Imperial College London

Smart Meter Disaggregation: Data Collection & Analysis

Jack Kelly & William Knottenbelt

dk3810@imperial.ac.uk
Department of Computing, Imperial College London

1) INTRODUCTION

1.1) Motivation

- Every house in the UK will have a smart meter by 2020.
- Smart meters measure whole-house aggregate power consumption.
- Disaggregated, appliance-by-appliance information enables consumers to manage their electricity consumption most effectively.

1.2) Aim of disaggregation

- Infer which appliances are active & the energy used by each appliance given only the whole-house aggregate smart meter signal. Appliance-by-appliance sub-metering is not required.

1.3) Contributions described this poster

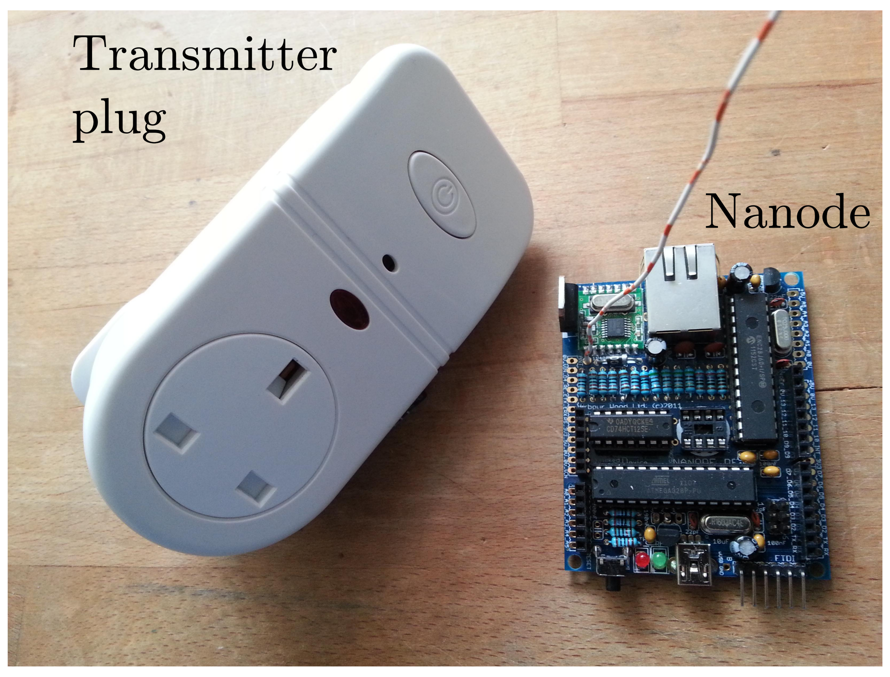
1. Low-cost, open-source data collection system.
2. Presentation of open-access dataset & analysis.
3. Plans for open-source disaggregation system.

2) RECORDING SYSTEM

- Research into disaggregation is inhibited by a lack of real data.
- We present a low-cost, open-source, wireless system for collecting "ground truth" power data from multiple appliances per home as well as recording whole-house voltage and current waveforms at 44.1 kHz.
- Our code is open source on github. Visit jack-kelly.com/energycode

2.1) Recording appliance power

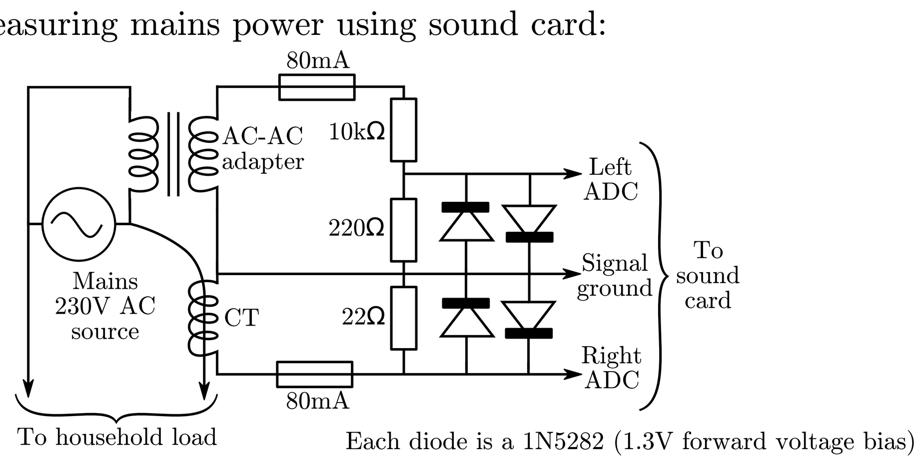
- Low-cost "EDF EcoManager Transmitter Plugs" on each appliance.
- Wirelessly reports active power once every six seconds.
- Off-the-shelf base station is not suitable for our purposes.
- Built our own wireless base station on a Nanode. Nanodes include an ATmega 328P microcontroller & a HopeRF RFM12b RF module.



