

# Household Occupancy Monitoring Using Electricity Meters

**Wilhelm Kleiminger**  
Dept. of Computer Science  
ETH Zurich, Switzerland  
wilhelmk@ethz.ch

**Christian Beckel**  
Dept. of Computer Science  
ETH Zurich, Switzerland  
beckel@inf.ethz.ch

**Silvia Santini**  
Embedded Systems Lab  
TU Dresden, Germany  
silvia.santini@tu-dresden.de

## ABSTRACT

Occupancy monitoring (*i.e.* sensing whether a building or room is currently occupied) is required by many building automation systems. An automatic heating system may, for example, use occupancy data to regulate the indoor temperature. Occupancy data is often obtained through dedicated hardware such as passive infrared sensors and magnetic reed switches. In this paper, we derive occupancy information from electric load curves measured by off-the-shelf smart electricity meters. Using the publicly available ECO dataset, we show that supervised machine learning algorithms can extract occupancy information with an accuracy between 83% and 94%. To this end we use a comprehensive feature set containing 35 features. Thereby we found that the inclusion of features that capture changes in the activation state of appliances provides the best occupancy detection accuracy.

## Author Keywords

Occupancy detection; Electricity consumption; Opportunistic sensing; Smart meter.

## ACM Classification Keywords

H.m. Information systems: Miscellaneous.

## INTRODUCTION

Home and building automation systems may help to save energy and contribute to the occupants' level of comfort [24]. To this end, such systems often need to determine the *occupancy state* of a building or room – *i.e.* to estimate whether the residents are present or not. In a residential setting, for instance, an automatic heating system monitors occupancy to keep the building at a comfortable temperature *whenever the residents are at home*. At the same time, the system avoids unnecessary energy waste by allowing the temperature to drop *whenever the household is unoccupied* [2, 20, 21].

Current building automation systems typically use dedicated sensing devices such as passive infrared (PIR) sensors and reed switches to provide occupancy monitoring capabilities [1, 21, 29]. Recent results also show the feasibility of opportunistically using network logins and GPS trackers to

monitor occupancy [10, 17, 20, 24]. In such systems, sensor readings are often combined to increase the overall occupancy detection accuracy. For instance, the system presented in [2] combines door-mounted magnetic reed switches and PIR sensors to compensate for the poor accuracy obtained when only PIR sensors are used. The number and type of sensors included in an occupancy monitoring system typically result from a trade-off between the required occupancy detection accuracy and the overall cost and complexity of the system. For this reason, commercial smart thermostats for the residential environment often only include a single PIR sensor. This restricts their ability to accurately monitor the occupancy throughout the building and results in erroneous control decisions. As a result, users of such smart thermostats often turn off automatic control to regain authority over the system [32].

In this paper, we discuss and quantitatively evaluate the suitability of digital electricity meters to be used for occupancy monitoring in residential buildings. Being already present – or about to be installed – in millions of households worldwide, the installation, use and maintenance of smart meters does not impose additional costs on the residents. The opportunistic use of existing sensors thus increases the occupancy monitoring capabilities and therefore the acceptance of building automation systems.

The fact that electricity consumption measurements might indicate the presence of residents in a household is intuitive and has already been observed earlier [18, 23]. However there does not yet exist a comprehensive, quantitative analysis of the accuracy achievable using electricity meters to detect household occupancy. This is mainly due to the lack of data sets containing both electricity consumption measurements and ground truth occupancy data. To overcome this problem, we have collected and published<sup>1</sup> a data set that contains both electricity consumption measurements and occupancy information [4]. The data set includes records of the electricity consumption of both the whole household and of selected appliances. The data is available for five different households and for a period of more than six months.

In our previous work [18], we report a preliminary analysis of this data set and show that digital electricity meters are suitable to be used as occupancy sensors. In this paper, we build and improve upon our previous work and present a detailed analysis of supervised machine learning approaches to detect occupancy from electricity consumption data. We make the following contributions:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or to publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).  
*UbiComp '15*, September 07 - 11, 2015, Osaka, Japan  
Copyright is held by the owner/author(s). Publication rights licensed to ACM.  
ACM 978-1-4503-3574-4/15/09...\$15.00  
DOI: <http://dx.doi.org/10.1145/2750858.2807538>

<sup>1</sup><http://vs.inf.ethz.ch/res/show.html?what=eco-data>. See also [4].

- **Exhaustive analysis of the feature space:** To investigate which characteristics of the load curve best reveal occupancy, we extend the feature set of our preliminary work [18] from 10 to 35 features. To deal with the extended feature set, we employ dimensionality reduction, in particular principal component analysis (PCA) and feature selection. We show that features capturing appliance state changes are best suited for occupancy detection.
- **Improved classification performance:** We consider a set of seven classifiers (instead of 4 in our previous work) and show that the enlarged feature space in conjunction with dimensionality reduction achieves classification accuracies up to 94%.
- **Feasibility analysis with regards to smart heating:** By analysing the ability of our approach to monitor occupancy transitions, we evaluate the feasibility of using digital electricity meters to provide occupancy data for controlling a smart thermostat.

## RELATED WORK

The approaches most related to our work can be found in the fields of building occupancy detection, nonintrusive load monitoring and analysis of electricity consumption data.

### Building occupancy detection

Several authors have focussed on the design, deployment and evaluation of approaches to detect occupancy both in commercial and residential buildings. A good overview of existing approaches for occupancy detection is provided by Nguyen and Aiello [24]. One observation made by the authors is that only a small fraction of the approaches presented in the literature have been evaluated in real deployments. The authors thus stress “*a vital need*” to verify conceptual results in “*real-life installations*” [24]. Recent work shows results from such real-life deployments [1, 7, 8, 21].

However, most occupancy monitoring systems require dedicated hardware and are evaluated over a short time horizon only. While some authors combine PIR sensors with reed switches [1, 21, 29], several authors have also combined PIR sensors with call monitoring [7] or microphones [25], while another solution requires dedicated camera networks [8]. Lu *et al.* instrumented eight homes with PIR sensors and reed switches for a duration varying from one to two weeks depending on the household. In a similar approach, Agarwal *et al.* describe the control of a heating, ventilation and cooling (HVAC) system in a university building [1]. These approaches are similar to our work because they investigate and quantitatively evaluate the use of specific sensors to detect occupancy. However, instead of relying on a dedicated infrastructure, we explore the possibility of using off-the-shelf digital electricity meters – which are becoming mandatory in many countries [26] – to reduce the need for additional hardware. Furthermore, our analysis relies on data collected over significantly longer periods of time than previous work.

### Nonintrusive load monitoring (NILM)

A number of authors have looked into so-called nonintrusive load monitoring (NILM) approaches to infer the disaggregated (*i.e.* device-level) electricity consumption from aggregate data. NILM is closely related to our approach as the activation state of home appliances may give an indication of the current activity of the occupants and thus the buildings occupancy. Froehlich *et al.* provide an overview of these techniques in [9].

Early NILM research was led by George Hart, who used device signatures based on step changes in the electricity consumption to detect individual appliances [12]. However, follow-up work has shown that current algorithms are only able to reliably detect a few appliances (*e.g.* cooling appliances or the washing machine) when the electricity consumption is sampled at a granularity of 1 Hz [4]. Our results show that these appliances are not suitable for occupancy monitoring as their operation exhibits a low correlation with occupancy.

More recent approaches make use of transient [27] or continuous [11] electrical noise on the power line to detect the activation state of appliances. However, like the approach by Hart, these require additional instrumentation and training. Froehlich *et al.* note that the calibration requires users to “*walk around the home, activating and deactivating each device or appliance at least once*” [9].

