provides power for a number of hours after sunset. The reduction in the average hourly energy imported for increasing battery capacity can be seen in Figure 5.8. This reduction appears most significant during the afternoon and evening, which corresponds with an increased level of consumption, usually of grid-imported power, that instead the battery storage is able to provide.

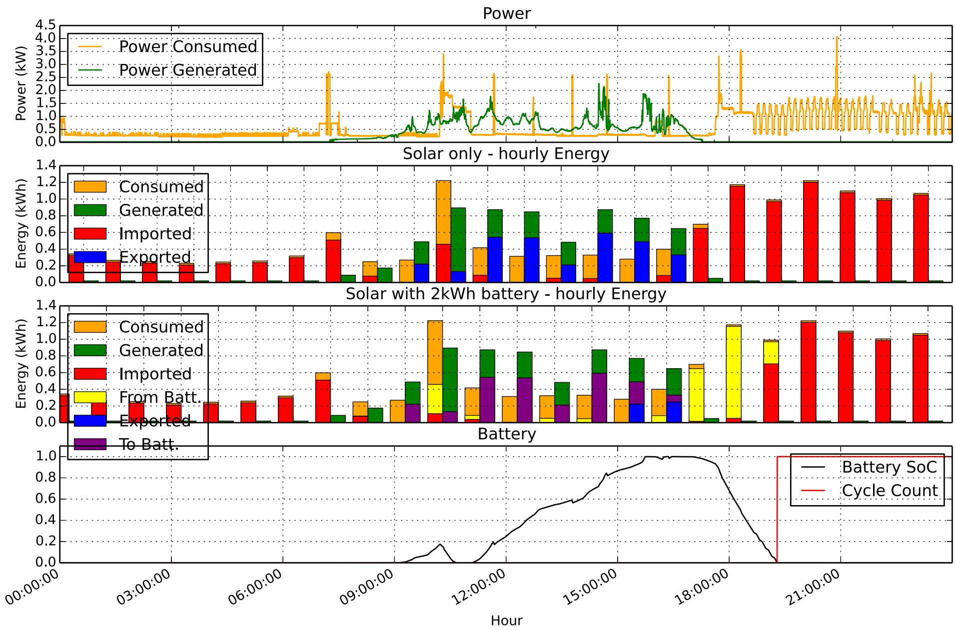

Figure 5.7: Comparison of power and energy over a day, with and without battery storage

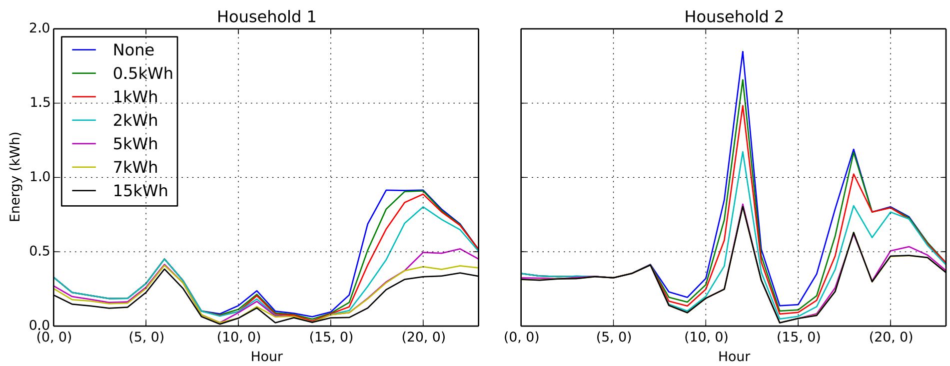

Figure 5.8: Impact of battery capacity on average energy import per 15min

The full disharge-charge-discharge cycle count for the range of battery sizes is given in Table 5.3. As the batteries are power limited to 0.5C in charging and discharging the smaller capacity batteries don't under-go excessive cycle counts. However, performing a full-cycle daily indicates that at some point the battery reached full capacity and excess energy would have been exported to the grid. A higher cycle count would also result in a shorter battery lifespan.

Full-cycle count	Household 1	Household 2
0.5kWh	23	28
1kWh	23	24
2kWh	20	21
5kWh	10	6
7kWh	6	1
15kWh	2	0

Table 5.3: Battery full-cycles for 2015/05/14 to 2015/06/05 inclusive

5.4 Costs

Dependant on the tariff and timing of generation and consumption the cost of importing energy compared to exporting varies between 149% to 604%, so even small fluctuations between power generation and consumption can be multiplied significantly in cost due to timing and tariff. The total costs over the period are given in Table 5.4 and include a base with and without solar generation in addition to a 2kWh and 7kWh battery storage system. The energy costs of solar combined with the range of battery capacities are broken into daily averages and visualised in Figure 5.9.

Immediately it is visible that the PS1 tariff, which favours a reduction in weekday afternoon and evening energy consumption is the most economical for the period analysed. As solar only generates energy during the daylight hours, alone it can only impact on reducing the amount of consumption during the PS1 tariff super-peak (2pm to 8pm) period until sunset. The reduction in energy import and thus super-peak rates, when adding battery storage capacity is evident after sunset until 8pm in Figures 5.10 to 5.15.