2.2) Recording whole-house power

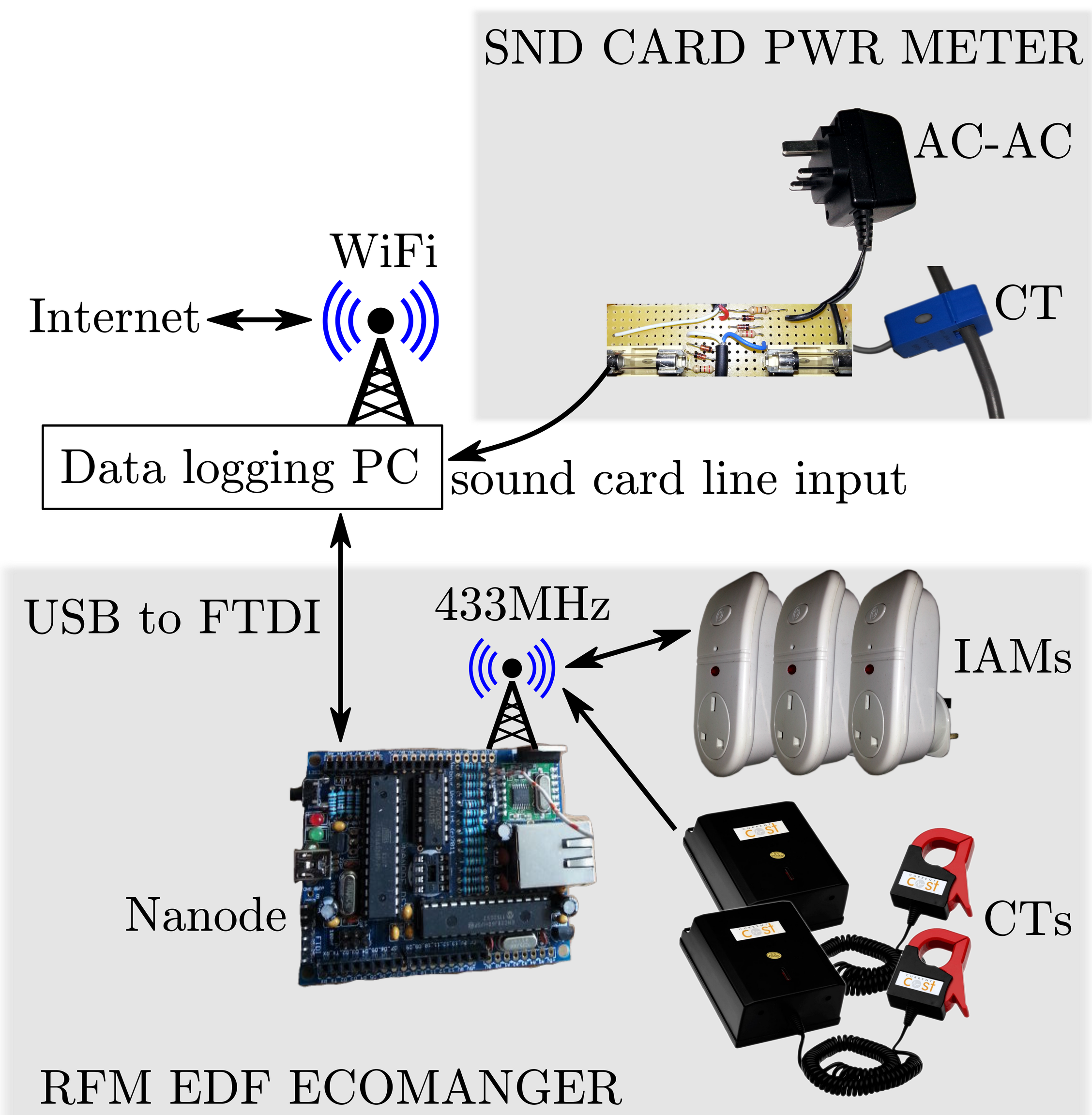
- Home energy monitors do not measure active power or reactive power or voltage. (OpenEnergyMonitor does but its resolution is ~14W)
- "Proper" (SMETS2) smart meters are not yet available.
- We built our own meter using a "current transformer" (CT) clamp to measure mains current; an AC-AC adapter ("wall wart") to measure mains voltage & a PC sound card as an analogue to digital converter.
- Records V and I at 44.1kHz and calculates active & apparent power.

