User Modeling and User-Adapted Interaction manuscript No.
(will be inserted by the editor)

Non-parametric User Activity Modelling and Prediction

Yannick De Bock · Andres Auquilla · Ann Nowé · Joost R. Duflou

Received: date / Accepted: date

Abstract Modelling the occupancy of buildings, rooms or the usage of machines has many applications in varying fields, exemplified by the fairly recent emergence of smart, self-learning thermostats. Typically, the aim of such systems is to provide insight into user behaviour and incentivise energy savings or to automatically reduce consumption while maintaining user comfort. This paper presents a non-parametric user activity modelling algorithm, i.e. a Dirichlet process mixture (DPM) model, implemented by Gibbs sampling and the stick-breaking process, to infer the underlying patterns in user behaviour from the data. The technique deals with multiple activities, such as <*present, absent, sleeping*>, of multiple users. Furthermore, it can also be used for modelling and predicting appliance usage (e.g. <*on, standby, off*>). The algorithm is evaluated, both on cluster validity and predictive performance, using three case studies of varying complexity. The obtained results indicate that the method is able to properly assign the activity data into well-defined clusters. Moreover, the high prediction accuracy demonstrates that these clusters can be exploited to anticipate future behaviour, facilitating the development of intelligent building management systems.

Keywords Clustering · Dirichlet Process Mixture · Occupancy prediction · Activity recognition

Y. De Bock, J. Duflou
Department of Mechanical Engineering KU Leuven - Flanders Make
Celestijnenlaan 300, 3001 Leuven, Belgium
Tel.: +32-16-377938
E-mail: yannick.debock@kuleuven.be

A. Auquilla
Department of Mechanical Engineering KU Leuven - Flanders Make, Leuven, Belgium
Department of Computer Science, University of Cuenca, Cuenca, Ecuador

A. Nowé
Computational Modeling Lab, Vrije Universiteit Brussel
Pleinlaan 2, Brussels, Belgium

1 Introduction

Nowadays, everything is getting smart, from smart phones to smart thermostats, to smart buildings and even smart cities. This evolution is a direct result of the rise of the Internet of Things (IoT), which, in combination with a growing environmental awareness, has led to the identification of the potential of ICT for improving the energy efficiency of, for example, appliances or buildings. The Global e-Sustainability Initiative (GeSI) more generally noted in their #SMARTer2030 report [23] that ICT could enable a 20% reduction of global CO_2 emissions by 2030 and substantial other environmental benefits, such as increasing agricultural crop yields by 30%, saving over 300 trillion litres of water and 25 billion barrels of oil per year.

Introducing intelligence into devices, by modelling their usage, creating so-called intelligent products, is one avenue to achieve energy savings. Examples are found in a variety of application areas such as smart heating systems [2, 12, 17, 24, 33, 36, 38, 39, 41, 43, 53, 64], which already found their way to the market (e.g. Nest, Heat Genius, Anna), appliance usage forecasting [3, 4, 8, 20, 29, 51, 60, 67], dynamic power management [35, 45, 46, 55], smart charging of electric vehicles [28,37] and fault detection [7,40,49,68]. Ventä [66] defines intelligence as products and systems that 1) continuously monitor their status and environment, 2) react and adapt to environmental and operational conditions, 3) maintain optimal performance in variable circumstances, also in exceptional cases and 4) actively communicate with the user, environment or with other products and systems. Next to energy savings, these autonomous systems also aim to improve user comfort through a range of smart features and increase convenience by alleviating user interactions. For example, about half of the programmable thermostats in U.S. households are improperly used due to complex user interfaces, lengthy manuals, and social and practical barriers [47]. As a result, the heating schedule is never updated, leading to outdated schemes and energy wastage. In contrast with this suboptimal use, smart thermostats autonomously generate and dynamically adjust tailored heating schedules. Hereto, they extract patterns from historical occupancy data, representing typical behaviour, and, together with user preferences, form user models. For example, in the case of a smart heating system, the patterns specify occupancy schemes, such as workdays, weekends and more complex schedules. The preferences are then defined by the user's desired temperature set points. Once a model, representing the preferences and behaviour of a specific (group of) user(s), is built, it can be exploited to anticipate future demand. For the smart heating example, such an adaptive control system would automatically switch to a setback temperature in case of absence and ensure that the comfort temperature is again reached upon arrival. Hence, wasteful behaviour is avoided, as unoccupied spaces are not heated, while user comfort is satisfied.

This paper proposes a non-parametric user activity modelling method which automatically identifies the number of distinct activity patterns. These patterns provide insights to the users on their consumption and help identify extra saving opportunities. New incoming data then enable the system to

estimate the current pattern and adapt control accordingly, in order to minimise consumption. In addition, the detected pattern indicates the expected future demand (e.g. the occupancy schedule for the current day), providing valuable information for an intuitive user interface. The effectiveness of the proposed modelling and prediction strategy is evaluated through empirical evaluation on real-world data. The remainder of this paper is organised as follows. Section 2 provides a discussion on related work. In Section 3 the modelling algorithm is described. Furthermore, it presents how future demand is inferred from the user model, i.e. the prediction method. The performance of the modelling framework is illustrated through several case studies in Section 4 and discussed in Section 5. Finally, the contribution is summarised and future work is identified in Section 6.

2 Methods and applications

As in modern society automation has become omnipresent and resource conservation is increasingly more important, a vast amount of research regarding energy-efficient intelligent control systems has been conducted. These systems aim to minimise consumption while maintaining functionality and meeting user requirements. Hereto, patterns are mined from historical behaviour, temporal and possibly environmental data in order to, based on new incoming data, anticipate future needs by adjusting control accordingly. Basically, two approaches can be identified: an unsupervised approach, which infers the natural structure present in the data; and a supervised approach, which learns a function that, given a sample of data and corresponding outputs, best approximates the relationship between input and output observable in the data. The former explicitly extracts profiles by grouping similar patterns together and uses these profiles for prediction. The latter directly builds a prediction model from historical data (and implicitly models the behaviour). The unsupervised approach, which is the focus of this study, has the advantage that it provides more insight into the system's behaviour as the identified profiles could be incorporated into the user interface, leading to a better understanding of the system. For example, a smart heating system could use the profiles to define heating schedules and show the expected occupancy profile of the current day, informing the user what it has learned and why certain actions, e.g. lowering the temperature when absence is expected, are taken. Supervised methods do not generate a set of profiles and their implicit models, e.g. (complex) decision trees or the decision boundary of a support vector machine, are not that easily interpretable. Furthermore, long-horizon (e.g. 24h) predictions are straightforward with the unsupervised approach as it suffices to identify the current profile. A wide variety of applications can benefit from intelligent control systems, like the one presented in this paper, which can be illustrated by the framework developed in [11] that identifies the reduction potential of any application by intelligent control.

A popular and intensively studied application area is smart heating. Heating, ventilation and air conditioning (HVAC) is, due to the system's inertia, particularly well suited for automated, anticipative control. Furthermore, occupancy-driven HVAC control has received substantial attention given that it accounts for a large fraction of total building energy consumption (over 40% for both residential and commercial buildings in the US [15, 16], and 68% for residential space heating in the EU [18]), making it an interesting application for energy conservation. Such intelligent heating control comprises occupancy detection and prediction. Nguyen [44] and Kleiminger [31] present an overview of the most common technologies for occupancy detection and prediction. A wireless network of simple sensors, which is cheap, non-intrusive and easily retrofitted, is the most popular approach for occupancy detection [44]. For example, in [30] a combination of passive infrared (PIR) and door sensors is used. Camera networks, though expensive and considered intrusive by the user [44], are also used to determine occupancy. Other approaches reported in literature are RFID badges, entry-exit logs, proximity sensors and ultrasonic localization [44]. Furthermore, GPS or smart phone data and a combination of thermal and PIR sensors are respectively used in [24, 33], [50] and [5]. Modelling user occupancy facilitates energy conservation by automatically turning the heating off when vacancy is expected. Additionally, learning the building's inertia [62] allows to derive the latest possible time the heating should be turned on in order to reach the desired temperature upon the user's expected arrival. The unsupervised approach was followed by Vázquez and Kastner (2011), who evaluated several traditional clustering methods with their self-developed pattern and method credibility index [65]. Fuzzy C-means and Exclusive Self-Organizing Maps were identified as the best methods. For prediction, the most significant profile, i.e. the largest cluster, is selected by default. In case of notable changes, the algorithm switches to another, more appropriate profile. A different approach is taken in [2], where in an offline-mode cross-correlations between day patterns are determined to identify the occupancy profiles. Then, the repetition period of a room's profile sequence, i.e. the period with which profiles recur, is determined using autocorrelation, and the probability of the next profile is computed based on the profiles of the past days. Finally, in real-time mode, the cross-correlations between the stored profiles and the real-time detected ones are computed dynamically throughout the day and the most similar profile is selected. A non-parametric clustering method, Dirichlet process mixture, is developed in [58] and extended with a weekday variable in [12]. Prediction is then facilitated by selecting the most appropriate profile based on a profile's prior probability, pattern and weekday probability. In [33] the problem is formulated as a linear matrix problem and solved with least squares. The matrix consists of one row for each week and N columns where N equals the number of time slots in a week (336 for 30-minute time slots). On each row the value 1 indicates at which time on which weekday the user was present. Nacer, lastly, uses a combination of k-means and expectation-maximisation, to model user occupancy, and a probabilistic prediction method [43]. Well-known supervised approaches use neural networks

(the neurothermostat [41]), hidden markov models (the smart thermostat [38]), markov chains (OBSERVE [17]), k-nearest neighbours (PreHeat [53]) or logistic regression [39]. Alternatively, environmental information can be exploited, such as GPS information in the drive time algorithm of Gupta's GPS thermostat [24]. Such smart heating solutions can be further extended by incorporating more activities besides presence and absence, such as sleeping or leisure. The ability to predict these activities allows the system to provide more comfort by tailoring heating control towards the user's temperature preferences for the given activities. Human activity can be inferred from various information sources, e.g. the energy consumption of appliances [32], sensor networks [57, 63] and wearables (e.g. smart phones) [34, 50].