In contrast to most NILM approaches, our system only requires the annotation of the occupancy state of the household. While asking the user to supply these annotations is still burdensome, the effort is small when compared to calibrating all appliances. In addition, the effort can be significantly reduced by running a simple heuristic unsupervised occupancy detection (*e.g.* by comparing the current electricity consumption to the mean of the nighttime consumption) and proposing possible ground truth occupancy schedules to the user.

### Occupancy and the electric load curve

In [23], Molina-Markham *et al.* suggest that household activities can be inferred from aggregate electricity consumption data. They collected data at 1 Hz from three homes over two months and let occupants annotate which appliances they have used at what time. The authors observed that there are differences in the consumption data depending on whether the occupants are present or absent. However, this observation is based on visual inspection of the electric load curves. No quantitative analysis of the possibility of using aggregate electricity consumption data to automatically detect occupancy is provided.

In a recent workshop publication we presented the results of a preliminary analysis of an occupancy monitoring infrastructure relying on electricity consumption data [18]. At the same workshop, Chen *et al.* discussed the potential of digital electricity meters to be used for performing non-intrusive occupancy monitoring [6]. In particular, they presented a threshold-based method to detect occupancy from aggregate electricity consumption data. The authors evaluated their method using data collected in two homes over a summer

week. We build upon this and our previous work by considering a large set of features including those used by Chen *et al.* and base our analysis on a data set collected in five homes and over a period of more than six months. While in [18] we focussed on the description of the data set and the presentation of preliminary, encouraging results, this paper presents a more detailed analysis of the potential of common digital electricity meters to be used as occupancy sensors.

Other authors utilise device-level information to monitor occupancy. Ming *et al.* suggest a zero-training algorithm based on rough estimates of the participants’ working schedules [16]. Their so-called “PresenceSense” approach estimates occupancy in an office environment using the average power, standard deviation and absolute maximum power change of individual appliances measured by ACme nodes [15]. In contrast to PresenceSense, our work requires only the installation of a single digital electricity meter and focusses on residential buildings.

### ECO DATA SET

For our analysis we use the publicly available Electricity Consumption and Occupancy (ECO) data set. To the best of our knowledge, this data set is the largest one containing both electricity consumption data and ground truth occupancy information<sup>2</sup>. We have collected this data in the period from June 2012 to January 2013 – for a total of more than six months – in five Swiss households. The characteristics of the households (number of occupants, type of household, etc.) are described in [18] and [4]. We refer to the households as *r1*, *r2*, etc. to preserve their anonymity (we use the same numbering as in [18]).

The samples of electricity consumption have been collected every second using off-the-shelf digital electricity meters installed in the households. One sample represents the average power (in watts) consumed by the household during the second preceding the measurement. We refer to these records as the *aggregate* electricity consumption as they refer to the consumption of the whole household.

Fine-grained occupancy data is available for two periods, referred to as *summer* (July to September 2012) and *winter* (November 2012 to January 2013). Before using this data for our analysis, we perform the same pre-processing steps described in [18] to eliminate erroneous ground truth data. After this data cleaning phase, the number of days for which ground truth data is available for households *r1*, *r2*, *r3*, *r4* and *r5* are 39, 83, 57, 38 and 43 for the summer period; and 46, 45, 21, 48 and 31 for the winter period, respectively.

### SYSTEM DESIGN

In residential buildings, a change in the overall electricity consumption often provides an indication of occupancy as many appliances are used to increase comfort and/or to replace manual labour. Figure 1 shows the electricity consumption of a representative day for household *r2* and the output of a simple thresholding classifier. The latter assumes the household to be occupied whenever the current power is higher than

<sup>2</sup>Ground truth occupancy data has been entered manually by the residents using tablet computers mounted near the main entrance.

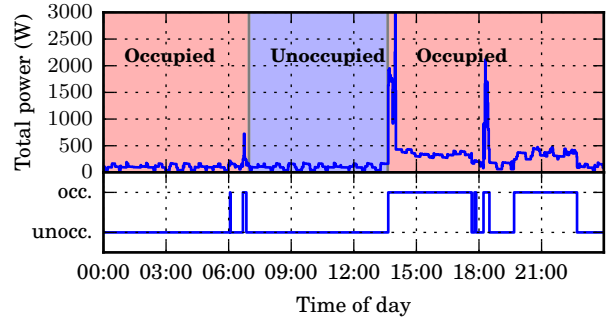


Figure 1: Example of an occupancy detection algorithm based on mean thresholding.

the 24-hour mean. The performance of this classifier shows that even a primitive strategy may detect occupancy. In this paper, we build upon this observation by analysing how electric load curves can be used to monitor occupancy.

### Deriving features from the electric load curve

In order to infer occupancy from the raw electricity consumption data, we identify features of the electric load curve that are indicative of occupancy. A good indication of occupancy are appliance state changes triggered by the interaction of an occupant (*e.g.* an occupant turning on the television or stove) [23]. For this reason, we focus on identifying features that relate to the operation of occupancy-relevant appliances and allow to directly infer occupancy from the aggregate electricity consumption.

In order to identify such features, we compare the day-time<sup>3</sup> (6 a.m. to 10 p.m.) electricity consumption during occupied periods to times when the household is unoccupied.

Table 1 summarises the selected features. All features are computed at 15-minute intervals. Every day is represented as a sequence of  $N_s$  time slots of length  $T_s$ . As the aggregate electricity consumption is sampled at 1 Hz, the interval length of  $N_s = 15$  means that each feature is computed on a 900-element vector (*i.e.*  $T_s = 900$ ). All features listed in Table 1 – apart from  $p_{\text{prob}}$ ,  $p_{\text{fixed}}$  and  $p_{\text{time}}$  – are computed separately for each phase and the sum of all three phases. The subscripts 1, 2 or 3 are used to indicate that a feature has been computed on the data corresponding to phase 1, 2 or 3, respectively. Likewise, the subscript 123 indicates a feature computed on the combined consumption of all three phases. In summary, we consider the features  $\text{min}$ ,  $\text{max}$ ,  $\text{mean}$ ,  $\text{std}$ ,  $\text{sad}$ ,  $\text{cor1}$ ,  $\text{onoff}$ ,  $\text{range}$ ,  $p_{\text{prob}}$ ,  $p_{\text{fixed}}$  and  $p_{\text{time}}$  computed on the three electrical phases. Using these features we aim to capture both the *absolute value* as well as the *variability* of the electricity consumption.

#### Absolute value of the power consumption

The  $\text{min}$ ,  $\text{max}$  and  $\text{mean}$  features denote the minimum, maximum and arithmetic average of each slot.

<sup>3</sup>The ECO data set does not contain ground truth data on sleeping patterns, we thus leave the detection of sleep for future work.

Table 1: Features computed on the aggregate electricity consumption traces.