Circuit for measuring mains power using sound card:



Each diode is a 1N5282 (1.3V forward voltage bias)

Our complete data collection system:

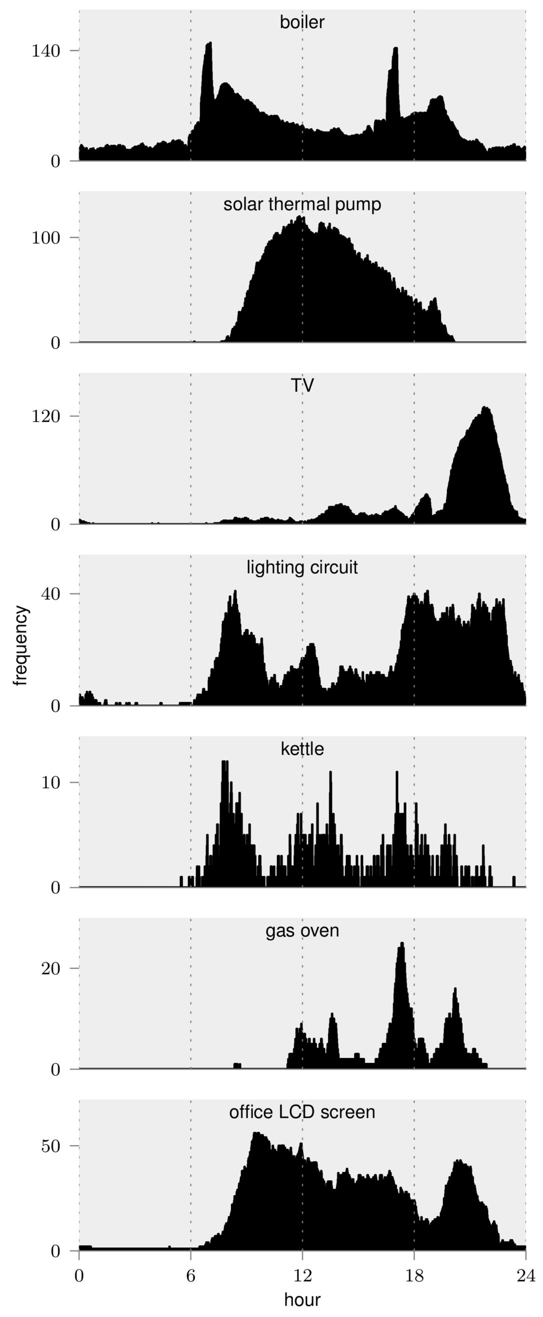


IAM = Individual Appliance Monitor;
CT = Current Transformer;
AC-AC = transformer;
FTDI = RS-232 serial at TTL voltages;
Nanode = open-source rapid development board; a bit like an Arduino with RF built in

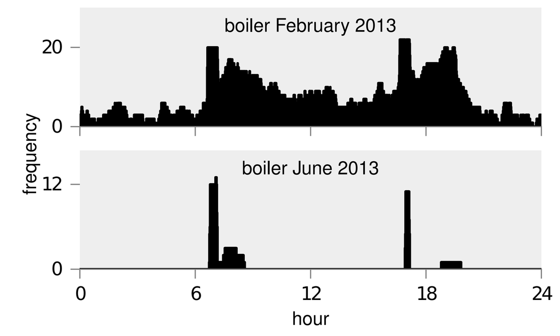
3) DATASET

- We recorded power data from four UK houses over multiple months.
- Available from jack-kelly.com/powerdata
- First public UK dataset with temporal resolution less than 2 minutes.
- We present an analysis of this dataset, focussing on patterns and correlations which could be learnt by a disaggregation system.