Although the scope of this paper is limited to modelling and predicting user activity, methods for occupancy prediction, as exemplified by the works of Tominaga [58] and Truong [60], can be adopted for forecasting the usage of appliances, which is another profoundly researched application area. Dynamic power management and load shifting, for example, are identified as energy conservation opportunities. The former entails adjusting an appliance's power mode in accordance with anticipated demand, e.g. completely power off the machine at night (instead of switching to a standby mode). The latter, on the other hand, is concerned with scheduling usage to minimise peak demand and as such avoid the need for peaking power plants and accompanying emissions. Both, however, rely on the capability of predicting usage. Barbato proposes a three-stage prediction strategy [3], consisting of status, time and duration prediction. Dirichlet process mixture was used to model and predict usage of appliances in [60]. Additionally, this approach also models the interdependency between multiple appliances. Lee et al. developed a usage and occupancy profiling strategy for automatic standby power management in [35]. First, an appliance's power modes are modelled and it is classified in one of three groups (background, occupancy-reactive and user-interactive). Then, future usage is predicted by the conditional probability of usage given the time of day, the day of the week, the present users, and the time the appliance was deactivated. A three-level hierarchical hidden Markov model was used to learn usage patterns in [55] and future usage is predicted by a dynamic Bayesian network. Basu et al. propose to improve learning algorithms by including expert knowledge [4] and, like Chang et al. [8], compare different classifiers for prediction. In the former, where a comparison is made between a C4.5 decision tree, decision table, and Bayesian network, the decision tree and decision table achieve the best results. In the latter, logistic regression outperforms all other standard supervised learning classifiers, including neural networks, support vector machines, decision trees and Bayesian networks. A probabilistic and markov model for appliance usage prediction is respectively used in [67] and [29]. Finally, the benefits of usage profiling for load shifting are illustrated in [20, 61, 67].

In building automation, besides smart heating and dynamic power management, implicit or explicit usage modelling can also be used to achieve savings in lighting [22, 52, 56] and hot water control [21, 48]. Other applications are smart charging of electric vehicles, where environmental and behavioural in-

formation such as driving patterns and electricity cost can be incorporated in the charging process to determine when and how much to charge [28,37]; fault detection and diagnosis, where the aim is to detect anomalies (i.e. deviations from typical behaviour) in system operation [7,40,68]; shared mobility, where users are matched based on their mobility profiles [59]; and elevator group control [10].

While the research discussed above either focusses on a single application and/or a single source of binary information and/or lack a mechanism to cope with evolving data; this paper presents a versatile unsupervised/explicit dynamic multi-state activity pattern mining algorithm. Additionally, a classical problem with clustering algorithms is deciding on the number of clusters. Most standard clustering techniques, such as k-means, assume a fixed number of clusters. Although some methods, such as assessing the quality of the cluster configuration by the silhouette score or the prediction strength, could help determine the appropriate number, the problem is more fundamental. A small sample might not allow the detection of all inherent groups. Therefore, instead of specifying a fixed number of clusters, non-parametric Bayesian models, implemented here by the Dirichlet process mixture [1, 19], allow the number of clusters to grow as more data become available. The proposed algorithm is a multinomial generalization of the binomial models of [12, 58, 60]. The models of [60] and [12, 58] are respectively restricted to usage or no usage and presence or absence. However, people perform a variety of different activities (and many appliances have different modes) requiring distinct actions. For example, in the case of a smart heating system, people might prefer different set points for sleeping, absence, household work, recreation, etc. The contribution of this method lies therefore in its ability to dynamically model these different states for multiple users/devices and various applications. Additionally, outliers are automatically detected and grouped together in an outlier cluster.

3 Non-parametric user modelling

The aim of the modelling algorithm is to find natural groups in activity data, such that tailored control can be provided. This entails identifying patterns indicating the probability of the residents' activity (e.g. eating, housekeeping, sleeping, leisure, etc.). Hereto, the data is organised into fixed time frames, such as days or weeks, depending on the periodicity of the data. In turn, these time frames are further segmented into time slots such as hours, minutes, etc. Since human behaviour typically follows a daily routine, a day is considered as the time frame in the remainder of this text. Furthermore, the collected data has a 15-minute resolution, meaning that each day consists of 96 time slots. Therefore, the likelihood of a user's (l) observed activity x on day n at time t is defined by the conditional probability $p(x_{n,l,t}|\mathbf{X}, n, l, t)$ where $\mathbf{X}$ represents the historical activities.

Since the number of groups is not known beforehand, and could in principle be infinite, and the data can consist of multiple activities or include multiple

users, the modelling approach needs to be flexible and versatile. The presented non-parametric modelling algorithm, allows parameters to change with the data, i.e. the method starts by positing an infinite number of hidden clusters and then gradually evolves to a mature clustering configuration. However, as it is impossible to sample an infinite dimensional parameter, the number of clusters is truncated. An upper bound K is set on the true number of classes C, such that $K >> C$. Data points, e.g. days of occupancy information, are assigned to classes by the stick-breaking process [54] which essentially generates weights, by a procedure resembling the breaking of a unit-length stick (hence its name), indicating the probability with which data points fall into each group. This process can be described as follows:

1. Take a stick of length one.
2. Generate a random variable $\mathbf{v}_1 \sim \mathcal{B}(1, \alpha)$. By the definition of the Beta distribution, this will be a real number between 0 and 1, with expected value $\frac{1}{1+\alpha}$, and break off the stick at $\mathbf{v}_1$. The length of this piece is then the first weight w_1, i.e. the proportion of points that will be assigned to cluster 1.
3. Now, take the remaining part of the stick. Generate $\mathbf{v}_2 \sim \mathcal{B}(1, \alpha)$ and break off the stick at $\mathbf{v}_2$. Again, w_2 is the length of the broken off part of the stick, i.e., $w_2 = (1 - w_1) \cdot \mathbf{v}_2$.
4. Repeat this process for all clusters k.

Finally, a good cluster configuration is inferred by Gibbs sampling [27]. First, the class assignments are randomly initialised, i.e. the stick-breaking weights are uniformly distributed over all clusters K ($\frac{1}{K} \forall k \in K$). Then, whilst fixing the assignments of the other data points, each data point is assigned to a class depending on the class assignments and values of all other data points. The weights are in this case determined by the number of data points in each cluster, and the likelihood of the data points' weekday and activity pattern given the cluster's weekday distribution and prototype respectively. Clusters with only a very limited number of data points (e.g. $< 3\%$) are considered outliers and grouped together in an outlier cluster. This process is repeated until the assignments are stabilised and only C clusters remain. In the following section, the algorithm is described in more detail by means of the Bayesian model in Figure 1 where: L, N and K respectively represent the number of users, number of days in the data set, and maximum number of classes; the day of the week $q_n \in \{1, \ldots, 7\}$ is parametrised by multinomial ($\mathcal{M}$) distribution σ_k with prior Dirichlet ($\mathcal{D}$) distribution γ, the observation $\mathbf{x}_{n,l} \in \{0, 1\}$ is parametrised by the multinomial distribution of the classes μ_k with prior Dirichlet distribution β, and class assignment $\mathbf{z}_n$ is generated from the multinomial distribution π_k with prior Beta ($\mathcal{B}$) distribution α (stick-breaking process).

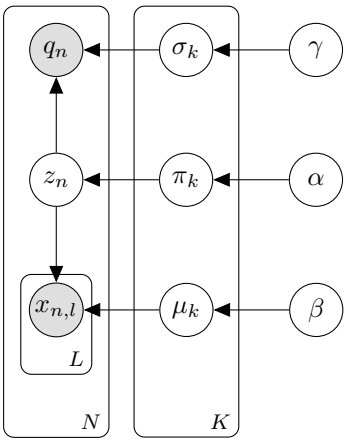


Fig. 1: Bayesian model for finding typical behaviour in historical data [60]. Shaded nodes represent observed information.

3.1 Dirichlet process mixture

Random initialisation Initially, the weights of the stick-breaking process are uniformly assigned: $\pi_k = \frac{1}{K}, \forall k \in K$. Then, the observed days $\mathbf{x}_{n,l}$ are randomly assigned to the K different classes by sampling the multinomial distribution $\mathcal{M}(\mathbf{z}_n|\pi)$, where $\pi = [\pi_1, \ldots, \pi_K]$. As a result, approximately the same number of days will be assigned to each cluster k. After this random allocation, the method iteratively resamples from the posterior distributions of the unknown random variables $\pi_k, \mu_k, \sigma_k,$ and z_n.

Sampling π_k First, new stick-breaking weights are computed by:

$$\pi_k = \mathbf{v}_k \prod_{i=1}^{k-1}(1-\mathbf{v}_i), \mathbf{v}_k \sim \mathcal{B}(\mathbf{v}_k|1+\sum_n \mathbf{z}_{n,k}, \alpha + \sum_{i=k+1}^{K}\sum_n \mathbf{z}_{n,i}) \quad (1)$$

$$\{\mathbf{Z}\}_{n,k} = \{\mathbf{z}_{n,k}|\mathbf{z}_{n,k} = \{0,1\}, \text{and} \sum_k \mathbf{z}_{n,k} = 1\}$$

where α ($\in \mathbb{R}^+$) is a preset hyperparameter, $\{\mathbf{Z}\}_{n,k}$ is an indicator which specifies whether day n belongs to cluster k. Now, the random variables $\mathbf{v}_k$ are generated by sampling from the beta distribution of 1 + the number of days in cluster k and α + the number of days in clusters $k+1$ till K. Then, the proportion of data points that will be assigned to cluster k, π_k, is obtained by multiplying $\mathbf{v}_k$ with the length of the remainder of the stick, which is given by $\prod_{i=1}^{k-1}(1-\mathbf{v}_i)$. The algorithm tends to create more clusters for larger values of α. However, this effect is cancelled out by a large number of iterations.