The most effectively utilised battery can be selected for each household from the levelling out of daily costs as capacity increases shown in Figure 5.9. For both households this point occurs at the 5 to 7kWh capacities. The yearly savings compared with the household importing all energy consumed are given in Table 5.4, and from this a value/kWh to add battery capacity for each household is calculated. For household 1 this value is $67.68/kWh when adding a 2kWh battery but drops to $35.67/kWh if adding a 7kWh battery, for household 2 these values are $101.06/kWh and $46.32/kWh, respectively. A review by Schoenung [65] found $600(USD)/kWh, assuming the power subsystem costs are covered in the solar PV system installation, as the cost of a suitable Lithium-ion energy storage subsystem a with 10-year lifespan. Given these costs, a 2kWh battery storage system could have been economically viable at the time of solar

PV installation for household 2.

Costs ($)	**A1**	**SM1**	**PS1**
Household 1			
None	55.59	58.61	51.95
Solar	39.63	41.34	35.48
Solar + 2kWh	33.90	31.05	26.58
Solar + 7kWh	26.11	21.15	19.06
Household 2			
None	114.54	124.50	111.73
Solar	61.30	65.36	57.32
Solar + 2kWh	50.93	50.15	44.03
Solar + 7kWh	43.66	40.76	36.00

Table 5.4: Total costs for 2015/05/14 to 2015/06/05 inclusive

Yearly Saving ($)	**Household 1 Yearly Saving**	**Household 2 Yearly Saving**
Solar	\$250.48 (32%)	\$827.49 (49%)
Solar + 2kWh	\$385.84 (49%)	\$1,029.60 (61%)
Solar + 7kWh	\$500.20 (63%)	\$1,151.73 (68%)

Table 5.5: Yearly savings compared to household without generation (PS1 tariff)

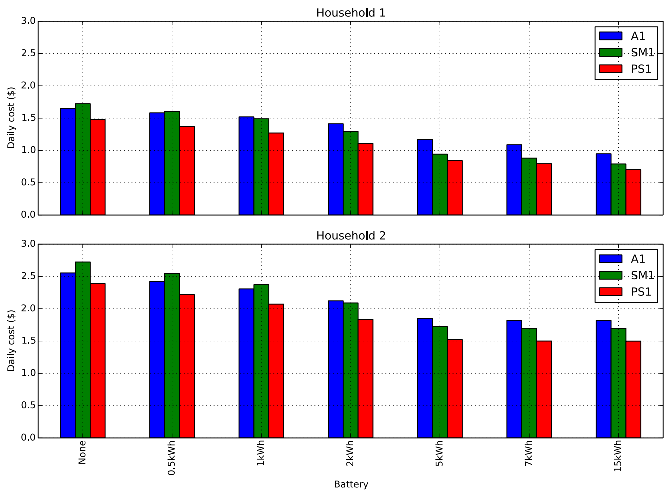

Figure 5.9: Average daily costs with the addition of battery capacity for 2015/05/14 and 2015/06/05 inclusive

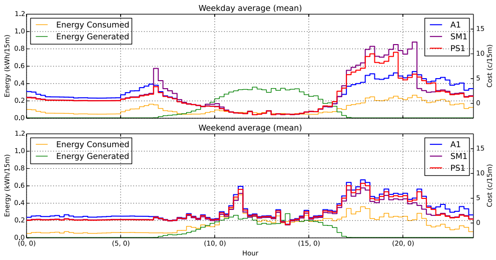

Figure 5.10: Average weekday and weekend energy and tariff costs for household 1

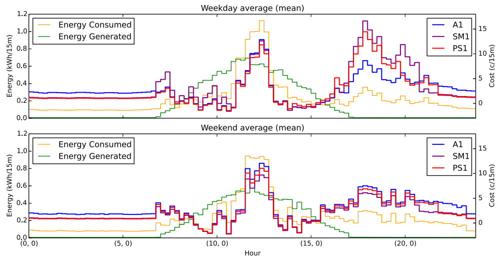

Figure 5.11: Average weekday and weekend energy and tariff costs for household 2

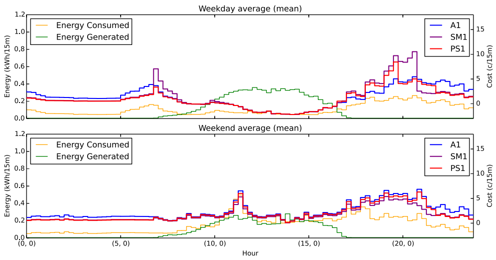

Figure 5.12: Average weekday and weekend energy and tariff costs for household 1 with 2kWh battery storage

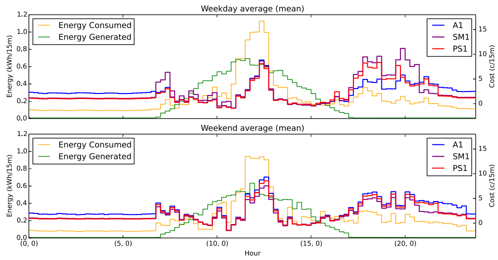

Figure 5.13: Average weekday and weekend energy and tariff costs for household 2 with 2kWh battery storage