#	Feature names	Description
f <sub>1</sub> , f <sub>2</sub> , f <sub>3</sub>	min <sub>1</sub> , min <sub>2</sub> , min <sub>3</sub>	Minimum of the samples for phase 1, 2 and 3
f <sub>4</sub>	min <sub>123</sub>	Minimum of the samples for the sum of phase 1, 2 and 3
f <sub>5</sub> , f <sub>6</sub> , f <sub>7</sub>	max <sub>1</sub> , max <sub>2</sub> , max <sub>3</sub>	Maximum of the samples for phase 1, 2 and 3
f <sub>8</sub>	max <sub>123</sub>	Maximum of the samples for the sum of phase 1, 2 and 3
f <sub>9</sub> , f <sub>10</sub> , f <sub>11</sub>	mean <sub>1</sub> , mean <sub>2</sub> , mean <sub>3</sub>	Arithmetic average of the samples for phase 1, 2 and 3
f <sub>12</sub>	mean <sub>123</sub>	Arithmetic average of the samples for the sum of phase 1, 2 and 3
f <sub>13</sub> , f <sub>14</sub> , f <sub>15</sub>	std <sub>1</sub> , std <sub>2</sub> , std <sub>3</sub>	Standard deviation of the samples for phase 1, 2 and 3
f <sub>16</sub>	std <sub>123</sub>	Standard deviation of the samples for the sum of phase 1, 2 and 3
f <sub>17</sub> , f <sub>18</sub> , f <sub>19</sub>	sad <sub>1</sub> , sad <sub>2</sub> , sad <sub>3</sub>	Sum of absolute differences of the samples for phase 1, 2 and 3
f <sub>20</sub>	sad <sub>123</sub>	Sum of absolute differences of the samples for the sum of phase 1, 2 and 3
f <sub>21</sub> , f <sub>22</sub> , f <sub>23</sub>	corl <sub>1</sub> , corl <sub>2</sub> , corl <sub>3</sub>	Autocorrelation at lag 1 computed over the samples for phase 1, 2 and 3
f <sub>24</sub>	corl <sub>123</sub>	Autocorrelation at lag 1 computed over the samples for the sum of phase 1, 2 and 3
f <sub>25</sub> , f <sub>26</sub> , f <sub>27</sub>	onoff <sub>1</sub> , onoff <sub>2</sub> , onoff <sub>3</sub>	Number of detected on/off events for phase 1, 2 and 3
f <sub>28</sub>	onoff <sub>123</sub>	Number of detected on/off events for the sum of phase 1, 2 and 3
f <sub>29</sub> , f <sub>30</sub> , f <sub>31</sub>	range <sub>1</sub> , range <sub>2</sub> , range <sub>3</sub>	Range of the samples for phase 1, 2 and 3
f <sub>32</sub>	range <sub>123</sub>	Range of the samples for the sum of phase 1, 2 and 3
f <sub>33</sub>	p <sub>prob</sub>	Empirical probability of the slot to be occupied
f <sub>34</sub>	p <sub>fixed</sub>	1 (occupied) from 9 a.m. to 5 p.m., 0 (unoccupied) otherwise
f <sub>35</sub>	p <sub>time</sub>	Slot number ( <i>i.e.</i> 1 – 65)

### Variability of the power consumption

A high variability in the electricity consumption may provide an indicator of human activity. Significant changes in the power consumption are often the result of human actions (*e.g.* by operating the stove) or the operation of appliances with varying consumption levels (*e.g.* a television set with LED backlight). We chose the `std` (standard deviation), `sad` (sum of absolute differences), `corl` (autocorrelation at lag one) and `onoff`<sup>4</sup> features as indicators of such variability.

### Temporal dependence of occupancy

As building occupancy is also dependent upon the current time of the day, we use the `pprob`, `pfixed` and `ptime` features to model the temporal aspects of occupancy. `pprob` is the empirical prior probability of a 15-minute slot to be occupied. `pprob` is computed from the ground truth occupancy data. To this end, only data from the training set is used. `pfixed` is a “dummy” prior probability that assumes the household to be always unoccupied between 9 a.m. and 5 p.m. on weekdays and to be always occupied on weekends. `ptime` is the number of the current slot and thus directly introduces a notion of time. Slots are numbered from 1 to 65, with the first slot corresponding to the period between 6 a.m. and 6:15 a.m. and the last one to the time between 10 p.m. and 10:15 p.m.

### Classifiers

In order to use a building’s electricity consumption to monitor occupancy one requires a mapping from the feature space (*e.g.* a mean consumption of 100 W) to occupancy *classes* (*i.e.* `[feature] → {home, away}`). In supervised machine learning, this mapping function – the *classifier* – is inferred from labelled training data. The training data is used to

<sup>4</sup>On/off events occur when an appliance is switched on or off. We detect these events using a simple heuristic: If the difference between a sample and its predecessor is bigger than a threshold  $ThA$  and this difference remains higher than  $ThA$  for at least  $ThT$  seconds, an on/off event is detected. We set  $ThA = 30$  W and  $ThT = 30$  s.

iteratively refine the classifier to maximise the number of examples (*i.e.* `[[feature], class]` tuples) correctly assigned by the classifier. To make sure that the classifier is not overfitting the data (*i.e.* it does capture the underlying relationship between features and classes rather than the noise in the data), we divide the data into *training* and *test* sets. Thereby, the test set provides an unbiased test of the performance of the classifier on previously unseen data. A number of learning algorithms to build classifiers have been suggested in the literature. The learning algorithms used in this paper are support vector machines (SVMs), K-nearest neighbours (KNNs), Gaussian mixture models (GMMs), hidden Markov models (HMMs) and a simple thresholding (THR) approach.

SVMs are widely-used supervised learning models and algorithms that perform linear and non-linear classification. For the implementation of the SVM classifier we employed the LIBSVM library by Chang and Lin [5].

A KNN is a non-parametric model for classification which classifies an example using a majority vote on the classes of its  $k$  most similar neighbours. This means it does not require an explicit learning phase. We used the `ClassificationKNN` classes from the Matlab Statistics Toolbox to implement our KNN classifier. We empirically determined  $k = 1$  and use the Euclidean distance measure to obtain the nearest neighbours.

The limited size of the ECO data set prohibits us from building empirical multivariate probability density functions for a combination of all features. To alleviate this problem, we use GMMs, which allow to approximate these by a weighted sum of individual Gaussian component distributions [28]. The GMMs are built by iteratively refining the parameters of its  $k$  Gaussian component distributions to fit the input data. To avoid overfitting, we chose a suitable  $k$  by minimising the Akaike information criterion (AIC) [3]. The AIC penalises a larger number of components while rewarding goodness of fit. For the implementation we chose the `gmdistribution` class from the Matlab Statistics Tool-

box. The training data is used to create GMMs for both the occupied and unoccupied distribution. The classification of unknown data is performed by maximum-likelihood (*i.e.* comparing the likelihood of the sample belonging to either distribution).

The hitherto presented classifiers are stateless – *i.e.* they do not take into account the previous occupancy state. Occupancy, however, is stateful. In fact, during any particular 15-minute interval a household is most likely to stay in its current state. An occupancy monitoring system should thus focus on detecting occupancy transitions (from occupied to unoccupied and vice versa). To investigate such a stateful occupancy monitoring system, we use a HMM classifier which relates its hidden states (*i.e.* occupied, unoccupied) to emissions (*i.e.* the observed features of the electricity consumption) using matrices of emission and transition probabilities. The transition matrix contains the probability of staying in or moving out of any state for all states. To obtain the matrix of emission probabilities we first construct a 2-dimensional GMM of the first principal component and the  $p_{\text{time}}$  feature (*i.e.* the slot number) for both the occupied and unoccupied states, respectively. The discrete emission probabilities are then obtained by numerally evaluating the integral of the GMMs over a matrix of  $30 \times 16$  bins.

From Figure 1 we conjecture that a high electricity consumption may have a positive correlation with occupancy. To investigate whether a simple classifier may use this correlation, we implemented the THR classifier which computes the mean for each feature vector during all *unoccupied* times. It then uses these means as thresholds to label a feature as occupied. To obtain the final classification, the THR classifier computes a majority vote over all features of a particular interval.

### Dimensionality reduction

The 35 features introduced in Table 1 allow us to capture various characteristics of the electric load curve. However, while some classifiers may utilise the full feature set, others perform best on a subset of these features. Indeed, for each classifier there exists an optimal subset of features that maximises its performance [30]. To find these subsets and to limit the features to the most descriptive ones, we used sequential forward selection (SFS) and principal component analysis (PCA).

#### Feature selection

The optimal set of features may be found by performing a brute-force evaluation of all possible combinations [30]. Alas, the complexity of such an exhaustive search grows exponentially with the number of features. In this paper, we consider the sequential forward selection (SFS) [30] algorithm to heuristically identify reasonable subsets of features.