Sampling μ_k Then the classes (μ_k), indicating the probability of the different activities for each user at each time slot, are generated. Hereto, the Dirichlet

distribution of the number of state occurrences for user l at time slot t on the days assigned to cluster k, is sampled for each user l and each cluster k at each time slot t:

$$\mu_{k,l,t,s} = \mathcal{D}(\mu_{k,l,t,s}|\beta_{k,l,t,s} + \sum_n x_{n,l,t}\mathbf{z}_{n,k}, 1) \tag{2}$$
$$\{\mathbf{X}\}_{n,l,t} = \{\mathbf{x}_{n,l,t}|\mathbf{x}_{n,l,t} = \{0,1\}, \text{and} \sum_s \mathbf{x}_{n,l,t} = 1\}$$

where β is a preset hyperparameter, $\{\mathbf{X}\}_{n,l,t}$ is an indicator which specifies whether activity s ($\in S$) occurs on day n at time t for user l, $t \in$ the number of time slots T and S represents the number of activities. Suppose for example that there are two activities s_1 and s_2 (e.g. absent and present), a single user and only one cluster. Then, if s_1 occurs significantly more than s_2 at time slot 1, Equation 2 will, for this time slot, return a higher probability for s_1 than for s_2. Hyperparameter β is an array of size KxLxTxS, with values $\in \mathbb{R}_{>0}$, which indicates prior knowledge of user behaviour. Thus, each cluster prototype μ_k represents a sequence:

$$\mu_k = \begin{bmatrix} \mu_{k,l,1,1} & \cdots & \mu_{k,l,1,S} \\ \vdots & \vdots & \vdots \\ \mu_{k,l,T,1} & \cdots & \mu_{k,l,T,S} \end{bmatrix}$$

Now, the likelihood of observed user activity for day n (i.e. $\mathbf{x}_n$) belonging to cluster k, is defined by:

$$p(\mathbf{x}_n|\mu_k) = \prod_{l=1}^{L}\prod_{t=1}^{T} \mu_{k,l,t,s} \tag{3}$$

Sampling σ_k Here, the weekday probabilities for a given cluster are generated by a similar process as for μ_k. That is, sampling the posterior distribution of weekday occurrences per cluster:

$$\sigma_{k,w} \sim \mathcal{D}(\omega_k + \sum_n \mathbf{q}_n z_{n,k}, 1) \tag{4}$$

where $w(\in$ the number of distinct weekdays $W)$ indicates the weekday, and ω is a preset hyperparameter: an array of size KxW with values $\in \mathbb{R}_{>0}$, which represents prior knowledge of the weekday distribution. If cluster k mainly contains weekend days, then Equation 4 will produce higher probabilities for Saturday and Sunday than for the other weekdays. The weekday likelihood for cluster k can then be computed by:

$$p(\mathbf{q}_n|\sigma_k) = \sigma_{k,w} \tag{5}$$

where $w = \mathbf{q}_n$.

Sampling Z_n Finally, the observed days are reassigned to the different classes by combining Equations 3 and 5 and the stick-breaking weights π_k to compute the likelihood of the observations for the different day types, i.e. π^*. In turn the multinomial distribution, $\mathcal{M}$, of π^* is used to sample the cluster assignments $\mathbf{z}_n$ for each day:

$$\mathbf{z}_n \sim \mathcal{M}(\mathbf{z}_n|\pi^*), \pi^* := \frac{\pi_k p(\mathbf{x}_n|\mu_k)p(\mathbf{q}_n|\sigma_k)}{\sum_k \pi_k p(\mathbf{x}_n|\mu_k)p(\mathbf{q}_n|\sigma_k)} \tag{6}$$

This sampling procedure is repeated until the number of clusters is constant for Φ_1 iterations, inflicting a state of pseudo-convergence as Gibbs sampling cannot achieve complete convergence. Then, to obtain independent and identically distributed samples, every third sample of the last Φ_2 iterations is selected. The mean of these samples represent the cluster prototypes. Finally, each day is assigned the most frequently occurring day type, i.e. $\mathbf{z}_n$ equals the mode of $\mathbf{Z}$ where $\mathbf{Z} = \{\mathbf{z}_{n,1}, \ldots, \mathbf{z}_{n,\frac{\Phi_2}{3}}\}$. For the experiments in this paper, Φ_1 and Φ_2 were respectively set to 800 and 100. These values are sufficiently large to ensure pseudo-convergence and representative class assignments. Furthermore, there was no prior behaviour information available, imposing a uniform distribution on the β and γ hyperparameters. Lastly, hyperparameter α was set to 1. The influence of α on the number of classes, reduces with increasing Φ_1. The identified day types for the different case studies are discussed in Section 4.3 and illustrated in Figures 3, 4 and 5.

3.2 Evolving data

The algorithm described in the preceding section allows to extract activity patterns from a set of historical information. However, over time behaviour patterns may evolve, known as concept drift in the machine learning field. In order to adapt to these changes, the extracted clusters have to be updated. The stick-breaking algorithm, as defined in Equation 1, can be used to assign a new day of data to any of the existing patterns or to a newly emerging cluster. Alternatively, the modelling method can be extended by the algorithm proposed in a previous work [13] which dynamically updates the identified clusters by performing the following six actions each time a new day of information becomes available:

1. The current cluster compositions are analysed to check whether two or more clusters are very similar and should merge to one bigger cluster.
2. Additionally, the uncertainty of each cluster is investigated. Extended periods of high uncertainty, i.e. similar probabilities for two (or more) states, result in the cluster being split into two (or more) parts. This occurs when a user alternates between activities for these periods (e.g. a biweekly routine). The cluster is then split into one for each of the activities.
3. The new day is matched with the existing clusters, using Equation 3, and assigned to the most likely cluster (if they are sufficiently similar), which updates the cluster prototype.

4. After assigning the new day to its corresponding cluster, the weights of the days in each cluster are updated. Starting with a weight of 1, the contribution of data is gradually decreased over time by applying a decay function, assuming more recent data conveys more relevant information.
5. If the new day did not match with any of the existing clusters it is stored in a potential new cluster. When enough days are added to this potential cluster, it will be activated.
6. Finally, when the number of clusters becomes too big, obsolete clusters, i.e. clusters that have not been updated for a long time and thus have low weights, are removed.

This way, the clusters will gradually adapt to the changing behaviour. However, when completely different behaviour is exerted, it is initially considered an outlier. The model only incorporates the new behaviour when it has occurred multiple times. As such, the algorithm deals with the plasticity-stability dilemma [6], which is manifested in the trade-off between stability, in order to handle noise, and flexibility, to be able to learn new patterns. This strategy can be further improved by tracking the predictive performance of the system and triggering a reclustering stage, i.e. running the modelling algorithm on the new data, in case of a significant reduction in accuracy.

3.3 Prediction

Future behaviour is typically highly correlated with previous activity and weekday information. Hence, the likelihood functions of Equations 3 and 5 can be used to predict future behaviour. As soon as some new observations become available, the overall likelihood of the observed behaviour so far, $x_{n+1,0:t}$, belonging to each of the clusters is computed. This is achieved by multiplying the cluster likelihood, i.e. the stick-breaking weights π_k, with the weekday likelihood $\sigma_{k,w}$, where w represents the weekday on which the observations are made, and the pattern likelihood $p(\mathbf{x}_n|\mu_k)$. The last two values are respectively found by Equations 5 and 3. Then for each of the clusters, the next value, $\mu_{k,t+1}$ is returned for prediction. The final prediction is then obtained by the average of these values, weighted by the overall likelihoods.

4 Experimental results

To evaluate the performance of the presented algorithm, several case studies were performed. This section first describes the experiment setting and evaluation method, followed by an analysis of the results.

4.1 Setting

Experiments were conducted on three different cases with varying data properties to illustrate the algorithm's versatility. The first two cases are concerned

with occupancy modelling of university offices. The rooms were equipped with a wireless sensor network, monitoring the presence of its occupants for 732 and 181 days respectively. In the second case, the office was occupied by two people as opposed to the single-user office of the first case. The sensor network for both cases consisted of two multi-purpose sensors, measuring motion, ambient temperature, ambient lighting and humidity; and one door sensor to register the position of the door. For the shared office, the motion sensors are dedicated to one user by orienting them such that they cover the part of the office occupied by the respective user. In addition, the office chairs were equipped with pressure sensors. This allows to accurately distinguish between both users. Occupancy is then derived by a simple post-processing logic, consisting of simple rules. Presence is assumed when pressure on the chair is registered, or when the door is open and motion is detected, or when the door is closed but motion is detected between 6 am and 21 pm. The detection rules are validated by 3 weeks of ground truth data, collected by the occupant of the single-user office by means of a diary, resulting in an occupancy detection accuracy of 97%.

The aim of the third case is to model an extended set of activities. To this end, the WSU CASAS Kyoto 2010 data set [9], which contains annotated time series data of 13 distinct activities such as sleeping, watching TV, housekeeping, leaving home, etc. for 2 users, was used. Since occurrences of some activities were scarce[1], and to limit the number of different activities, related activities were grouped together. As such, the activities "Eating" and "Meal preparation" were combined into a single "Food"-related activity. "Enter home", "Housekeeping" and "Wandering in room" were merged in a "Go about" state; "Bathing", "Bed toilet transition" and "Personal hygiene" in a "Sanitary" activity; "Watching TV" and "Work" (actually computer activity) in a "Leisure" state; and "Sleep" and "Sleep not in bed" in a "Sleep" state. A "Going out" state is defined, which starts with a "Leave home" and lasts until an "Enter home" event. When none of the preceding activities are in effect, the "Go about" state is presumed.