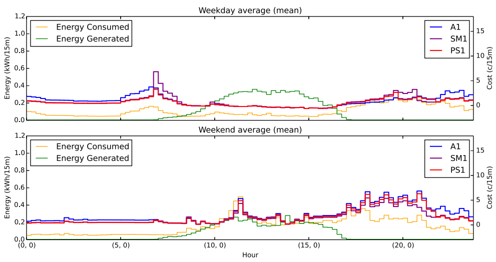

Figure 5.14: Average weekday and weekend energy and tariff costs for household 1 with 7kWh battery storage

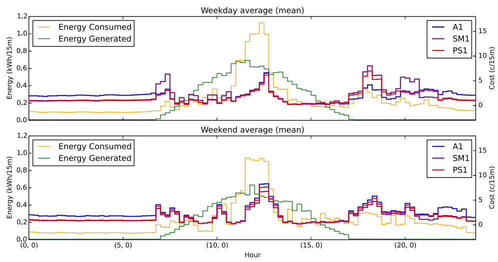

Figure 5.15: Average weekday and weekend energy and tariff costs for household 2 with 7kWh battery storage

5.5 Energy Disaggregation

Disaggregation of appliance energy usage from the power consumption data ingested into the system was attempted using algorithms from both NILMTK [66] and WikiEnergy [67] projects. NILMTK required labeled sub-metered power consumption data to first use to learn appliance models, using the datasets supplied with NILMTK for training to then disaggregating from the power consumption data ingested in our system provided little success. The WikiEnergy algorithm uses a database of appliance models instead of learning. Applying this algorithm using appliance models made from measurements taken from the Tracebase[68] repository, resulted in an inability to match any energy consumption with the models of washing-machine and dishwasher available. These were the only elastic-load appliances tested with. Appliance models were created using the power consumption data shown in Appendix B. More success is likely with either more generalised appliance models or models specific to match the specifications of the appliances in the household in question.

5.6 Limitations

During the development of the system some limitations of the design become apparent.

- **No feedback for notifications** is possible using a service like Pushover to deliver push notification. This means we are unable to determine with the occupant found the notification useful or whether it was even acted upon.
- **Limited solar PV model** not taking into account tilt-angle or reflected/diffuse radiation. The current energy generation prediction algorithm assumes a flat panel, that results in an estimate of equal morning and afternoon generation, which is not the case for an angled panel.
- The **limited query language** in InfluxDB when compared with a traditional SQL database constrained the possible metric generation at query-time.
- **No nested queries** are possible, though some can be achieved using continuous queries these must be kept continuously generating data.
- Results from **continuous queries aren't generated until the end of the period** the query is grouped by. This means that for an hourly average, the current hours average isn't available until the next hour has started.
- **Panning the graphs** in Grafana is not possible, as only data for the time span shown in a graph is queried from the database.

- **Conversion losses are internalised to the battery model** and as such weren't recorded separately from the amount of energy put into or taken out of the battery.
- Battery model simulation **doesn't take into account the best time to discharge** the battery to maximise cost savings for variable-rate tariffs.
- Only a **short duration (24 days) of measurements** were available from the monitoring system to analyse. These measurements were only for a single month so are likely to be biased due to seasonal (winter) factors.

CHAPTER 6

Conclusion

6.1 Future Work

It can be seen from the previous section that there are limitations of the system that could be addressed by future work.

The most beneficial enhancement would be to create a method of determining whether the occupant found the notification to be useful and whether they have, or plan to act on the recommendation. This is likely to require the development of a native mobile application to allow greater flexibility in the presentation and function of notifications on the device.

Currently notifications are sent to a device, or set of devices, with no regard to which of the occupants in a household may be able to apply the recommendation. Given that a typical household has more than one occupant and there is variability in which occupants may be present at a given time, the notification system could be extended to only notify an occupant who was in a position to act on the given recommendation.

For the analysis of the data captured by the monitoring system enhancements could be made to the battery model used for simulations. A model able to optimise charge and discharge with regards to time-variable energy tariffs may prove to be more economically viable. Future work could also include further comparisons between battery parameters, particularly the impact of optimal peak and continuous charge/discharge rates for battery capacity sizing selection.

6.2 Conclusion

Although reducing energy consumption and switching to more energy efficient appliances is a step in the right direction, there is a lower-bound to the energy savings possible with this approach. At this point making more effective use of the variable generation of renewable energy sources requires a shift in consumption to closer follow generation. As we have shown in this study, technology can be applied to create a system capable of monitoring energy consumption and generation, processing data and providing feedback. With improvements in consumption and generation prediction techniques this type of system becomes more relevant in helping to guide behaviours toward efficient energy use.

A model of a battery energy storage system proved useful in simulating the viability of incorporating energy storage into solar PV systems. In particular, a combination with a suitable energy rate tariff can further increase the value of such a system. While costs are unlikely to be viable to retrofit existing installations with low-capacity battery energy storage, they may be as an addition to a system at the time of install.