Listing 1 shows the pseudocode for the sequential forward selection (SFS) algorithm. The first iteration serves to find a single feature  $x$  that maximises a performance metric  $J$ . At each subsequent iteration, SFS considers, in turn, the inclusion of the remaining features. For each iteration  $k$ , the feature maximising  $J$  is included in the feature set  $Y_k$ . This procedure is stopped whenever the remaining features have been exhausted or a (user-specified) number of features  $m$

Listing 1: Sequential Feature Selection (SFS).

```

1  $X = [x_0 \dots x_n]$ ; // Set of all features
2  $Y = \{\emptyset\}$ ; // Best feature set of length  $k$ 
3  $m$ ; // Maximum number of features
4
5 while ( $k \leq |X|$  &&  $k \leq m$ ) {
6 // Inclusion of best feature
7  $x^+ = \arg \max_{x \notin Y_k} [J(Y_k + x)]$ ;
8
9  $Y_k = Y_k + x^+$ ;
10  $k = k + 1$ ;
11 }
12 return  $Y_m$ 

```

has been reached. We do not impose a limit on the number of features (*e.g.*  $m = 35$ ) and use the occupancy detection accuracy (as defined in the next section) as the performance metric  $J$ .

#### Principal component analysis

The features defined in Table 1 contain a certain degree of redundancy. The `max` and `min` features, for example, are closely related to the `range` feature. This redundancy makes choosing the best subset of features using SFS difficult. In fact, combining similar features into a single feature may be more descriptive of the data. Such a transformation is achieved by principal component analysis (PCA). PCA transforms the original features into a set of linear combinations (*i.e.* components). Usually, the first few components often account for most of the variance in the input data. In order to reduce the input of the classifier and to remove redundant features, we thus restrict the number of components to the first  $L$  components that account for at least 95% of the variance of the input data.

### EVALUATION

Figure 2 shows the setup of our classification system using feature selection and principal component analysis, respectively. In both cases, the raw consumption data is first divided into 15-minute slots from which the features are extracted. For each slot, the feature data is combined with its ground truth label to create an *example* for classification. These examples are then divided into test and training sets using cross-validation. During the training phase, the behaviour differs between SFS and PCA. PCA finds the most descriptive features, and then restricts the classifier to use these features only. PCA thus finds a transformation  $W$  of the input data and identifies the first  $L$  components. In contrast to SFS, the testing phase for PCA is performed on the transformed data.

#### Performance measures

If we are only interested in whether a household is occupied or unoccupied, occupancy classification can be regarded as a binary problem. Thus, we refer to instances of correctly classifying an *occupied* household as a *true positive* classification (*tp*). Similarly, we refer to a correct classification of an unoccupied period as a *true negative* classification (*tn*). *False positive* (*fp*) and *false negative* classifications (*fn*) then

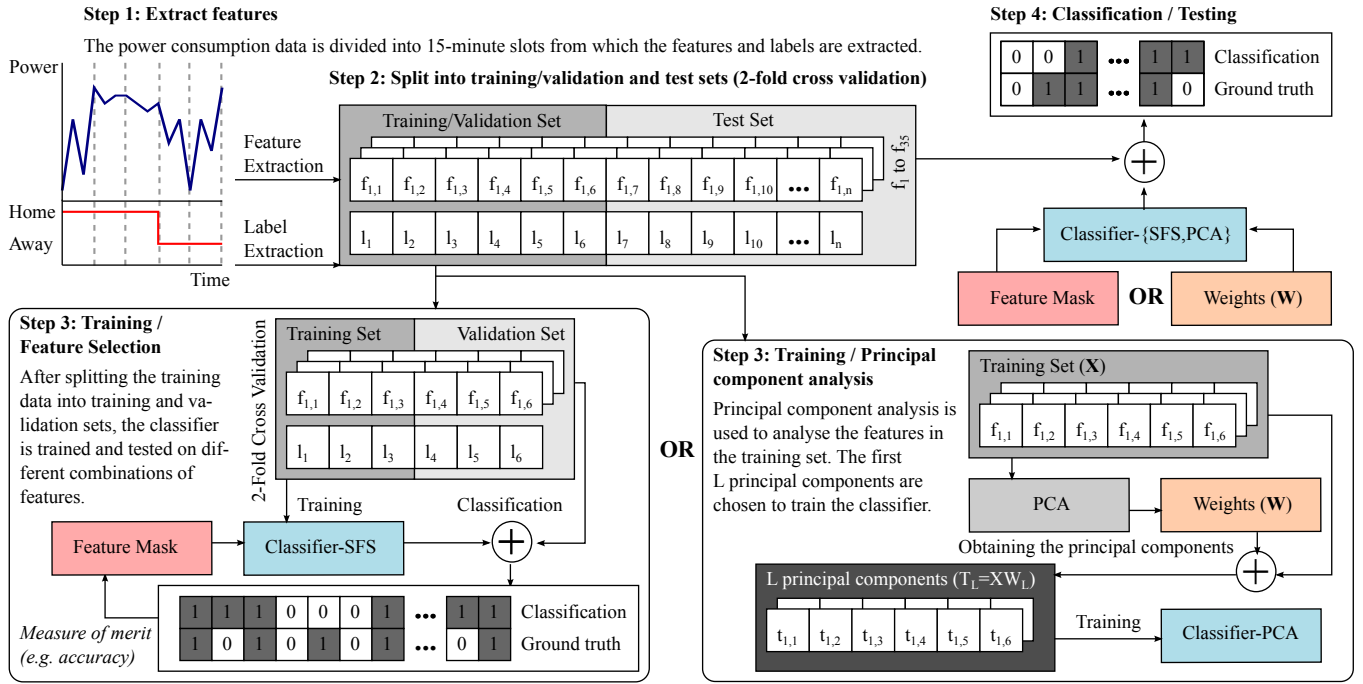


Figure 2: Setup of the evaluation for SFS feature selection and principal component analysis (PCA).

denote incorrect *occupied* or *unoccupied* classifications, respectively. In the following, we use this notation to derive several performance criteria.

#### Classification accuracy

The *classification accuracy* gives a measure of how often the classification is correct. For a classifier  $c$  it is thus computed as the number of correct classifications divided by the total number of classifications:

$$\text{Acc}_c = \frac{tp+tn}{tp+tn+fp+fn}.$$

To obtain a suitable baseline for the classification accuracy, we introduce the maximum-likelihood classifier *Prior* that assigns data points to the class of the majority of data points in the training set. Since the occupancy exceeds 50% in all households<sup>5</sup>, *Prior* is the probability of a household being occupied.

However, as the classification accuracy does not take into account the relative cost of misclassifications, it may only partially describe a classifier's performance. This problem is summarised by Witten *et al.* "[An] evaluation by classification accuracy tacitly assumes equal error costs" [31]. If the objective is to control a smart heating system, for instance, a false negative classification (*i.e.* occupants are wrongly assumed to be away) may erroneously lead to the system lowering the temperature setpoint. On a cold day, this may result in severe thermal discomfort for the occupants.

Furthermore, an unbalanced class distribution (*i.e.* very high or low occupancy) may cause misleading classification accuracies. Households r4 and r5 are occupied over 90% of the

<sup>5</sup>It lies between 63% (r2, winter) and 95% (r4, winter).

time. Thus, for these, even the *Prior* classifier achieves classification accuracies exceeding 90%.