The data sets of all cases consist of time-series of occupancy states/activities with a 15-minute resolution, i.e. 96 time slots per day, for each distinct user. Each state is represented by a distinct number, e.g. 1 = 'Present'. The data sets of the first and third case study consist of approximately 5 months of activity information. For the second case study only about 4 months of data were available. Each data set contains multiple occurrences of each weekday to ensure capturing the weekday distribution of the behaviour. These data sets are then split into a training and test set for each case, resulting in a training set of 14 weeks for Case 1, 12 weeks for Case 2 and 3; and test sets of 2, 4 and 8 (except for the third case) weeks. An overview of the cases is presented in Table 1.

[1] As some activities, such as eating, occur multiple times per day, scarceness could possibly be attributed to sensor failures or missing annotations. However, given it is an external data set, this could not be investigated. Additionally, these activities are typically very short and since the data is segmented into 15-minute time slots, for which only one activity can be in effect (the one that was started the last in that time slot), some occurrences might be lost.

Table 1: Overview of different case studies.

#	Description	Type	Training set	Test set
1.	Occupancy information of a professor's office	Single-user, binary	14 weeks	2, 4 and 8 weeks
2.	Occupancy information of a shared office (2 people)	Multi-user, binary	12 weeks	2 and 4 weeks
3.	Activities of daily living	Multi-user, multi-state	12 weeks	2, 4 and 8 weeks

For each case, the training data is then fed into the modelling algorithm to identify the user's/users' different activity patterns, i.e. cluster prototypes. These patterns, in combination with the weekday (σ) and cluster (π) likelihoods, are in turn used by the prediction method by comparing them to the available testing data for the new day.

4.2 Evaluation

The proposed profiling algorithm is applied on these three data sets and evaluated both on its clusters' quality and predictive performance. Generally, there are three approaches, i.e. external, internal and relative, for assessing cluster validity [25]. The first approach, i.e. external, evaluates the clustering result based on a pre-specified structure which reflects our intuition about the structure of the data set. The second approach (i.e. internal), evaluates the cluster configuration in terms of quantities that involve vectors of the data set itself, such as a proximity matrix and compare it with a predefined clustering scheme. Approaches based on relative criteria, finally, compute scores with the goal of comparing clustering schemes, resulting from the same algorithm with varying input parameters, in order to find the best one. The first two approaches (those based on external and internal criteria), which rely on statistical hypothesis testing, are not suited for this task as they establish the degree of the cluster configuration confirming an a-priori defined scheme [26]. Those based on relative criteria, such as the distance between clusters or the diameter of the clusters, are suited for the given application. However, they aim to rank different cluster configurations and therefore do not necessarily provide normalised scores, i.e. scores ranging between 0 and 1 (e.g. Dunn's, Davies-Bouldin, SD and S_Dbw index). Thus, they are not always easily interpretable. The cluster silhouette, is a measure, based on relative criteria, for the goodness of fit of the clustering scheme, and will be used to assess the quality of the clustering configurations. The scores range between -1 and 1, where 1 is the best value, negative values indicate wrong assignments, and values near zero indicate overlapping clusters.

In addition, the likelihood of a day belonging to any of the formed clusters can be calculated with Equation 3. Computing this score for every day of the training set provides insight into the quality of the clustering configuration.

Days assigned to cluster k should have high likelihood scores for cluster k and low scores for the other clusters, allowing the analysis of the cohesion and separation of the identified clusters. However, this is a product of probabilities and therefore not very informative on itself, as the numbers will be very small, making them hard to interpret and compare. Therefore, the geometric mean of the likelihood ℓ, is computed as defined in Equation 7.

$$\ell = \sqrt[T*L]{p(\mathbf{x}_n|\mu_k)} \tag{7}$$

For each day of the training set assigned to cluster k, this likelihood score is computed for each of the clusters and their median score is calculated. As a result, cluster k should have a high likelihood (intra-cluster score) score whereas the scores for the other clusters (inter-cluster scores) should be substantially lower. This process is repeated $\forall k \in C$.

The prediction method has been evaluated on a test set of two, four and eight weeks of occupancy information. The method predicts the next activity based on prior cluster probability, weekday information and the already available data of the new day. Each time a new observation becomes available, the prediction is updated. Standard metrics, such as accuracy, f1-score, precision, recall, and logarithmic loss, are used to assess the results. The latter is especially of interest as it heavily penalises classifiers that are confident about incorrect classifications. Furthermore, the results are compared with two baselines, majority class and average weekday prediction, for validation. The first baseline always predicts the most frequently occurring class in the dataset, i.e. the class with the highest support. The second, computes an average day for each day of the week. Then during prediction, the pattern corresponding to the weekday of the test data is used for prediction.

However, the frequency of activities is typically uneven, for example, people generally spend only a third of their time (8h) at the office. This imbalance has to be dealt with. As pointed out in [67], accuracy is not an appropriate performance metric in such scenarios, as predicting the majority class will already result in relatively high scores (depending on the extent of the imbalance). Under- or over-sampling methods help overcome this problem. However, these cannot be applied as data of each time slot of a day is required, thus no data points can be removed or added. Therefore, evaluation metrics that take this imbalance into account are introduced.

4.3 Cluster quality analysis

The iterative clustering process for the first case study is illustrated in Figure 2. At first, the different days of the 14 weeks of occupancy information are randomly distributed over all clusters (in this case $K = 20$). Then, the iterative resampling process gradually reassigns the days to obtain a suitable configuration. It is apparent from the image that the number of clusters quickly stabilises to four. Between iterations 600 and 700 some days of Clusters 2 are reassigned to Cluster 4 and switch back afterwards. This is a result of the

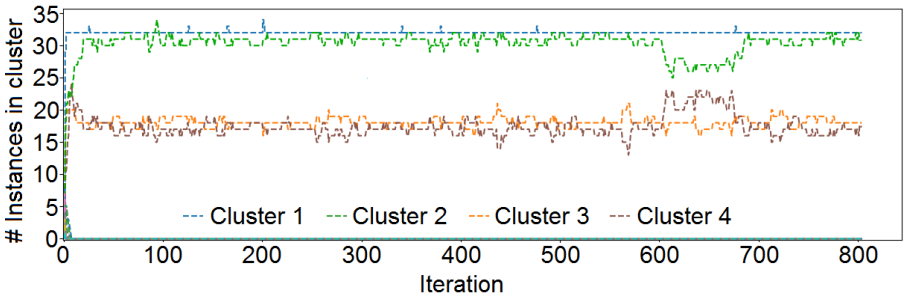


Fig. 2: Evolution of the iterative clustering process for Case 1. The size of the clusters varies with every iteration. The process converges to four clusters after only a few iterations.

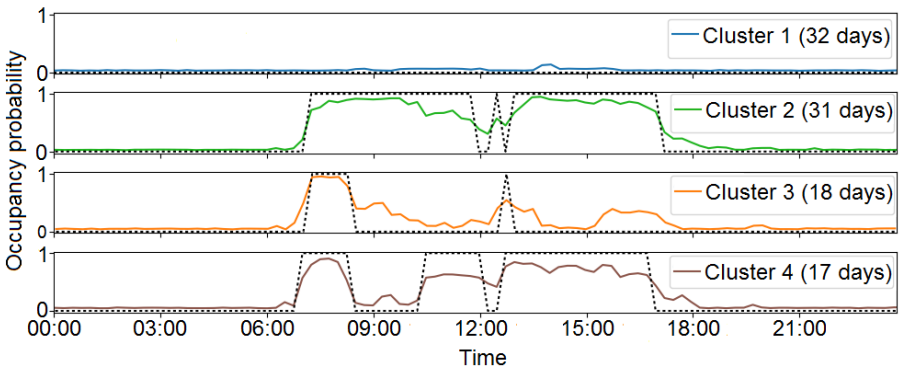


Fig. 3: Extracted typical patterns for Case study 1, where the black dotted line indicates the most probable state at each time slot (arg max $\mu_{k,l,t}$). Clusters 1, 2, 3 and 4 respectively contain 32, 31, 18 and 17 days.

fact that clusters 2 and 4 are quite similar, as is evident from Figure 3 which depicts the prototypes (i.e. μ_k) of the clusters, and of the random properties of the sampling process. The latter refers to the fact that drawing samples from a distribution, as in the sampling of π_k, σ_k, and Z_n, is a stochastic process. At the end, Clusters 1, 2, 3 and 4 contain 32, 31, 18 and 17 instances respectively. A possible description of the identified patterns could be: weekends and holidays for Cluster 1, normal working day with lunch break for Cluster 2 and working day with some lectures and/or meetings for Clusters 3 and 4. In Cluster 3, probabilities around lunchtime and the late afternoon are close to 50%, meaning that sometimes his meetings and lectures take up almost the entire day (starting from approximately 8.30 in the morning). And other times the meetings and lectures are confined to periods between 9 and 12 and/or 13 and 15 o'clock. The most likely state at each moment in time is indicated by a black dotted line in Figure 3. There appears to be a spike in presence at lunchtime in both Cluster 2 and 3. Data cleaning operations could be ap-

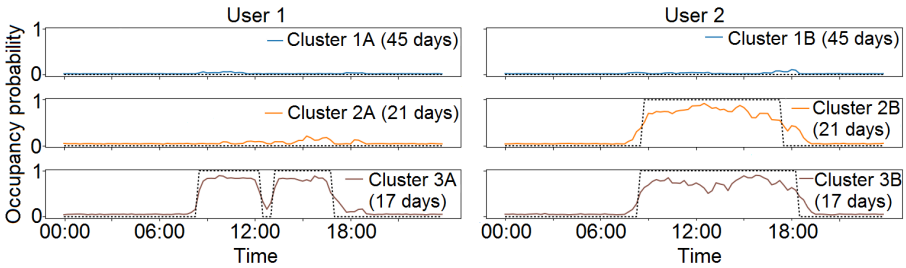


Fig. 4: Extracted typical patterns for Case study 2, where the black dotted line indicates the most probable state at each time slot (arg max $\mu_{k,l,t}$). The left and right figures respectively show the profiles of User 1 and 2. Clusters 1, 2 and 3 respectively contain 45, 21 and 17 days.

plied to remove such spikes. However, the spikes are explained by the fact that the professor usually grabs his lunch before going to the cafeteria. Smoothing the data would result in losing this information, and more generally all brief activities, which could be valuable for certain applications, such as modelling activities of daily living (ADL) as in the third case study.