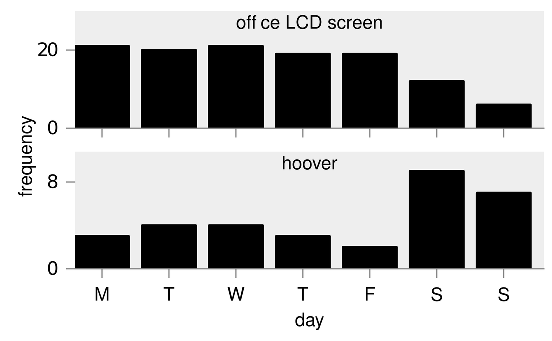
Histograms showing appliance usage over average day:



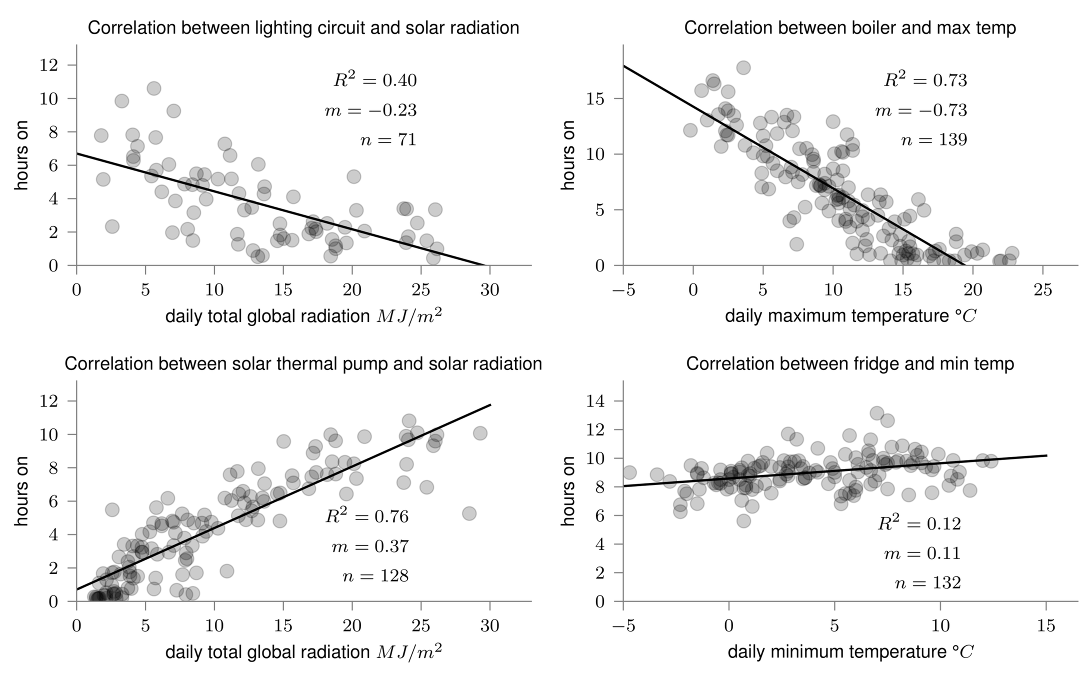
Histograms showing seasonal variation in daily boiler usage:



Histograms showing weekly appliance usage:

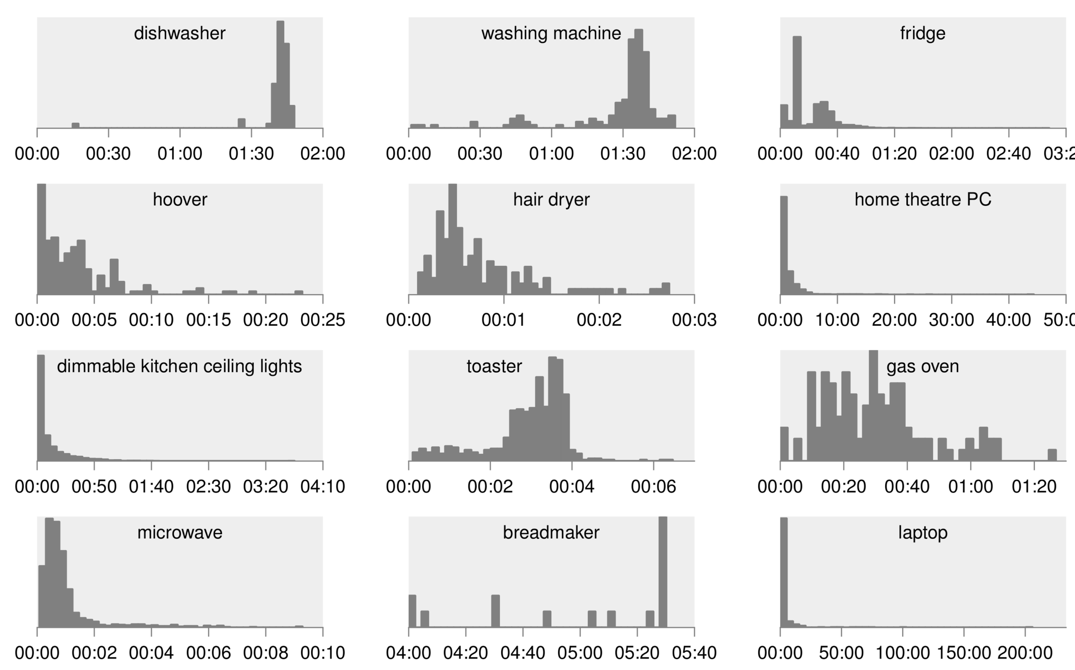


Correlations between appliance usage and weather:



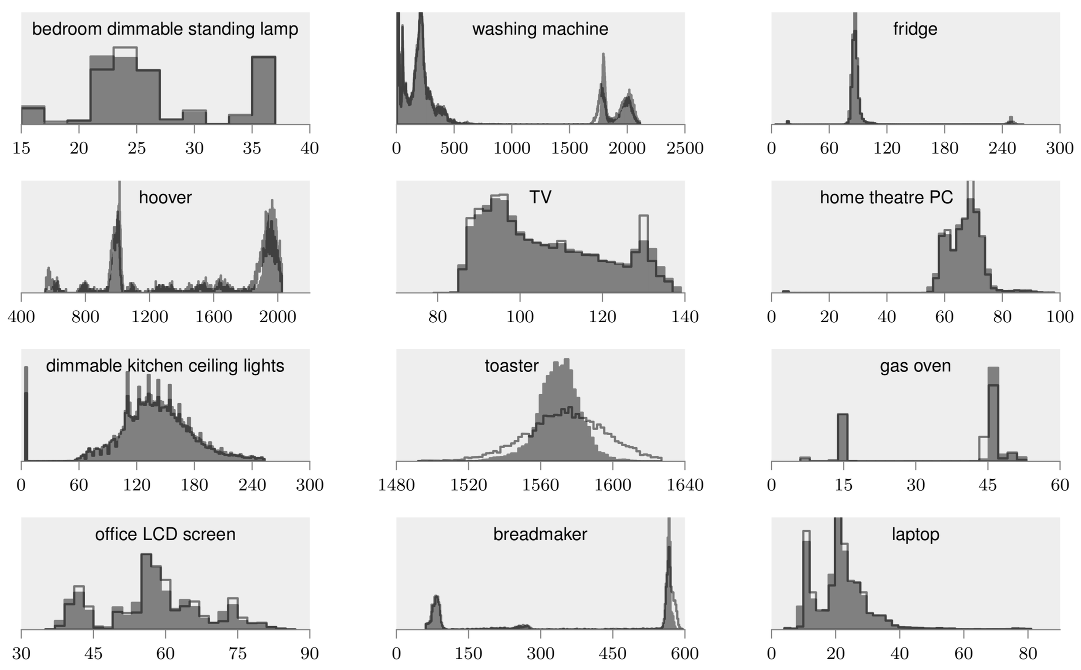
Each data-point represents one day. Historical daily averages from Heathrow weather station (20 miles due west of the premises under consideration) were obtained from the UK Met Office under their Educational program. Days for which the appliance usage was zero were ignored because we assume that the house was unoccupied on these days.

Histograms of appliance on-durations:



Horizontal axis denotes duration formatted as HH:MM.
Vertical axis indicates frequency.