#### Matthews correlation coefficient

A reliable occupancy monitoring system correctly detects both occupied ( $tp$ ) and unoccupied ( $tn$ ) states. To test the performance of our proposed system we thus also computed the Matthews correlation coefficient (MCC) of our classification results [22]. The MCC provides a balanced measure even for heavily skewed input data. A coefficient of +1 represents a perfect prediction. The opposite (*i.e.* a value of -1) is assumed if no single instance was classified correctly. The MCC of a classifier  $c$  is calculated as:

$$\text{MCC}_c = \frac{tp \times tn - fp \times fn}{\sqrt{(tp+fp)(tp+fn)(tn+fp)(tn+fn)}}.$$

#### False negative and false positive rate

False negatives ( $fn$ ) occur when an occupied household is falsely labelled as *unoccupied*. In certain scenarios, such as a heating control application, these errors are particularly grave. They may result in discomfort as the temperature is lowered automatically – even though the occupants are present. We use the false negative rate (FNR) to quantify the number of such misclassifications. The FNR of a classifier  $c$  is defined as the number of false negatives divided by all unoccupied intervals (true and false negatives):

$$\text{FNR}_c = \frac{fn}{fn+tn}.$$

Falsely labelling an unoccupied household as *occupied* results in a *false positive* classification. False positives reduce the efficiency of a heating control system as they trigger it to raise the temperature while the building is unoccupied. The frequency of these errors is denoted by the false positive rate



(FPR). The FPR of a classifier  $c$  is defined as the number of false positives divided by all occupied intervals (true and false positives):

$$\text{FPR}_c = \frac{fp}{fp+tp}.$$

### Cross-validation

We randomly divide the data ten times into different, equi-sized *training* and *testing* sets. For each of these runs, the training set is used to train the classifiers and the testing set is used to evaluate their performance. Afterwards, the roles of the sets are switched and the process repeated. So in total, classification is performed 20 times. The two-fold cross-validation tries to avoid that a specific allocation of data into training and testing sets creates artefacts in the results. The use of ten runs also allows for an assessment of the stability of the feature selection<sup>6</sup> – *i.e.* to analyse if different training data yield different feature sets. The overall performance is computed as the average of the performance over the 20 rounds of classifications.

#### Classification limited to daytime hours

During the night, the electricity consumption is usually less indicative of occupancy. Since we also do not have ground truth data on sleep patterns, we restrict occupancy classification to the time between 6 a.m. and 10.15 p.m.

#### Cross validation at day-granularity

For each classification, the HMM classifier requires the previous occupancy state – *e.g.* at 9.15 a.m. it requires knowledge about the occupancy at 9 a.m. To facilitate this, the input data is assigned to training and testing sets at day-granularity.

## RESULTS

In this section, we use the features derived from the aggregate electricity consumption to quantitatively evaluate the occupancy monitoring performance. We have implemented the SVM, KNN, THR, GMM and HMM classifiers as well as the *Prior* classifier for baseline comparison. The classifiers estimate the occupancy state for each time slot from the set of features computed on the aggregate electricity consumption. To reduce the dimensionality of the feature space, we evaluate the SFS feature selection algorithm and PCA. The used method is indicated by the appendices “-SFS” or “-PCA”, where applicable (*e.g.* SVM-SFS denotes the usage of the SVM classifier trained using SFS feature selection).

First, we discuss the overall occupancy detection performance in terms of accuracy, FNR, FPR and MCC. We then evaluate the performance of the different classifiers and the results of the feature selection. We conclude this section with discussion on the suitability of monitoring building occupancy using electricity consumption data in a smart heating scenario.

### Overall occupancy detection performance

For each household, we use  $C$  to denote the *classifier achieving the highest accuracy* (*i.e.*  $Acc_C$ ). To put  $Acc_C$  into context, we also evaluate the false positive ( $FPR_C$ ) and false

<sup>6</sup>Feature selection is actually performed using an additional two-fold cross-validation on the training data (*cf.* Figure 2).

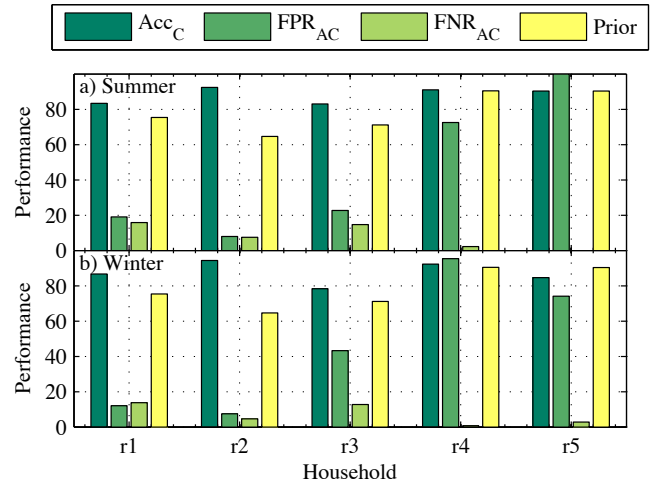


Figure 3: Best accuracy ( $Acc_C$ ) and corresponding performance measures in percent over all classifiers.

negative rates ( $FNR_C$ ) achieved by  $C$ . Figure 3 shows the performance in terms of  $Acc_C$ ,  $FPR_C$  and  $FNR_C$  for all five households in the (a) summer and (b) winter data sets.

In households r1 to r3, the best classifier  $C$  outperforms the *Prior* accuracy as determined by the maximum-likelihood classifier during both summer and winter. The best result is obtained in r2 where  $Acc_C$  is on average 29% higher than the *Prior* accuracy. Here, the mean accuracy of the summer and winter periods is 93%. This results, on average, in approximately one hour misclassified per day. At the same time,  $FNR_C$  is 6% on average. Thus, a low fraction of intervals is misclassified as unoccupied. The average  $FPR_C$  is 8%, which means that  $C$  incorrectly assumes the building to be unoccupied for 38 minutes on average per day.

In households r1 and r3, the best classifiers achieve a ten percent improvement over the *Prior* baseline ( $Acc_C = 85%$  for r1 and  $Acc_C = 81%$  for r3). In contrast to r1, however, the classifications come with a potential of discomfort caused by their false negative rate. In household r1, we observe a  $FNR$  of 15%; meaning that 110 minutes are falsely classified as unoccupied while the participants were actually at home. In household r3, one hour and 35 minutes are misclassified as unoccupied as  $FNR_C = 14%$ .

In households r4 and r5, the accuracy of the best classifier  $C$  does not significantly exceed the performance of the *Prior*. In contrast to r1 to r4, these two households have very high (*i.e.* around 90%) occupancy levels. The reasons for the inability of the classifiers to outperform the *Prior* classifier may not be conclusively established as we lack detailed data on the behaviour of the occupants. We assume, however, that the results can in part be explained by the behaviour of the occupants. The high occupancy in r4 and r5 means that there may be periods where a building is occupied while no electrical appliances are in operation. For the classifier, these periods look identical to those encountered when the building is actually unoccupied. Furthermore, as the buildings are almost always occupied, the number of such *inactive* periods is likely

Table 2: **Classification accuracy** (expressed as percentages) for each household and algorithm in summer and winter.

#	SFS			PCA				Prior
	SVM	KNN	THR	SVM	KNN	GMM	HMM	
<i>Summer</i>								
r1	80	76	77	<b>83</b>	80	78	<b>83</b>	75
r2	91	88	76	<b>92</b>	89	76	90	65
r3	78	76	71	<b>83</b>	79	70	82	71
r4	90	90	85	<b>91</b>	88	70	87	90
r5	90	88	81	90	84	59	79	90
<i>Winter</i>								
r1	82	78	83	84	81	79	<b>87</b>	73
r2	93	91	77	<b>94</b>	91	88	92	63
r3	70	71	66	<b>78</b>	76	59	71	71
r4	92	92	90	92	90	70	84	93
r5	82	80	77	<b>85</b>	79	63	74	82

to exceed the occupied periods. This would result in the classifier almost always classifying the home as occupied.

This problem is well-known in machine learning and may be alleviated partially by undersampling the training data to obtain an even split of occupied and unoccupied intervals [13, 14]. The downside of this approach is that it may increase the number of intervals misclassified as unoccupied, which implies a higher false positive rate. We believe, however, that the main objective of a smart heating system is to ensure comfort at all times. Therefore, such an approach is not feasible. After all, very high occupancy households do not represent viable targets for a smart heating system in the first place. The high occupancy prevents the system to lower the temperature over significant periods of time, resulting in little energy savings<sup>7</sup>. For the remainder of this paper, we list the results for households r4 and r5 for completeness' sake, but refrain from including them in our analysis due to their limited suitability to a smart heating scenario.