Disaggregation remains an area of study with ever growing improvements, though as smart devices, with built-in monitoring, become more prevalent its relevance will diminish. A trend toward increased home automation will help with load shifting, allowing elastic-load appliances to schedule themselves to suit the circumstances rather than requiring an occupant to do so manually.

APPENDIX A

System Design Details

Host 1	Host 2
Database	Ingest-1
Dashboard	Ingest-2
	Processing-1
	Task Controller
	Message Queue

Table A.1: Docker containers deployed on each host

```
{
 'weekly-notification': {
      'task': 'tasks.processing.notifier.summary_report',
      'schedule': crontab(hour=18, minute=0,
                          day_of_week='sunday'),
      'args': ('weekly'),
  },
  'efergy-poll': {
      'task': 'tasks.ingest.efergy.poll',
      'schedule': timedelta(seconds=6),
      'args': (API_KEY, HOUSE_ID)
  },
  'allsolus-poll': {
      'task': 'tasks.ingest.allsolus.poll',
      'schedule': timedelta(seconds=600),
      'args': (SITE_ID, DEVICE_ID, MEASUREMENT_ID)
  }
}
```

Figure A.1: Example of schedule dictionary

Task	Ingest Poll
Period	6s
Type	Web service - JSON
Measurement Precision	1s
Measurements	UTC Timestamp (Unix)
	Instantaneous power per transmitter (W)

Table A.2: Efergy Engage ingest

Task	Ingest Backfill Poll
Period	10m
Type	Web service - JSON
Measurement Precision	1m
Measurements	Date-time string
	Instantaneous power per inverter (W)
	Instantaneous power consumption (W)
	Instantaneous battery voltage (V)
	Instantaneous battery current (A)

Table A.3: Allsolus Access ingest

Task	Ingest Backfill Poll
Period	15m
Type	HTTP access to CSV files
Measurement Precision	5m
Measurements	Date-time string
	Average power over 5 minutes (W)
	Total energy generated since install (Wh)

Table A.4: UWA Solar ingest

Task	Ingest Poll
Period	1h
Type	Web service - JSON
Measurement Precision	1h
Measurements	UTC Timestamp (Unix)
	Current temperature (°C)
	Current cloud cover (% of sky)
	Hourly temperature forecast (°C)
	Hourly cloud cover forecast (% of sky)

Table A.5: Forecast.io ingest

Routing key	Destination
`*.*.generation.power`	Stream + DB
`*.*.consumption.power`	Stream + DB
`*.*.energy_metered`	DB
`*.*.battery.voltage`	DB
`*.*.battery.current`	DB
`*.weather.current`	Stream + DB
`*.weather.forecast`	Stream
`*.notification`	Stream + DB

Table A.6: RabbitMQ Routes

```
POST /1/messages.json
Host: api.pushover.net
Content-Type: application/x-www-form-urlencoded
Content-Length: 50

token=API_KEY&user=ID&title=TITLE&message=CONTENT
```

Figure A.2: Example of push notification POST request

APPENDIX B

Household Appliance Usage

The data in Table B.1 is derived from values given in MacKay [69]. Figures B.1,B.2,B.3 are plots of power consumption from measurements captured as part of the Tracebase repository created by Reinhardt et al. [68]

Appliance	Power (kW)	Daily usage (h)	Energy per day (kWh/d)
Tumble dryer	2.5	0.8	2
Dishwasher	2.5	0.6	1.5
Washing machine	2.5	0.4	1