The results of the cluster validity assessment are presented, along with the results of the other case studies, in Table 2. The intra-cluster, or cluster cohesion, scores demonstrate a (relatively) high level of similarity between the prototypes and the cluster items. Furthermore, inter-cluster scores are significantly lower, indicating that the clusters are well separated. However, Clusters 1, 3, Clusters 2, 4 and Clusters 3, 4, though clearly distinct, share some similarities as can be seen in Figure 3, explaining their higher inter-cluster score (74.5%, 68.3% and 66.8%). Moreover, these commonalities account for the less pronounced difference between the inter- and intra-cluster scores for Cluster 4, and the limited silhouette coefficient of 0.42.

The second case, concerning the occupancy of a multi-user office, again shows fairly high intra-cluster scores with 97.2%, 83.7% and 79.9% for Clusters 1, 2 and 3 respectively, as listed in Table 2. From the identified patterns in Figure 4 it is clear that they represent the following configurations during business hours: both persons absent; person 1 absent and person 2 present; and both users present. Here, each cluster consists of two patterns, one for User 1 and one for User 2. As such, they also depict the interdependency of both users. For example, while the patterns for User 2 are quite similar for Cluster 2 and 3, they are very different for User 1. When User 2 exerts the behaviour of the pattern 2B, the presence pattern of User 1 will look like that of Cluster 2A. The limited inter-cluster similarities also translate into a quite high silhouette score of 0.70, meaning that the days are well matched to their own cluster and poorly matched to the other clusters.

The multitude of activities complicates the final case study, as is evidenced by the lower intra-cluster scores. Five clusters are identified by the algorithm, as is apparent from Figure 5. Nevertheless, the pattern matching results, with

Table 2: Cluster quality results for all case studies. Intra-cluster likelihood should be high and is computed by matching the cluster prototype with each day belonging to that cluster. Lower scores should be obtained for inter-cluster likelihoods, which are computed by matching those days with the other cluster prototypes.

Case 1

		Inter-cluster			
Intra-cluster		*Cluster 1*	*Cluster 2*	*Cluster 3*	*Cluster 4*
96.2%	*Cluster 1*	-	46.8%	74.5%	61.7%
81.7%	*Cluster 2*	33.5%	-	55.9%	68.3%
81.2%	*Cluster 3*	60.7%	55.8%	-	66.8%
76.4%	*Cluster 4*	45.4%	63.1%	63.3%	-

Silhouette coefficient: 0.42

Case 2

		Inter-cluster		
Intra-cluster		*Cluster 1*	*Cluster 2*	*Cluster 3*
97.2%	*Cluster 1*	-	72.2%	54.5%
83.7%	*Cluster 2*	55.2%	-	65.1%
79.9%	*Cluster 3*	33.1%	58.1%	-

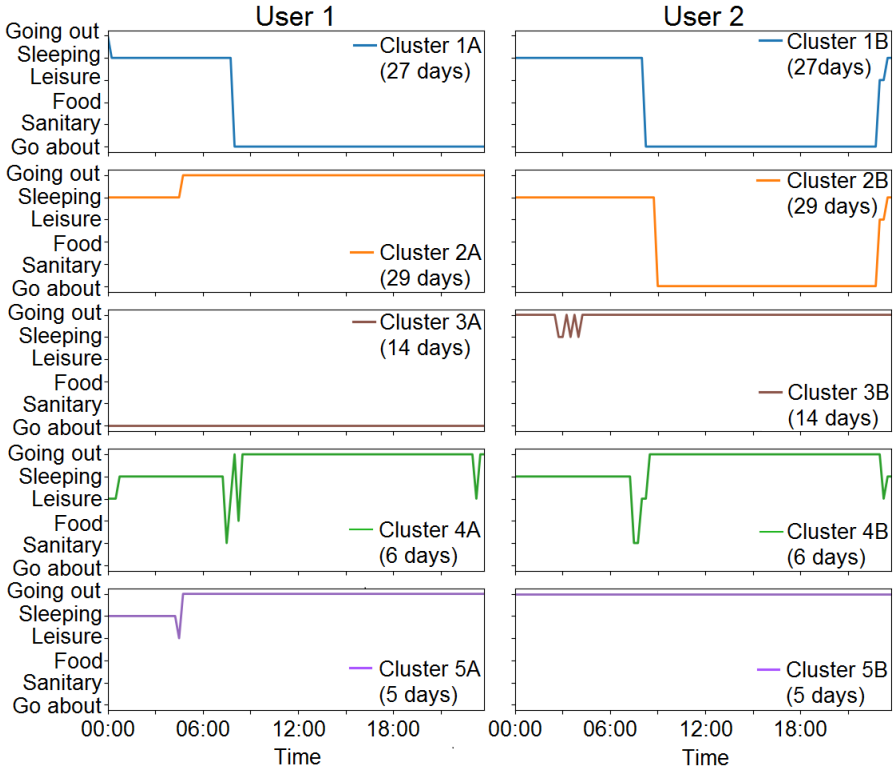
Silhouette coefficient: 0.70

Case 3

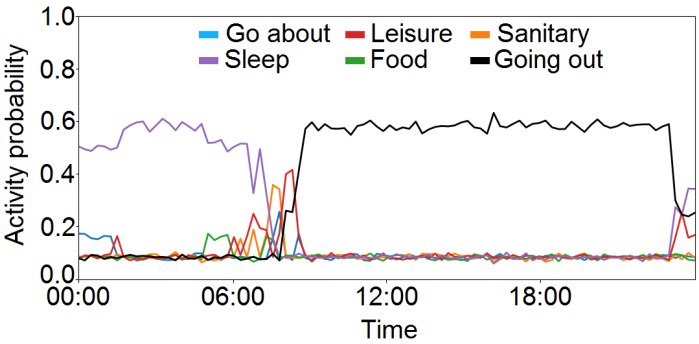
		Inter-cluster				
Intra-cluster		*Cluster 1*	*Cluster 2*	*Cluster 3*	*Cluster 4*	*Cluster 5*
66.3%	*Cluster 1*	-	21.2%	20.2%	13.9%	10.3%
71.2%	*Cluster 2*	20.8%	-	7.9%	25.4%	20.7%
59.1%	*Cluster 3*	14.7%	4.3%	-	18.9%	21.2%
49.1%	*Cluster 4*	7.7%	20.7%	19.5%	-	31.3%
49.8%	*Cluster 5*	4.1%	15.1%	19.6%	34.7%	-

Silhouette coefficient: 0.66

intra-cluster likelihoods significantly higher than the inter-cluster ones, substantiate that also for these complex cases a suitable cluster configuration can be found, resulting in a silhouette score of 0.66.



(a) Most likely activities of the cluster prototypes for Case study 3.



(b) Detailed depiction of one cluster prototype illustrating the probabilities of each activity at each time slot. The most dominant activities are "Sleep" and "Going out". In between there is some probability for various activities, with "Sanitary" and "Leisure" being the most prominent ones.

Fig. 5: Extracted typical patterns for Case study 3. Clusters 1, 2, 3, 4 and 5 respectively contain 27, 29, 14, 6 and 5 days. Top: Most probable state at each time slot (arg max $\mu_{k,l,t}$). The left and right figures respectively show the profiles of User 1 and 2. Bottom: A more detailed, representation of Cluster 3 of User 2, illustrating the probability of each activity at each time slot.

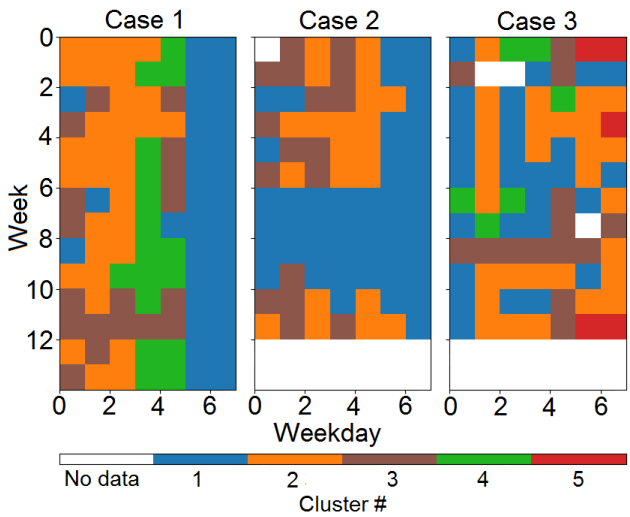


Fig. 6: Cluster membership of training days for the different cases. For the first case 14 weeks of training data were used, while for the second and third case 12 weeks were used.