Histograms of appliance power consumption::



Horizontal axis denotes appliance power in watts. Vertical axis indicates frequency. The filled grey plots show histograms of "normalised power". The thin, grey, semi-transparent lines drawn over the filled plots show histograms of un-normalised power. Normalisation is calculated using the following formula:

$$P_{Norm}(t) = \left(\frac{230}{V(t)}\right)^2 \times P(t)$$

4) DISAGGREGATION

We propose to build an open-source disaggregation system which will process data using the following four stages:

1) Process smart meter data with a bank of feature detectors
2) Decode features into multiple probabilistic appliance hypotheses
3) Refine hypotheses using a probabilistic graphical model representing higher-order relationships
4) Further refine hypotheses by reconstructing appliance waveforms and fitting these reconstructed waveforms to the aggregate signal

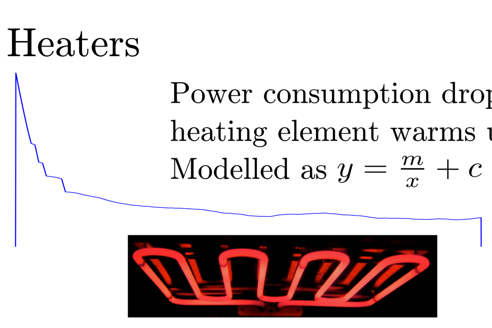
Steps 1) and 3) are illustrated below:

4.1) Feature extraction

- All existing disaggregation algorithms we are aware of extract a single feature from the smart meter signal: changes in steady states.
- We believe that disaggregation performance can be enhanced by extracting additional features, especially:

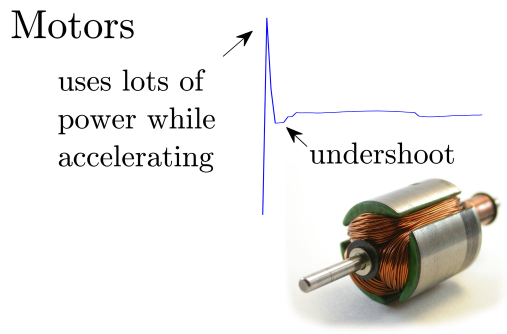
Heaters

Power consumption drops as heating element warms up. Modelled as $y = \frac{m}{x} + c$



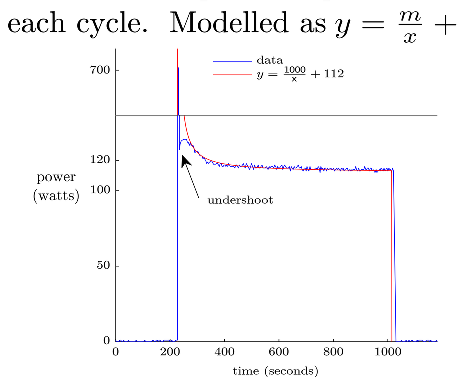
Motors

uses lots of power while accelerating



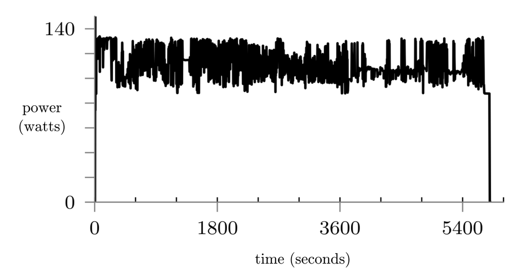
Fridges

Power consumption drops across each cycle. Modelled as $y = \frac{m}{x} + c$



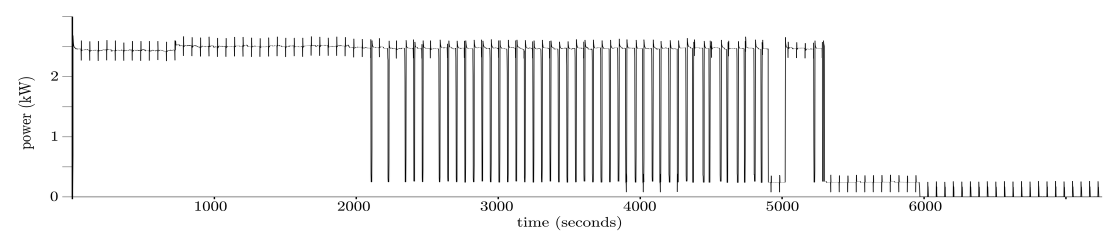
Screens

Power consumption varies rapidly as screen brightness varies.