Table B.1: Appliance usage with power and energy consumption

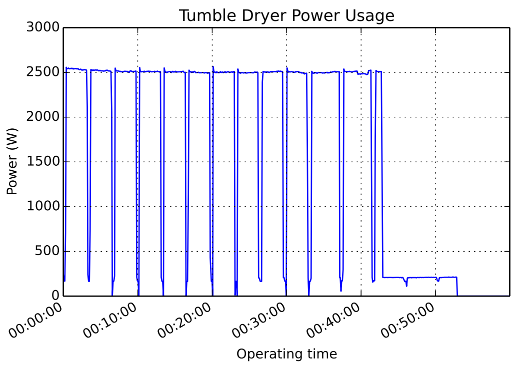

Figure B.1: Tumble dryer power consumption

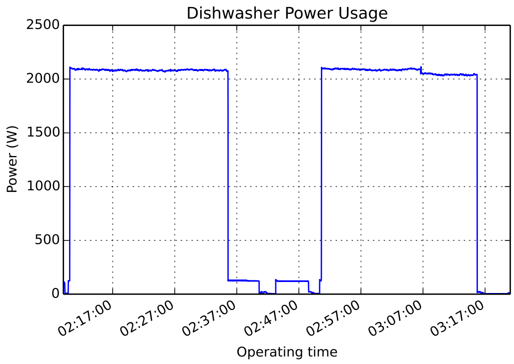

Figure B.2: Dishwasher power consumption

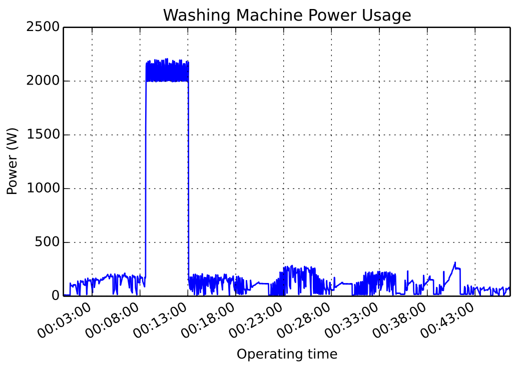

Figure B.3: Washing machine power consumption

APPENDIX C

InfluxDB Time-series

- futurefarm.sma(2001543467).generation.power
- futurefarm.sma(2100473018).generation.power
- futurefarm.smaisland.battery.voltage
- futurefarm.smaisland.battery.current
- futurefarm.smaisland.total.power
- humanmovement.webbox.generation.power
- humanmovement.webbox.energy_metered
- household1.consumption.power
- household1.generation.power
- household2.consumption.power
- household2.generation.power
- household3.consumption.power

APPENDIX D

Battery Model Code Listing

```
class Battery:

    def __init__(self, capacity, discharge_depth=0, max_fullcharge=1, discharge_rate=2400, charge_rate=2400,
                 peak_rate=1.5, peak_rate_period=60, peak_rate_recovery_ratio=1, conversion_loss=0):
        """
        :param capacity: Maximum battery capacity (Wh)
        :param discharge_depth: Minimum level to discharge to (ratio compared to full capacity)
        :param max_fullchage: Maximum level to charge to (ratio compared to full capacity)
        :param discharge_rate: Maximum sustained discharge rate (W)
        :param charge_rate: Maximum sustained charge rate (W)
        :param peak_rate: Peak rate (ratio compared to charge/discharge rate)
        :param peak_rate_period: Maximum continuous peak period (s)
        :param peak_rate_recovery_ratio: Duration to recover from peak rate (ratio compared to peak rate period)
        :param conversion_loss: Loss in energy during conversion to/from storage (ratio lost compared to input amount)
        """

        self.discharge_depth = discharge_depth
        self.max_fullcharge = max_fullcharge
        self.capacity = capacity

        effective_capacity = (capacity * max_fullcharge) - (capacity * discharge_depth)
        self.energy_storage = IdealStorage(effective_capacity)

        self.conversion_loss = conversion_loss

        self.discharge_rate = discharge_rate
        self.charge_rate = charge_rate

        self.peak_rate = peak_rate
        self.peak_rate_period = peak_rate_period
        self.peak_rate_recovery_ratio = peak_rate_recovery_ratio

        self.peak_period_remaining = peak_rate_period
        self.peak_period_recovery = peak_rate_period
        self.peak_period_depleted = False

        self.cycle_count = 0
        self.current_mode = 'HIT_EMPTY'

    def __str__(self):
        return 'Battery - Capacity: %dWh, DoD: %.2f, Full charge: %.2f, Discharge: %dW, Charge: %dW, Peak: %.2f,' \
               ' Peak dur: %ds, Recovery Ratio: %d, Conversion loss: %.2f' % (self.capacity, self.discharge_depth,
                                                                              self.max_fullcharge, self.discharge_rate,
                                                                              self.charge_rate, self.peak_rate,
                                                                              self.peak_rate_period,
                                                                              self.peak_rate_recovery_ratio,
                                                                              self.conversion_loss)

    def charge(self, power, duration):
        """
        Charge battery for duration
        :param power: Constant power for duration (W)
        :param duration: Duration of charge (s)
        :return: Amount of energy stored (Wh)
        """
        if power > self.charge_rate and self.peak_period_remaining >= 0 and not self.peak_period_depleted:
            power = min(power, self.peak_rate * self.charge_rate)
            self.peak_period_remaining -= duration
            if self.peak_period_remaining <= 0 and not self.peak_period_depleted:
                self.peak_period_recovery = self.peak_rate_period
                self.peak_period_depleted = True
        else:
            power = min(power, self.charge_rate)
```

```
            if not self.peak_period_depleted:
                self.peak_period_remaining += duration * self.peak_rate_recovery_ratio
                self.peak_period_remaining = min(self.peak_period_remaining, self.peak_rate_period)

        if self.peak_period_depleted:
            self.peak_period_recovery -= duration * self.peak_rate_recovery_ratio
            if self.peak_period_recovery <= 0:
                self.peak_period_remaining = self.peak_rate_period
                self.peak_period_depleted = False

        assert power >= 0

        energy = power * duration / (60.0 * 60.0)

        stored = self.energy_storage.add_energy(energy)
        losses = stored * self.conversion_loss

        if self.energy_storage.soc >= 1 and self.current_mode != 'HIT_FULL':
            self.current_mode = 'HIT_FULL'
            self.cycle_count += 1

        # Return the amount the battery removed plus how much it wasted in losses
        return stored + losses

    def discharge(self, power, duration):
        """
        Discharge battery for duration
        :param power: Constant power for duration (W)
        :param duration: Duration of discharge (s)
        :return: Amount of energy returned (Wh)
        """
        if power > self.discharge_rate and self.peak_period_remaining >= 0 and not self.peak_period_depleted:
            power = min(power, self.peak_rate * self.discharge_rate)
            self.peak_period_remaining -= duration
            if self.peak_period_remaining <= 0 and not self.peak_period_depleted:
                self.peak_period_recovery = self.peak_rate_period
                self.peak_period_depleted = True
        else:
            power = min(power, self.discharge_rate)
            if not self.peak_period_depleted:
                self.peak_period_remaining += duration * self.peak_rate_recovery_ratio
                self.peak_period_remaining = min(self.peak_period_remaining, self.peak_rate_period)

        if self.peak_period_depleted:
            self.peak_period_recovery -= duration * self.peak_rate_recovery_ratio
            if self.peak_period_recovery <= 0:
                self.peak_period_remaining = self.peak_rate_period
                self.peak_period_depleted = False

        assert power >= 0

        energy = power * duration / (60.0 * 60.0)

        removed = self.energy_storage.remove_energy(energy)
        losses = removed * self.conversion_loss

        if self.energy_storage.soc <= 0 and self.current_mode != 'HIT_EMPTY':
            self.current_mode = 'HIT_EMPTY'
            self.cycle_count += 1

        # Return the amount the battery removed minus how much it wasted in losses
        return removed - losses

    @property
    def stored(self):
        return self.energy_storage.stored + (self.capacity * self.discharge_depth)

    @property
    def soc(self):
        return (float(self.energy_storage.stored) / float(self.capacity)) + self.discharge_depth

    @property
    def full_cycles(self):
        return self.cycle_count / 2


class IdealBattery:
    def __init__(self, capacity):
        """
        :param capacity: Maximum storage capacity (Wh)
        """
        self.capacity = capacity
        self.energy_storage = IdealStorage(capacity)
```

```python

        self.cycle_count = 0
        self.current_mode = 'HIT_EMPTY'

    def __str__(self):
        return 'IdealBattery - Capacity: %dWh' % self.energy_storage.capacity

    def charge(self, power, duration):
        """
        Charge battery for duration
        :param power: Constant power for duration (W)
        :param duration: Duration of charge (s)
        :return: Amount of energy stored (Wh)
        """
        assert power >= 0

        energy = power * duration / (60.0 * 60.0)

        if self.energy_storage.soc >= 1 and self.current_mode != 'HIT_FULL':
            self.current_mode = 'HIT_FULL'
            self.cycle_count += 1

        stored = self.energy_storage.add_energy(energy)
        return stored

    def discharge(self, power, duration):
        """
        Discharge battery for duration
        :param power: Constant power for duration (W)
        :param duration: Duration of discharge (s)
        :return: Amount of energy returned (Wh)
        """
        assert power >= 0

        energy = power * duration / (60.0 * 60.0)
        removed = self.energy_storage.remove_energy(energy)

        if self.energy_storage.soc <= 0 and self.current_mode != 'HIT_EMPTY':
            self.current_mode = 'HIT_EMPTY'
            self.cycle_count += 1

        return removed

    @property
    def stored(self):
        return self.energy_storage.stored

    @property
    def soc(self):
        return self.energy_storage.soc

    @property
    def full_cycles(self):
        return self.cycle_count / 2



class IdealStorage:
    def __init__(self, capacity):
        """
        :param capacity: Maximum storage capacity
        """

        self.capacity = capacity
        self.stored = 0

    def __str__(self):
        return 'Ideal EnergyStorage - Capacity: %dWh' % self.capacity

    def add_energy(self, amount):
        """
        Add energy to storage

        :param amount: Energy to try to store (Wh)
        :return: Actual amount stored (Wh)
        """

        assert amount >= 0

        available_capacity = self.capacity - self.stored
        amount_to_store = min(available_capacity, amount)
        self.stored += amount_to_store