### Performance by classifier

In the previous section we have discussed the best performance over all classifiers. In this section, we analyse how the classifiers perform relative to each other. To this end, Table 2 shows the classification accuracy for all combinations of classifiers and households for both the summer and winter data sets. For each household, the best classifier(s) are indicated in bold print. The table shows that, overall, the SVM-PCA classifier is the best classifier in terms of classification accuracy, outperforming the other classifiers in seven out of ten cases. It achieves an average accuracy of 86% for households r1 to r3. The main reason for this is that it adopts well to the non-linearity of the feature space. The HMM classifier performs best for household r1 in winter and performs equally well as the SVM-PCA classifier in summer. Its average accuracy for households r1 to r3 is 84%. The worst performing classifier is the simple THR-SFS classifier. It only achieves an average accuracy of 75%, outperforming the Prior by 5% only.

The results in Table 2 show that the use of PCA to reduce the dimensionality of the features outperforms feature selection with the SFS algorithm. While SVM-SFS comes close to the

<sup>7</sup>The relationship between energy savings and occupancy is further explored in [19].

Table 3: **Matthews correlation coefficient** for each household and algorithm in summer and winter.

#	SFS			PCA			
	SVM	KNN	THR	SVM	KNN	GMM	HMM
<i>Summer</i>							
r1	0.40	0.35	0.35	0.52	0.46	0.49	<b>0.60</b>
r2	0.81	0.73	0.45	<b>0.84</b>	0.76	0.55	0.79
r3	0.46	0.42	0.32	<b>0.61</b>	0.49	0.44	<b>0.61</b>
r4	0.14	0.15	0.19	0.35	0.35	0.32	<b>0.45</b>
r5	/	0	0.05	/	0.11	0.13	<b>0.19</b>
<i>Winter</i>							
r1	0.50	0.42	0.55	0.58	0.53	0.55	<b>0.70</b>
r2	0.84	0.81	0.51	<b>0.88</b>	0.82	0.75	0.84
r3	0.18	0.21	0.14	<b>0.46</b>	0.41	0.20	0.32
r4	0.10	0.09	0.19	0.15	0.20	0.22	<b>0.26</b>
r5	0.11	0.24	0.07	<b>0.35</b>	0.32	0.25	0.31

Table 4: **False negative rate** (expressed as percentages) for each household and algorithm in summer and winter.

#	SFS			PCA			
	SVM	KNN	THR	SVM	KNN	GMM	HMM
<i>Summer</i>							
r1	<b>9</b>	15	14	10	14	22	16
r2	8	10	11	<b>7</b>	9	30	9
r3	16	17	21	<b>15</b>	16	36	20
r4	<b>1</b>	2	9	2	7	31	11
r5	<b>0</b>	2	12	<b>0</b>	9	42	17
<i>Winter</i>							
r1	<b>8</b>	12	<b>8</b>	9	13	23	14
r2	6	7	15	<b>5</b>	7	15	9
r3	<b>13</b>	<b>13</b>	21	<b>13</b>	16	45	24
r4	<b>1</b>	<b>1</b>	5	<b>1</b>	4	30	14
r5	<b>2</b>	11	9	3	14	39	23

results of SVM-PCA in household r2, SFS is outperformed by PCA in all 5 households. We further analyse the features selected by SFS in the next section and discuss possible explanations for these results.

The results are similar if the MCC is used as a performance measure. Table 3 shows again that PCA outperforms SFS feature selection in all 5 households. Among the classifiers using PCA, we see that the performance of the HMM classifier approaches that of the SVM-PCA classifier. Both classifiers have an average MCC of 0.64 for households r1 to r3. While HMM achieves the highest MCC for r1, SVM-PCA performs similarly or better in households r2 and r3. The performance gap between the HMM and SVM-PCA classifiers in r1 can be explained by a more even split between false positives and false negatives which is rewarded by the MCC.

False negatives (*i.e.* a building falsely declared unoccupied) are important due to their impact on the occupant's thermal comfort. Table 4 shows the FNR for all combinations of classifiers and households. As in the previous tables, bold print indicates the best (*i.e.* lowest) values. The table shows that it may not be advisable to choose the classifier solely based on the classification accuracy (or MCC). We previously noted that  $Acc_C$  was 85% for household r1. This result was achieved by the HMM classifier<sup>8</sup>. However, by choosing the HMM classifier we incur an average FPR of 15%. If we use

<sup>8</sup>The accuracy of HMM classifier slightly exceeds that of the SVM-PCA classifier. This is not visible in Table 2 due to rounding errors.



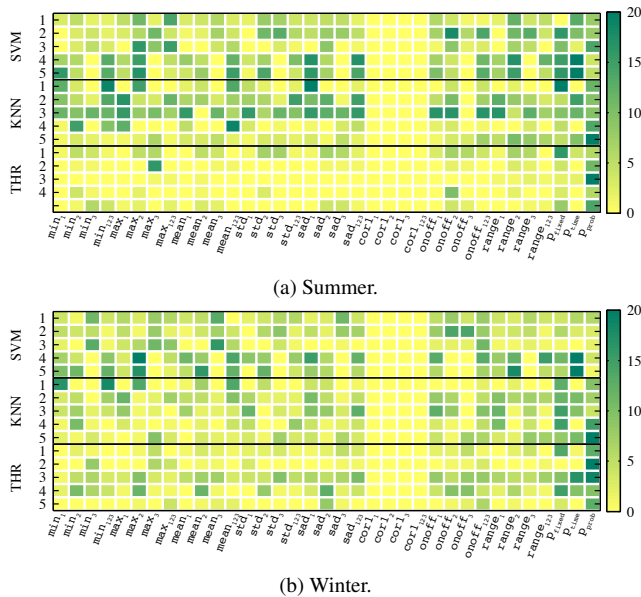


Figure 4: Number of times a specific feature has been chosen as part of the feature subset selected by the SFS algorithm for a particular household and classifier. A darker colour indicates a feature was chosen more frequently.

the SVM-PCA classifier instead, the FPR can be reduced to 10% at the expense of only a 1% reduction in accuracy.

### Features best describing occupancy

In this section we analyse the features chosen by the SFS algorithm. Figures 4a and 4b show – for the summer and winter periods, respectively – the number of times a particular feature has been chosen by SFS. The rows show for each of the 35 features listed on the x-axis, the number of times it has been chosen for a particular household and classifier.

Figures 4a and 4b show that in successive runs of the SFS feature selection, different features are being chosen for the same households. Furthermore, no feature is chosen consistently over all households. A possible explanation for this behaviour is that there is a high correlation between individual features. The `range` feature, for example, is computed from the absolute difference between the `min` and `max` features. Likewise, the `sad` and `onoff` are closely related. While `sad` computes the sum over all deltas of the electricity consumption, the `onoff` feature counts the number of occurrences of a specific delta. Due to this similarity, small variations in the classification accuracy resulting from the variance in the dataset rather than the descriptiveness of a particular feature cause different features to be selected. Incidentally, this may also explain the good performance of PCA. Through the combination of similar features and the selection of the first components, redundant information is ignored.

The `min`, `max`, `mean`, `std`, `sad`, `cor1`, `onoff` and `range` features are computed on the three phases as well as the sum of all phases resulting in 32 features overall (cf. Table 1). Figures 4a and 4b show the number of occurrences of these fea-

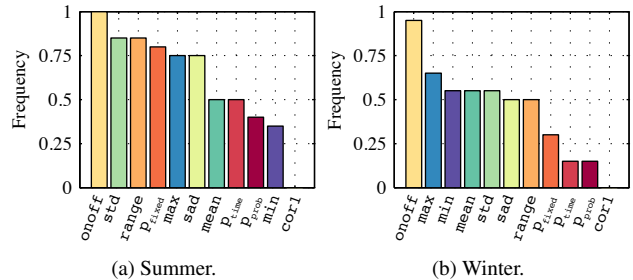


Figure 5: Combined features chosen by SFS (r2, SVM).