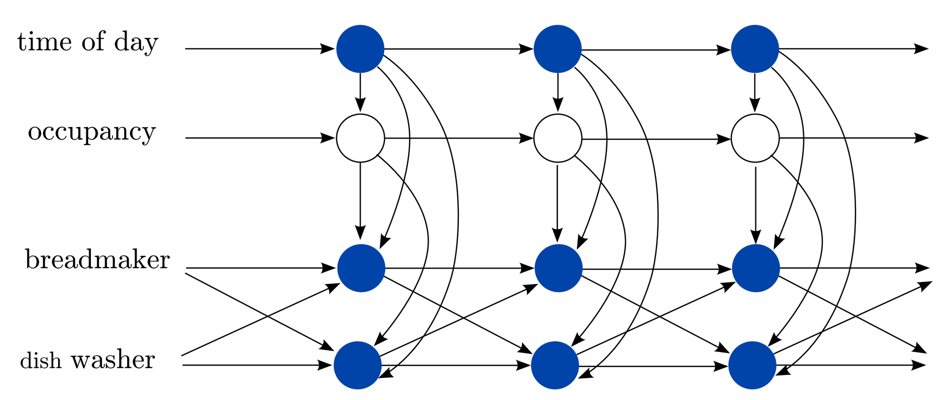
Repeatedly turning components on and off.

Appliances which need to modulate large loads like heating elements tend to do so by turning the load on and off rapidly. These repeated on-off events should be detectable. Below is a power signature of a tumble drier. The rapid 2kW on/off events are the tumble drier modulating its heater:



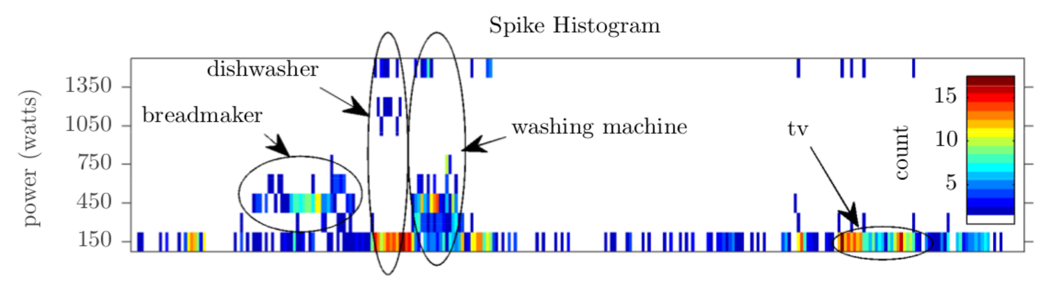
4.2) Model higher-order relationships

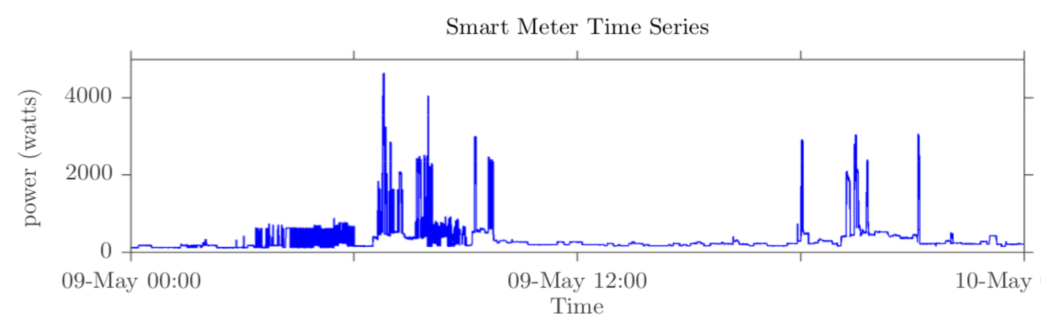
- Use a probabilistic graphical model (e.g. a Dynamic Bayes Net) to model:
- Correlations between appliances
- Hidden parameters e.g. occupancy



5) PRELIMINARY RESULTS

- Implemented a feature-detector designed to find frequent on-off events
- The figure below shows the output of our "spike histogram" feature detector in the top panel and the input smart meter time series in the lower panel.
- The time series is broken into 3 minute slices. For each time slice, a histogram of the forward difference is calculated. Each column in the figure represents a time slice.
- This feature detector is capable of resolving the differences between several appliances (manually annotated in the figure below).
- This will be one of several feature detectors.





6) NEXT STEPS

- Implement feature detectors and a complete disaggregation system
- Characterise the performance of our disaggregation system using our own dataset and other datasets.

ACKNOWLEDGEMENTS

This work was funded by a EPSRC DTA and an Intel PhD Fellowship.