```

```
        assert self.stored <= self.capacity

        return amount_to_store

    def remove_energy(self, amount):
        """
        Remove energy from storage

        :param amount: amount to try and remove
        :return: actual amount removed
        """

        assert amount >= 0

        amount_to_remove = min(self.stored, amount)
        self.stored -= amount_to_remove

        assert self.stored >= 0

        return amount_to_remove

    @property
    def soc(self):
        """
        State of charge
        :return: Ratio of stored amount to capacity
        """
        if self.capacity == 0:
            return 0.0
        return float(self.stored) / float(self.capacity)
```

Bibliography

[1] Environmental Issues: Energy Use and Conservation. Australian Bureau of Statistics, December 2014.

[2] Australia Reaches 4GW Installed Solar. Australian PV Institute, December 2014.

[3] Vic Smith. Small Scale Domestic Rooftop Solar Photovoltaic Systems. Technical report, University of Wollongong, October 2011.

[4] A Chersin, W Ongsakul, and J Mitra. Improving of uncertain power generation of rooftop solar PV using battery storage. In *Green Energy for Sustainable Development (ICUE), 2014 International Conference and Utility Exhibition on*, pages 1–4, 2014.

[5] Jack Gilding. Australian Photovoltaic Institute Market Analyses. July 2014.

[6] H Brohus, P Heiselberg, and A Simonsen. Uncertainty of energy consumption assessment of domestic buildings. In *Eleventh International IBPSA Conference*, Glasgow, 2009.