4.4 Predictive performance analysis

To evaluate the predictive capabilities of the algorithm, an increasingly large test set of 2, 4 and 8 weeks (1344, 2688 and 5376 time slots) was used for each case study. The next state is predicted by combining the weekday, cluster and activity likelihood. The first two can be visually interpreted from Figure 6. For all cases, some patterns of cluster occurrences can be easily recognised. For the first case, Cluster 1 clearly represents weekends and holidays, which confirms the hypothetical description derived from Figure 3. Cluster 4 on the other hand typically occurs on Thursdays or Fridays, while Cluster 2 occurs on Monday till Wednesday. The occurrence of the third cluster, finally, seems to be more stochastic. The weekend and holiday pattern can be observed for the second case as well. This data set was collected over the summer, which explains the occurrence of Cluster 1 for four consecutive weeks. Furthermore, one of the occupants worked part-time in this office, typically from Monday till Wednesday, and part-time in another office. The distribution of occurrences of Clusters 2 and 3 validates this behaviour. Lastly, for the ADL case, Cluster 1 typically occurs on Mondays and Wednesdays while Cluster 2 is most prominent on Tuesdays and weekends. Activities on Fridays regularly match with the pattern of Cluster 3. Clusters 4 and 5, finally, are more random and scarce. Moreover, the occurrence of Cluster 5 is limited to weekends.

Figure 7 shows the prediction for one day of the test set for each case study. The prediction matches the real data quite well. However, it could be further improved by adapting the prediction threshold. For this study, the most likely

state, which means a 0.5 threshold for binary cases, was always selected. From the black dotted line it is clear that lowering or raising this threshold substantially affects the prediction in a positive or negative way. The threshold should therefore be set in an appropriate manner which is typically case specific. For example, in previous work [11], a user tolerance parameter UT was defined, binarising the user profile by discarding, i.e. setting to zero, $UT\%$ of the area under the curve with the lowest probabilities and elevating the remaining probabilities to 1. In a scenario like the first case study, where the profile represents presence probabilities, this would lead to a more conservative prediction (depending on UT), i.e. absence will only be predicted when the system is very confident about it. As such, the level of discomfort (and accompanying potential savings) can be controlled by adapting this parameter.

The overall prediction results are presented in Table 3. For all cases, overall accuracy scores are fairly high, ranging from 85.6% to 93.7%, and significantly surpass the baseline results. Furthermore, the logarithmic-loss, which quickly ramps up when a classifier computes a low probability for the effective state, scores are quite low, varying between 0.279 and 0.178 for the first two cases and 0.573 for the ADL case. This means that on average a probability of respectively $76-84\%$ and 56% is attributed to the correct class. The score for the ADL case is still relatively high given the large number of classes. However, investigating the results for the distinct states separately reveals that for some of them, such as "Sanitary", "Food" and "Leisure" for the third case study, the predictive capabilities are restricted. This phenomenon is explained by the scarceness, i.e. low support, of the respective states/activities which diminishes the predictive power.

As indicated by the support values in Table 3, the case studies are confronted with severely imbalanced classes. The greater the extent of the imbalance, the higher the accuracy of majority class prediction, which explains the fairly high accuracy scores for this baseline. Evaluation metrics that take this imbalance into account, such as Matthews correlation coefficient (MCC), Area Under the Receiver Operating Characteristics Curve (ROCAUC) and Cohen's Kappa (κ), are thus more appropriate for this task. MCC returns a value between -1 (inverse prediction) and +1 (perfect prediction). The ROC curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) for various decision thresholds, while the AUC attempts to capture this information in a single score and ranges from 0 to 1 (perfect classification), with 0.5 indicating total randomness. Cohen's Kappa is in principal a metric for inter-rater agreement. However, it can also be used to asses the agreement between the true and predicted classes. It's more robust than accuracy as it takes the possibility of the agreement occurring by chance into account. κ scores range between -1, i.e. total disagreement, and 1, i.e. complete agreement. A value of 0 is obtained when there is no agreement except what would be expected by chance. Random predictions such as majority class, i.e. one of the baselines in Table 3, return a MCC and κ value of 0 and an AUC score of 0.5. Figure 8 presents the MCC, ROCAUC and κ scores for the different case studies with increasing test periods. Compared to the accuracy scores of Table 3, MCC

Table 3: Prediction results for all case studies.

Case 1				
State	Precision	Recall	F1-score	Support
Absent	96%	95%	95%	1037
Present	83%	85%	84%	307
Average	93%	93%	93%	1344 (Total)
Overall accuracy:	92.6%		Logarithmic loss:	0.197
Avg. Weekday accuracy:	89.5%		Majority class accuracy:	77.2%

Case 2				
Occupant 1				
State	Precision	Recall	F1-score	Support
Absent	96%	97%	96%	1174
Present	78%	69%	74%	170
Average	93%	94%	94%	1344 (Total)
Overall accuracy:	93.7%		Logarithmic loss:	0.178
Avg. Weekday accuracy:	86.9%		Majority class accuracy:	87.4%
Occupant 2				
State	Precision	Recall	F1-score	Support
Absent	92%	92%	92%	979
Present	79%	79%	79%	365
Average	89%	89%	89%	1344 (Total)
Overall accuracy:	88.8%		Logarithmic loss:	0.279
Avg. Weekday accuracy:	78.3%		Majority class accuracy:	72.8%

Case 3				
Resident 1				
State	Precision	Recall	F1-score	Support
Going out	96%	87%	92%	745
Go about	86%	92%	89%	310
Sanitary	0%	0%	0%	6
Food	0%	0%	0%	3
Leisure	0%	0%	0%	30
Sleep	64%	86%	73%	250
Average	85%	86%	85%	1344 (Total)
Overall accuracy:	85.6%		Logarithmic Loss:	0.573
Avg. Weekday accuracy:	60.6%		Majority class accuracy:	55.4%
Resident 2				
State	Precision	Recall	F1-score	Support
Going out	97%	95%	96%	153
Go about	97%	97%	97%	701
Sanitary	0%	0%	0%	9
Food	0%	0%	0%	11
Leisure	45%	27%	34%	52
Sleep	86%	94%	90%	418
Average	90%	91%	92%	1344 (Total)
Overall accuracy:	91.9%		Logarithmic loss:	0.431
Avg. Weekday accuracy:	81.1%		Majority class accuracy:	52.2%

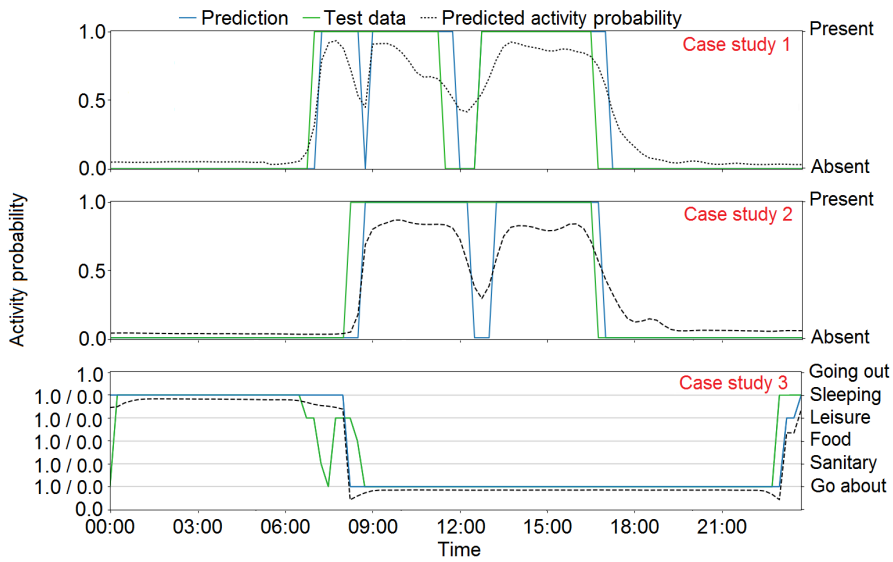


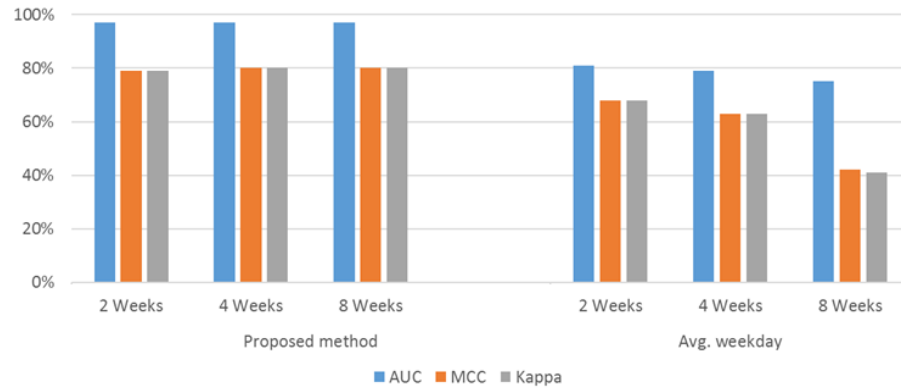
Fig. 7: Example of prediction sequences for one day of each case study. The blue line (Prediction) shows the final prediction. And the activity probability probability is represented by the black dotted line. For the first and second case study the probability of the present state is displayed. For the third case study, the probability of the most likely activity is plotted. Some of the horizontal grid lines have a double function, representing the maximum (1.0) probability of one activity and the minimum (0.0) probability of another activity.

scores are significantly lower. However, overall prediction results are still relatively high and clearly outperform the baseline. Furthermore, the predictive performance of the presented method remains fairly stable when the test period is increased, whereas the performance of the average weekday baseline deteriorates more quickly.