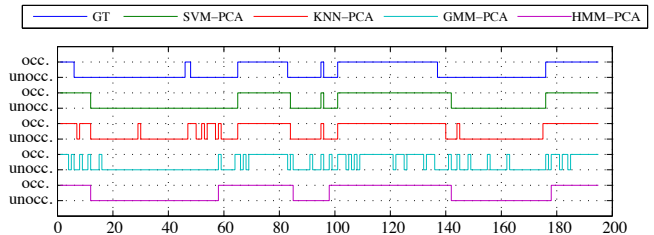


Figure 6: Ground truth (GT) and occupancy transitions for an exemplary classification of 195 intervals (3 days) of r2.

tures for each phase. To analyse a feature’s descriptiveness irrespective of the phase it was computed on, Figure 5 shows the cumulative probability of each particular feature for the summer (Figure 5a) and winter (Figure 5b) data sets. Overall, the `onoff` feature is chosen most often by the SFS feature selection. For the summer data set it is used in all runs. In the winter it is used in over 90% of the runs. The next most frequent features differ between the summer and winter data sets. While in winter, the `max` and `min` features follow the `onoff` feature, in summer the `std` and `range` come in second and third place.

From Figures 4a, 4b and 5 we can see that in summer, more features are chosen than during the winter. Figure 5 shows that in summer, the first six features are chosen in more than 75% of the runs. During the winter only the first feature exceeds a 75% probability of being chosen. The two data sets also differ in the frequency of the time features (*i.e.* `pprob`, `pfixed` and `ptime`). During the winter, the frequency of the time features approximately halves compared to the summer. A possible explanation is that these add information about the correlation between occupancy and the current time of day. During the summer in Switzerland, the sun rises before 6 a.m. and sets after 9 p.m., resulting in less energy spent on lighting. Thus when the electricity consumption alone (*e.g.* in the morning) is not sufficient for monitoring occupancy, the time features may provide a fallback.

### SUITABILITY FOR CONTROLLING A THERMOSTAT

Before we conclude the paper, we discuss the suitability and limitations of using electricity meters to monitor occupancy for a smart heating application.

Thus far, we have analysed the performance of the classifiers for individual intervals. Thereby we have treated each

Table 5: **RMSE** between the number of actual occupancy transitions per day and the predicted transitions; and **ADOT** for each household and algorithm in summer and winter.

#	SFS			PCA				ADOT
	SVM	KNN	THR	SVM	KNN	GMM	HMM	
<i>Summer</i>								
r1	11.7	9.3	11.5	7.4	<b>6.2</b>	9.2	8.7	2
r2	3.6	3.9	12.4	<b>3.4</b>	3.8	6.3	3.7	2.5
r3	10.1	7.2	9.9	7.8	<b>6.0</b>	11.1	10.4	2.3
r4	9.2	8.7	11.2	6.5	<b>5.6</b>	20.1	9.1	1.8
r5	12.1	11.1	9.0	12.1	<b>7.4</b>	22.9	11.7	1.3
<i>Winter</i>								
r1	8.4	8.5	8.5	<b>7.9</b>	9.8	14.0	9.1	1.1
r2	2.6	2.9	6.3	<b>2.5</b>	2.8	3.9	3.0	2.2
r3	5.3	5.9	5.6	4.0	5.2	8.2	<b>3.1</b>	1.9
r4	9.3	9.1	7.1	8.4	<b>5.7</b>	26.9	15.2	1.3
r5	13.8	8.1	9.1	10.3	<b>5.3</b>	12.3	6.9	2.1

interval independently. For the chosen metrics (*i.e.* classification accuracy, MCC, FPR and FNR), each correct or incorrect classification thus contributes with the same weight. However, when the controller is notified that the building has become occupied, it starts to heat to reach the comfort temperature. The ability to correctly detect *occupancy transitions* – *i.e.* changes in the occupancy state from occupied to unoccupied and vice versa – is crucial to the system. As each occupancy transition causes the controller to adapt its heating strategy, it is important not to over or under-estimate the number of transitions. Figure 6 shows the occupancy transitions for the first 195 slots of household r2 for the SVM-PCA, KNN-PCA GMM-PCA and HMM-PCA classifiers. The ground truth data contains 10 state transitions (six occupied periods and five unoccupied periods). The SVM-PCA classifier reproduces the occupancy transitions of the ground truth data most closely. Apart from missing a short period of occupancy on the first day, it shows the same number of transitions as the ground truth data. Owing to its stateful nature, the HMM-PCA classifier misses two short occupancy periods but otherwise follows the ground truth occupancy transitions. The KNN-PCA and GMM-PCA classifiers significantly overestimate the number of occupancy transitions, rendering them unsuitable for a smart heating controller.

To formalise this problem, we analyse the root mean square error (RMSE) between the number of actual occupancy transitions per day and the transitions predicted by the classifiers. In addition, we compute the average number of daily occupancy transitions (ADOT). We define the ADOT of a classifier  $c$  as:

$$\text{ADOT}_c = \frac{\sum_{d=1}^{\text{Total number of days}} \text{Number of transitions for day } d}{\text{Total number of days}}.$$

Table 5 shows that household r2 has the highest average number of daily occupancy transitions (ADOT) in the data set. Its occupancy changes 2.4 times per day on average. For household r2, the SVM-PCA classifier most closely predicts the true number of occupancy transitions with an average error of 3 transitions. In the other households all classifiers significantly overestimate the number of transitions. Thus, additional smoothing in the controller is required to avoid unnecessary switches. A possible remedy could be to wait a pre-defined period before declaring the building unoccupied.

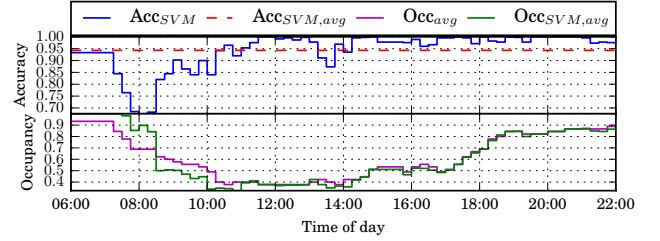


Figure 7: Mean accuracy over 24 hours (r2, SVM, winter).

### Limits to classification

Our results show that, for some households, the SVM-PCA classifier’s performance may warrant its inclusion into a smart heating system. However, the classification accuracy shows that further improvements are possible. Figure 7 shows the average classification accuracy of the SVM-PCA classifier from 6 a.m. to 10 p.m. The upper graph shows that, while for most of the day the accuracy (blue, solid line) stays close to or above the average accuracy (red, dotted line), there is a significant drop in the morning. The lower graph depicts these misclassifications in more detail. Up to 8.30 a.m., the SVM-PCA classifier overestimates occupancy. A potential explanation is that the participants are more likely to forgo a hot breakfast the earlier they leave the building. After 8.30 a.m., the situation reverses and the occupancy is underestimated. This could be due to the occupants sleeping in on weekends and a low utilisation of electrical appliances in the morning hours.

While their performance may not be suitable for real-time occupancy monitoring in all households, digital electricity meters can give a good overall indication of a building’s level of occupancy. As these meters are increasingly deployed, their ubiquity may be used to identify the households which would benefit most from a smart heating system.