[7] Sarah Darby. The Effectiveness of Feedback on Energy Consumption. Technical report, University of Oxford, April 2006.

[8] Hieu Huynh Chi. An effective approach using Eco-feedback to motivate energy conservation behaviors. In *Collaboration Technologies and Systems (CTS), 2014 International Conference on*, pages 602–606. IEEE, 2014.

[9] S S van Dam, C A Bakker, and J D M van Hal. Home energy monitors: impact over the medium-term. *Building Research & Information*, 38(5): 458–469, October 2010.

[10] J M Rosok. Combining smart energy meters with social media: Increasing energy awareness using data visualization and persuasive technologies. In *Collaboration Technologies and Systems (CTS), 2014 International Conference on*, pages 27–32. IEEE, 2014.

[11] Seungwoo Lee, Daye Ahn, Sukjun Lee, Rhan Ha, and Hojung Cha. Personalized Energy Auditor: Estimating personal electricity usage. In *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on*, pages 44–49. IEEE, 2014.

[12] Nilanjan Banerjee, Sami Rollins, and Kevin Moran. *Automating energy management in green homes*. ACM, New York, New York, USA, August 2011.

[13] G W Hart. Nonintrusive appliance load monitoring. In *Proceedings of the IEEE*, pages 1870–1891. IEEE, 1992.

[14] Jing Liao, Georgia Elafoudi, Lina Stankovic, and Vladimir Stankovic. Non-intrusive appliance load monitoring using low-resolution smart meter data. *2014 IEEE International Conference on Smart Grid Communications*, pages 535–540, 2014.

[15] Hyungsul Kim, Manish Marwah, Martin Arlitt, Geoff Lyon, and Jiawei Han. Unsupervised Disaggregation of Low Frequency Power Measurements. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2011.

[16] Nipun Batra, Oliver Parson, Mario Berges, Amarjeet Singh, and Alex Rogers. A comparison of non-intrusive load monitoring methods for commercial and residential buildings. *CoRR*, abs/1408.6595, 2014.

[17] R Dong, L Ratliff, H Ohlsson, and S S Sastry. A dynamical systems approach to energy disaggregation. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 6335–6340. IEEE, 2013.

[18] P Heracleous, P Angkititraku, and K Takeda. Stochastic modeling and disaggregation of energy-consumption behavior. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 8277–8281. IEEE, 2014.

[19] A Zoha, A Gluhak, M A Imran, and S Rajasegarar. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors*, 2012.

[20] K Carrie Armel, Abhay Gupta, Gireesh Shrimali, and Adrian Albert. Is disaggregation the holy grail of energy efficiency? The case of electricity. *Energy Policy*, 52:213–234, January 2013.

[21] John A Duffie and William A Beckman. *Solar Engineering of Thermal Processes*. John Wiley & Sons, April 2013.

[22] Srinivasan Iyengar, Navin Sharma, David Irwin, Prashant Shenoy, and Krithi Ramamritham. *SolarCast: a cloud-based black box solar predictor for smart homes*. a cloud-based black box solar predictor for smart homes. ACM, New York, New York, USA, November 2014.

[23] A Sharma, C R Chen, and N V Lan. Solar-energy drying systems: A review. *Renewable and Sustainable Energy Reviews*, 2009.

[24] Jacquelin Burgess and Michael Nye. Re-materialising energy use through transparent monitoring systems. *Energy Policy*, 36(12):4454–4459, December 2008.

[25] E Shove. Changing human behaviour and lifestyle: a challenge for sustainable consumption. *Department of Sociology*, 2003.

[26] Tom Hargreaves, Michael Nye, and Jacquelin Burgess. Making energy visible: A qualitative field study of how householders interact with feedback from smart energy monitors. *Energy Policy*, 38(10):6111–6119, October 2010.

[27] M Masoodian, B Endrass, R Buhling, P Ermolin, and E Andre. Time-Pie visualization: Providing Contextual Information for Energy Consumption Data. In *Information Visualisation (IV), 2013 17th International Conference*, pages 102–107. IEEE, 2013.

[28] Tiffany Grace Holmes. *Eco-visualization: combining art and technology to reduce energy consumption*. ACM, June 2007.

[29] Wokje Abrahamse, Linda Steg, Charles Vlek, and Talib Rothengatter. A review of intervention studies aimed at household energy conservation. *Journal of Environmental Psychology*, 25(3):273–291, September 2005.

[30] Corinna Fischer. Feedback on household electricity consumption: a tool for saving energy? *Energy Efficiency*, 1(1):79–104, 2008.

[31] B F Hobbs. Optimization methods for electric utility resource planning. *European Journal of Operational Research*, 1995.

[32] Aditya Mishra, David Irwin, Prashant Shenoy, Jim Kurose, and Ting Zhu. *SmartCharge: cutting the electricity bill in smart homes with energy storage*. ACM, May 2012.

[33] Andreas Veit, Christoph Goebel, Rohit Tidke, Christoph Doblander, and Hans-Arno Jacobsen. *Household electricity demand forecasting: benchmarking state-of-the-art methods*. ACM, June 2014.