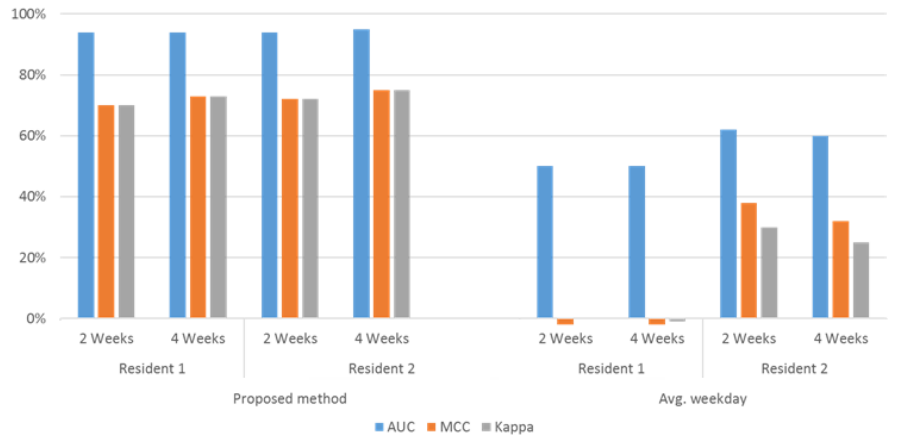
5 Discussion

The results illustrate that the proposed method automatically finds a number of well-defined clusters. Starting from a large number (K) of latent classes, the number of true clusters(C) emerges quickly from the iterative sampling process (as shown in Figure 2). As a result the parameters ϕ_1 and ϕ_2 can be substantially lowered. For each case study, the identified day types, i.e. cluster prototypes, represent natural patterns. Furthermore, the weekday likelihood (σ_k) of those patterns provides additional insight into the user's/users' behaviour. Table 4 presents the weekday likelihood for each case study and corroborate the findings from Figure 6.

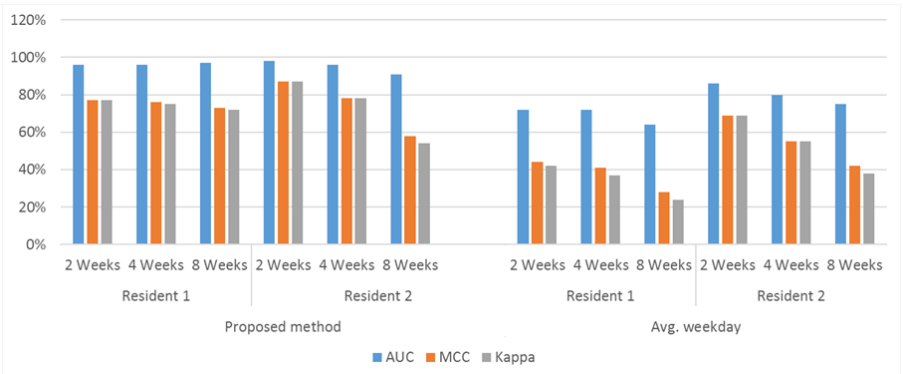
Intra-cluster likelihood is high for all clusters in all cases, indicating good cluster cohesion. The found clusters are generally also well separated. Al-



(a) Case study 1



(b) Case study 2



(c) Case study 3

Fig. 8: Effect of increasing test period on the predictive performance for each case study.

Table 4: Weekday likelihood of identified clusters for each case study.

		Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.
Case 1	*Cluster 1*	0.08	0.05	0.02	0.03	0.06	0.38	0.39
	Cluster 2	0.18	0.31	0.32	0.10	0.05	0.02	0.03
	Cluster 3	0.27	0.16	0.12	0.09	0.28	0.04	0.04
	Cluster 4	0.05	0.04	0.08	0.46	0.29	0.05	0.04
Case 2	*Cluster 1*	0.13	0.10	0.09	0.11	0.09	0.22	0.26
	Cluster 2	0.08	0.10	0.21	0.14	0.34	0.10	0.03
	Cluster 3	0.20	0.29	0.17	0.21	0.05	0.04	0.04
Case 3	*Cluster 1*	0.29	0.03	0.22	0.17	0.09	0.11	0.09
	Cluster 2	0.02	0.28	0.08	0.17	0.08	0.17	0.19
	Cluster 3	0.15	0.10	0.09	0.10	0.39	0.13	0.05
	Cluster 4	0.15	0.13	0.25	0.16	0.17	0.07	0.07
	Cluster 5	0.08	0.08	0.07	0.08	0.09	0.24	0.34

though, some clusters, such as Clusters 1, 3 and 2, 4 of the first case study, have rather high inter-cluster likelihood scores. However, these elevated likelihoods can be explained by the large parts of the day before and after the office hours that are identical for all clusters.

The main applications of the presented algorithm are systems with large inertia, such as smart heating and cooling. Activity affects the user's sense of thermal comfort. Therefore, predicting user activity allows to provide tailored heating control. Another example would be anticipating machine usage to automatically boot devices, such as a coffee-maker [14], photocopy machine, industrial machinery, preheating an oven etc. As such the waiting time can be eliminated. Device usage forecasting also allows renewable energy integration, eliminating standby power consumption (or booting a machine) by dynamic power management, anomaly detection, etc. Other applications would be hot water control and care for the elderly. Anticipating when a user will take a shower allows to condition the bathroom beforehand, or hot water could be circulated such that no water is wasted at the start of a shower, as was done in [21]. Deviations from the activity profile of an elderly person enables to trigger an alert, and thereby requesting help as in the Ger'Home project [2]. Although these examples already demonstrate the versatility of the modelling method, it is not limited to these applications.

Systems with an (almost) immediate reaction time, such as automatic lighting and ventilation systems, are less appropriate. Such systems typically rely on motion sensors to switch on the device, and then leave it on for a fixed amount of time when presence is no longer being detected, i.e. the delay time (e.g. 5 minutes). These systems are reactive rather than predictive, meaning that they act upon the last observation instead of predicting what will happen. It's hard to do better than a reactive system, which essentially maintains (or predicts) the previous observation. As illustrated in Table 5, only when the user (or machine) transitions from one state to another, the system is mis-

[2] http://www-sop.inria.fr/members/Francois.Bremond/topicsText/gerhomeProject.html

Table 5: Example of a reactive system with a prediction horizon of 1 time slot.

Observed state	0	0	0	1	1	1	1	1	0	1	1	0
Prediction of reactive system	/	0	0	0	1	1	1	1	1	0	1	1

Table 6: Comparison of the accuracy of the presented algorithm and a reactive approach in function of an increasing prediction horizon.

Prediction horizon	**Proposed method**			**Reactive system**		
	ROCAUC	*MCC*	κ	*ROCAUC*	*MCC*	κ
1 (15 min.)	97.3%	79.1%	79.1%	80.9%	87.7%	87.7%
2 (30 min.)	96.9%	77.5%	77.5%	80.9%	81.1%	81.1%
4 (1 h.)	96.4%	76.8%	76.8%	80.9%	70.7%	70.6%
8 (2 h.)	95.8%	73.3%	73.3%	80.9%	59.3%	59.3%

taken. For example, for Case 1, the prediction accuracy of a reactive system, for predicting the next time slot, would be 95.6% compared to the 92.6% of the presented algorithm. However, when the prediction horizon is increased, i.e. predicting several time slots in advance, as is required for systems with a larger inertia, the reactive system would lag several time slots behind. For example, in a smart heating scenario, conditioning a room to comfort temperature can take a substantial amount of time (e.g. 1h). With a reactive approach, the system will only start heating when it has observed occupancy, inducing discomfort for the first hour. Table 6 presents the ROCAUC, MCC and κ scores for different prediction horizons for both the presented and reactive algorithm. When the prediction horizon increases, the presented algorithm quickly outperforms the reactive system.

However, as Müller stated in [42]: "There does not seem to be a system that learns quickly is highly accurate, is nearly domain independent, does this from few examples with literally no bias and delivers a user model that is understandable and contains breaking news about the characteristics of the user". Thus, this methodology also has some limitations. Evidently, the data set must be sufficiently large to be able to extract behavioural patterns. Furthermore, the method requires categorical data, a limitation that can be, too some extent, overcome by discretising continuous data. Also, it is limited to a certain domain of applications. As already mentioned, there should be some level of inertia and since the model adapts to changing behaviour, which is desired for the discussed applications, it is not possible to detect gradual changes (necessary for e.g. detecting gradual heath deterioration). Rather, gradual changes will be integrated into the new normal behaviour. Lastly, the sampling process is stochastic, which means that running the algorithm several times will result in a slightly different model each time. A larger number of iterations, increasingly stabilises the process, limiting the stochasticity.

6 Conclusion & future work

In this work, an algorithm for user activity profiling was presented. The proposed method requires few input parameters, limiting the troubles of parameter tuning, and automatically determines the number of classes. The resulting profiles are intuitive, provide insight into the users' behaviour and might be useful for user interfaces of intelligent systems. Three case studies of varying complexity were performed for evaluation. The achieved results, both for modelling and prediction, indicate that the presented strategy can successfully find fitting cluster configurations and anticipate future activities with relatively high accuracy. However, activities should occur frequently in order to achieve a high prediction accuracy. In future work, the usefulness of the algorithm for a smart heating application will be demonstrated by a real-life experiment in a university residence hall. Also, the prediction method will be compared to well-known occupancy prediction strategies. Furthermore, evaluating the method on more data sets, and triggering a relearning stage when the behaviour completely changes (e.g. due to new work hours, new residents) are also identified as future work.

Acknowledgements The authors would like to recognize the financial support from Flanders Innovation and Entrepreneurship (VLAIO).