### CONCLUSION

In this paper we addressed the problem of performing automatic home occupancy detection using aggregate electricity consumption data. Our results improve upon our previous, preliminary work [18] and show that the use of smart electricity meters allows to achieve an average occupancy detection accuracy of up to 94%. We further showed that due to the varying setup in different households, no single feature set performs consistently well over all households. In terms of individual features, however, a feature that captures changes in the activation state of appliances (*e.g.* like the `onoff` feature in our case) should be used. To increase the occupancy monitoring performance, future work should look at fusing electricity consumption data with other sensory data.

### ACKNOWLEDGMENTS

We would like to thank the participating households for their collaboration. We further thank Friedemann Mattern and Thorsten Staake for their support. Finally, we want to express our gratitude to our anonymous shepherd and reviewers for their valuable comments.

## REFERENCES

1. Agarwal, Y., Balaji, B., Dutta, S., Gupta, R., and Weng, T. Duty-cycling buildings aggressively: The next frontier in HVAC control. In *Proceedings of the 10th International Conference on Information Processing in Sensor Networks (IPSN '11)*, IEEE (Chicago, IL, USA, April 2011), 246–257.
2. Agarwal, Y., Balaji, B., Gupta, R., Lyles, J., Wei, M., and Weng, T. Occupancy-driven energy management for smart building automation. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys '10)*, ACM (Zurich, Switzerland, September 2010), 1–6.
3. Akaike, H. Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*, E. Parzen, K. Tanabe, and G. Kitagawa, Eds., Springer Series in Statistics. Springer, 1998, 199–213.
4. Beckel, C., Kleiminger, W., Cicchetti, R., Staake, T., and Santini, S. The ECO data set and the performance of non-intrusive load monitoring algorithms. In *Proceedings of the 1st ACM International Conference on Embedded Systems for Energy-Efficient Buildings (BuildSys '14)*, ACM (Memphis, TN, USA, November 2014), 80–89.
5. Chang, C.-C., and Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2 (April 2011), 27:1–27:27.
6. Chen, D., Barker, S., Subbaswamy, A., Irwin, D., and Shenoy, P. Non-intrusive occupancy monitoring using smart meters. In *Proceedings of the 5th ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys '13)*, ACM (Rome, Italy, November 2013), 9:1–9:8.
7. Dodier, R. H., Henze, G. P., Tiller, D. K., and Guo, X. Building occupancy detection through sensor belief networks. *Energy and Buildings* 38, 9 (September 2006), 1033–1043.
8. Erickson, V. L., Achleitner, S., and Cerpa, A. E. POEM: Power-efficient occupancy-based energy management system. In *Proceedings of the 12th International Conference on Information Processing in Sensor Networks (IPSN '13)*, ACM/IEEE (Berlin, Germany, April 2013), 203–216.
9. Froehlich, J., Larson, E., Gupta, S., Cohn, G., Reynolds, M., and Patel, S. Disaggregated end-use energy sensing for the smart grid. *IEEE Pervasive Computing* 10, 1 (Januar 2011), 28–39.
10. Gupta, M., Intille, S. S., and Larson, K. Adding GPS-control to traditional thermostats: An exploration of potential energy savings and design challenges. In *Proceedings of the 7th International Conference on Pervasive Computing (Pervasive '09)*, Springer (Nara, Japan, May 2009), 1–18.
11. Gupta, S., Reynolds, M. S., and Patel, S. N. ElectriSense: Single-point sensing using EMI for electrical event detection and classification in the home. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing (UbiComp '10)*, ACM (Copenhagen, Denmark, September 2010), 139–148.
12. Hart, G. W. Nonintrusive appliance load monitoring. *Proceedings of the IEEE* 80, 12 (December 1992), 1870–1891.
13. He, H., and Garcia, E. A. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering* 21, 9 (September 2009), 1263–1284.
14. Japkowicz, N. The class imbalance problem: Significance and strategies. In *International Conference on Artificial Intelligence (ICAI)*, CSREA Press (2000), 111–117.
15. Jiang, X., Dawson-Haggerty, S., Dutta, P., and Culler, D. E. Design and implementation of a high-fidelity AC metering network. In *Proceedings of the 8th International Conference on Information Processing in Sensor Networks (IPSN '09)*, ACM (San Francisco, CA, USA, April 2009).
16. Jin, M., Jia, R., Kang, Z., Konstantakopoulos, I. C., and Spanos, C. J. Presencesense: Zero-training algorithm for individual presence detection based on power monitoring. *CoRR abs/1407.4395* (July 2014).
17. Kleiminger, W., Beckel, C., Dey, A., and Santini, S. Poster abstract: Using unlabeled Wi-Fi scan data to discover occupancy patterns of private households. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys '13)*, ACM (Rome, Italy, November 2013).
18. Kleiminger, W., Beckel, C., Staake, T., and Santini, S. Occupancy detection from electricity consumption data. In *Proceedings of the 5th ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys '13)*, ACM (Rome, Italy, November 2013), 1–8.
19. Kleiminger, W., Mattern, F., and Santini, S. Predicting household occupancy for smart heating control: A comparative performance analysis of state-of-the-art approaches. *Energy and Buildings* 85, 0 (December 2014), 493–505.
20. Krumm, J., and Brush, A. J. B. Learning time-based presence probabilities. In *Proceedings of the 9th International Conference on Pervasive Computing (Pervasive '11)*, IEEE (San Francisco, CA, USA, June 2011), 79–96.
21. Lu, J., Sookoor, T., Srinivasan, V., Gao, G., Holben, B., Stankovic, J., Field, E., and Whitehouse, K. The Smart Thermostat: Using occupancy sensors to save energy in homes. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys '10)*, ACM (Zurich, Switzerland, November 2010), 211–224.

22. Matthews, B. W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure* 405, 2 (October 1975), 442–451.
23. Molina-Markham, A., Shenoy, P., Fu, K., Cecchet, E., and Irwin, D. Private memoirs of a smart meter. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys '10)*, ACM (Zurich, Switzerland, September 2010), 61–66.
24. Nguyen, T. A., and Aiello, M. Energy intelligent buildings based on user activity: A survey. *Energy and Buildings* 56 (January 2013), 244–257.
25. Padmanabh, K., Malakarjuna, V. A., Sen, S., Katru, S. P., Kumar, A., Pawankumar, S. C., Vuppala, S. K., and Paul, S. iSense: A wireless sensor network based conference room management system. In *Proceedings of the 1st ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings (BuildSys '09)*, ACM (Berkeley, CA, USA, November 2009).
26. Parliament, E., and Council. Directive 2009/72/EC of the European Parliament and of the Council of 13 July 2009 concerning common rules for the internal market in electricity and repealing Directive 2003/54/EC, July 2009.
27. Patel, S., Robertson, T., Kientz, J., Reynolds, M., and Abowd, G. At the flick of a switch: Detecting and classifying unique electrical events on the residential power line. In *Proceedings of the 9th ACM International Conference on Ubiquitous Computing (UbiComp '07)*, Springer (Innsbruck, Austria, September 2007), 271–288.
28. Reynolds, D. Gaussian mixture models. *Encyclopedia of Biometrics* (2009), 659–663.
29. Scott, J., Brush, A. J. B., Krumm, J., Meyers, B., Hazas, M., Hodges, S., and Villar, N. Preheat: Controlling home heating using occupancy prediction. In *Proceedings of the 13th ACM International Conference on Ubiquitous Computing (UbiComp '11)*, ACM (Beijing, PRC, September 2011), 281–290.
30. Whitney, A. W. A direct method of nonparametric measurement selection. *IEEE Transactions on Computers* 100, 9 (September 1971), 1100–1103.
31. Witten, I. H., Frank, E., and Hall, M. A. *Data Mining: Practical Machine Learning Tools and Techniques, Third Edition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2011.
32. Yang, R., and Newman, M. W. Learning from a learning thermostat: Lessons for intelligent systems for the home. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*, ACM (Zurich, Switzerland, September 2013), 93–102.