[34] T D Hund, S Gonzalez, and K Barrett. Grid-Tied PV system energy smoothing. In *Photovoltaic Specialists Conference (PVSC), 2010 35th IEEE*, pages 002762–002766. IEEE, 2010.

[35] A Zahedi. Maximizing solar PV energy penetration using energy storage technology. *Renewable and Sustainable Energy Reviews*, 15(1):866–870, January 2011.

[36] Xiangjun Li, Dong Hui, and Xiaokang Lai. Battery Energy Storage Station (BESS)-Based Smoothing Control of Photovoltaic (PV) and Wind Power Generation Fluctuations. *Sustainable Energy, IEEE Transactions on*, 4(2): 464–473, 2013.

[37] J Z Kolter, Siddharth Batra, and Andrew Y Ng. Energy Disaggregation via Discriminative Sparse Coding. *Advances in Neural Information Processing Systems*, pages 1153–1161, 2010.

[38] Sidhant Gupta, Matthew S Reynolds, and Shwetak N Patel. *ElectriSense: single-point sensing using EMI for electrical event detection and classification in the home*. single-point sensing using EMI for electrical event detection and classification in the home. ACM, New York, New York, USA, September 2010.

[39] C Laughman, Kwangduk Lee, R Cox, S Shaw, S Leeb, L Norford, and P Armstrong. Power signature analysis. *Power and Energy Magazine, IEEE*, 1(2):56–63, 2003.

[40] Shwetak N Patel, Thomas Robertson, Julie A Kientz, Matthew S Reynolds, and Gregory D Abowd. At the Flick of a Switch: Detecting and Classifying Unique Electrical Events on the Residential Power Line (Nominated for the Best Paper Award). In *UbiComp 2007: Ubiquitous Computing*, pages 271–288. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[41] Mario Berges, Ethan Goldman, H Scott Matthews, and Lucio Soibelman. Training load monitoring algorithms on highly sub-metered home electricity consumption data. *Tsinghua Science and Technology*, 13:406–411, 2008.

[42] M Baranski and J Voss. Genetic algorithm for pattern detection in NIALM systems. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, pages 3462–3468. IEEE, 2004.

[43] D C Bergman, Dong Jin, J P Juen, N Tanaka, C A Gunter, and A K Wright. Distributed non-intrusive load monitoring. *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*, pages 1–8, 2011.

[44] J Z Kolter and M J Johnson. REDD: A public data set for energy disaggregation research. *Workshop on Data Mining Applications in Sustainability*, 2011.

[45] E Vogiatzis, G Kalogridis, and S Z Denic. Real-time and low cost energy disaggregation of coarse meter data. *Innovative Smart Grid Technologies Europe (ISGT EUROPE), 2013 4th IEEE/PES*, pages 1–5, 2013.

[46] G W Hart. Residential energy monitoring and computerized surveillance via utility power flows. *Technology and Society Magazine*, 1989.

[47] S Barker, S Kalra, and D Irwin. NILM Redux: The Case for Emphasizing Applications over Accuracy. *... Load Monitoring (NILM)*, 2014.

[48] Standard Electricity Pricesand Charges. pages 1–20, August 2014.

[49] Docker. URL `https://www.docker.com`.

[50] DigitalOcean. URL `https://www.digitalocean.com`.

[51] Carl Boettiger. An introduction to Docker for reproducible research, with examples from the R environment. *arXiv.org*, (1):71–79, October 2014.

[52] Celery, . URL `http://www.celeryproject.org`.

[53] Python. URL `https://www.python.org`.

[54] Forecast.io, . URL `http://forecast.io`.

[55] RabbitMQ. URL `http://www.rabbitmq.com`.

[56] Pika, . URL `https://github.com/pika/pika`.

[57] InfluxDB, . URL `http://influxdb.com`.

[58] InfluxDB Storage Exchange. URL `https://github.com/aweber/`.

[59] Grafana. URL `http://grafana.org`.

[60] AngularJS. URL `https://angularjs.org`.

[61] Flot. URL `http://www.flotcharts.org`.

[62] Pushover. URL `https://pushover.net`.

[63] PySolar. URL `http://pysolar.org`.

[64] Pandas, . URL `http://pandas.pydata.org`.

[65] S Schoenung. *Energy Storage Systems Cost Update (SAND2011-2730)*. Sandia National Laboratories, 2011.

[66] Nipun Batra, Jack Kelly, Oliver Parson, Haimonti Dutta, William Knottenbelt, Alex Rogers, Amarjeet Singh, and Mani Srivastava. *NILMTK: an open source toolkit for non-intrusive load monitoring.* ACM, June 2014.

[67] Oliver Parson. *Unsupervised Training Methods for Non-intrusive Appliance Load Monitoring from Smart Meter Data.* PhD thesis, University of Southampton, 2014.

[68] A Reinhardt, P Baumann, D Burgstahler, M Hollick, H Chonov, M Werner, and R Steinmetz. On the accuracy of appliance identification based on distributed load metering data. *Sustainable Internet and ICT for Sustainability (SustainIT), 2012*, pages 1–9, 2012.

[69] D MacKay. Sustainable Energy-without the hot air, 2008.