References

1. Antoniak, C. E.: Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. Annals of Statistics 2. 1152-1174 (1974)
2. Barbato, A., Borsani, L., Capone, A., Melzi, S.: Home energy saving throug a user profiling system based on wireless sensors. Proc. ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings. 49-54 (2009)
3. Barbato, A., Capone, A., Rodolfi, M., Tagliaferri, D: Forecasting the usage of household appliances through power meter sensors for demand management in the smart grid. Proc. IEEE SmartGridComm. 404-409 (2011)
4. Basu, K. Hawarah, L. Arghira, N.n Joumaa, H., Ploix, S.: A prediction system for home appliance usage. Energy Build. 67, 668-679 (2013)
5. Beltran, A., Erickson, V.L., Cerpa, A.E.: ThermoSense: Occupancy Thermal Based Sensing for HVAC Control. Proc. ACM Workshop on Embedded Systems For Energy-Efficient Buildings. 1-8 (2013)
6. Bouchachia, A.: Fuzzy classification in dynamic environments. Soft Computing vol. 15 (5), 1009-1022 (2011)
7. Capozzoli, A., Lauro, F., Khan, I. Fault detection analysis using data mining techniques for a cluster of smart office buildings. Expert Syst. Appl. 42(9), 4324-4338 (2015)
8. Chang, C., Verhaegen, P.A., Duflou, J.R.: A comparison of classifiers for intelligent machine usage prediction. Proc. IEEE IE. 198-201 (2014)
9. Cook, D.J., Scmitter-Edgecombe, M.: Assessing the quality of activities in a smart environment. Methods Inf Med. 48(5),480-485 (2009)
10. Crites, R.H., Barto, A.G.: Elevator group control using multiple reinforcement learning agents. Mach. Learn. 33(2), 235-262 (1998)
11. Duflou, J., Auquilla, A., De Bock, Y., Nowé, A., Kellens, K.: Impact reduction potential by usage anticipation under comfort trade-off conditions. CIRP Annals.65(1), 33-36 (2016)
12. De Bock, Y., Auquilla, A., Kellens, K., Vandevenne, D., Nowé, A., Duflou, J.R.: User adapting system design for improved energy efficiency during the use phase of products: case study of an occupancy-driven, self-learning thermostat. Sustainability through innovation in product life cycle design. 883-898 (2016)
13. De Bock, Y., Auquilla, A., Kellens, K., Nowé, A., Duflou, J.R.: Intelligent Occupancy-Driven Thermostat by Dynamic User Profiling. In Electronics Goes Green 2016+ (EGG), IEEE. 1-8 (2016)
14. De Hauwere, Y.M., Van Moffaert, K., Verhaegen, P.A., Nowé, A.: Networks as a tool to save energy while keeping up general user comfort in buildings. Proc. IEEE LANMAN. 1-6 (2013)
15. Energy Information Administration, Residential Energy Consumption Survey, `https://www.eia.gov/consumption/residential/data/2009/index.php?view=consumption#end-use` (2009). Accessed 1 June 2017
16. Energy Information Administration, Commercial Buildings Energy Consumption Survey, `https://www.eia.gov/consumption/commercial/data/2012/c&e/cfm/e1.php` (2012). Accessed 1 June 2017
17. Erickson, V.L., Achleitner, S., Cera, A.E.: POEM: Power-efficient occupancy-based energy management system. Proc. ACM/IEEE IPSN. 203-216 (2013)
18. European Environment Agency, Household energy consumption by end-use in the EU-27, `www.eea.europa.eu/data-and-maps/figures/households-energy-consumption-by-end-uses-4` (2012). Accessed 1 June 2017
19. Ferguson, T. S.: A Bayesian analysis of some nonparametric problems. Annals of Statistics 1. 209-230 (1973)
20. Fischer, J.E., Ramchurn, S.D., Osborne, M., Parson, O., Huynh, T.D., Alam, M., Pantidi, N., Moran, S., Bachour, K., Reece, S., Costanza, E.: Recommending energy tariffs and load shifting based on smart household usage profiling. Proc. ACM IUI. 383-394 (2013)
21. Frye, A. Goraczko, M., Liu, J., Prodhan, A., Whitehouse, K.: Circulo: saving energy with just-in-time hot water recirculation. Proc. ACM Buildsys. 1-8 (2013)
22. Garg, V., Bansal, N.K.: Smart occupancy sensors to reduce energy consumption. Energy build. 32(1), 81-87 (2000)

23. Global e-Sustainability Initiative, #SMARTer2030: ICT solutions for 21st century challenges, Brussels (2015), http://www.smarter2030.org. Accessed 1 June 2017
24. Gupta, M., Intille, S.S., Larson, K.: Adding GPS-control to traditional thermostats: an exploration of potential energy savings and design challenges. Proc PerCom. 95-114 (2009)
25. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: Cluster validity methods: part I, ACM Sigmod Record 31 (2). 40-45 (2002)
26. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: Clustering validity checking methods: part II, ACM Sigmod Record 31 (3). 19-27 (2002)
27. Ishwaran, H., James, L.F.: Gibbs sampling methods for stick-breaking priors. Journal of the American Statistical Association. 96, 161-173 (2001)
28. Iversen, E.B., Morales, J.M., Madsen, H.: Optimal charging of an electric vehicle using a markov decision process. APP. Energy. 123, 1-12 (2014)
29. Kalksma, M., Setz, B., Pratama, A.R., Georgievski, I., Aiello, M.: Mining sequential patterns for appliance usage prediction. Proc. smartgreens. 23-33 (2018)
30. Kim, S.H., Moon, H.J., Yoon, Y.R.: Improved occupancy detection accuracy using PIR and door sensors for a smart thermostat. Building simulation. 15, 2753-2758 (2017)
31. Kleiminger, W., Mattern, F., Santini, S.: Predicting household occupancy for smart heating control: a comparative performance analysis of state-of-the-art approaches. Energy build. 85, 493-505 (2014)
32. Kleiminger, W., Beckel, C., Staake, T. Santini, S.: Occupancy detection from electricity consumption data. Proc. ACM Workshop on Embedded Systems For Energy-Efficient Buildings. 1-8 (2013)
33. Krumm, J., Bush, A.B.: Learning time-based presence probabilities. Proc. PerCom. 79-96 (2011)
34. Lara, O.D., Labrador, M.A.: A survey on human activity recognition using wearable sensors. IEEE Communications Surveys and Tutorials, 15(3), 1192-1209 (2013).
35. Lee, S., Ryu, G., Chon, Y., Ha, R., Cha, H.: Automatic standby power management using usage profiling and prediction. IEEE T. Hum.-Mach. Syst. 43(6), 535-46 (2013)
36. Li, J., Yang, L., Shum, H., Sexton, G., Tan, Y.: Intelligent home heating controller using fuzzy rule interpolation. In UK workshop on Computational Intelligence (2015)
37. Lopez, K.L., Gagne, C., Gardner, M.-A.: Demand-side management using deep learning for smart charging of electric vehicles. IEEE Transactions on Smart Grid. 10(3), 2683-2691 (2019)
38. Lu, J., Sookoor, T., Srinivasan, V., Gao, G., Holben, B., Stankovic, J., Field, E., Whitehouse, K.: The smart thermostat: using occupancy sensors to save energy in homes. Proc. ACM Sensys. 211-224 (2010)
39. Mamidi, S., Chang, Y.H., Maheswaran, R.: Improving building energy efficiency with a network of sensing, learning and prediction agents. Proc. AAMAS. 1, 45-52 (2012)
40. Mellit, A., Tina, G.M., Kalogirou, S.A.: Fault detection and diagnosis methods for photovoltaic systems: A review. Renewable and sustainable energy reviews. 91, 1-17 (2018)
41. Mozer, M.C., Vidmar, L., Dodier, R.H.: The neurothermostat: predictive optimal control of residential heating systems. Proc. NIPS. 953-959 (1997)
42. Müller, M.E.: Can user models be learned at all? Inherent problems in machine learning for user modelling. The Knowledge Engineering Review 19(1):61-88 (2004).
43. Nacer, A., Marhic, B., Delahoche, L., Masson, J.: ALOS: Automatic learning of an occupancy schedule based on a new prediction model for a smart heating management system. Building and Environment 142:484-501 (2018)
44. Nguyen, T.A, Aiello, M.: Energy intelligent buildings based on user activity: a survey. Energy Build. 56, 244-257 (2013)
45. Ozkan, H.A.: A new real time home power management system. Energy and Buildings. 97, 56-64 (2015)
46. Ozkan, H.A.: Appliance based control for home power management systems. Energy. 114, 693-707 (2016)
47. Peffer, T., Pritoni, M., Meier, A., Aragon, C., Perry, D.: How people use thermostats in homes: a review. Build. Environ. 43(12), 2529-2541 (2011)
48. Prodhan, M.A., Whitehouse, K.: Hot water DJ: saving energy by pre-mixing hot water. Proc. ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings. 91-98 (2012)

49. Purarjomandlangrudi, A., Ghapanchi A.H., Esmalifalak, M.: A data mining approach for fault diagnosis: An application of anomaly detection algorithm. Measurement. 55, 343-352 (2014)
50. Rana, R., Kusy, B., Wall, J., Hu, W.: Novel activity classification and occupancy estimation methods for intelligent HVAC (heating, ventilation and air conditioning) systems. Energy. 93(1), 245-255 (2015)
51. Roy, S., Mishra, K., Maulik, U., Basu, S.: A distributed multilabel classification approach towards mining appliance usage in smart homes. Proc. IEEE Calcutta Conference (2017)
52. Sandhu, J.S., Agogino, A.M., Agogino, A.K.: Wireless sensor networks for commercial lighting control: decision making with multi-agent systems. Proc. AAAI. 131-140 (2004)
53. Scott, J., Brush, A.B., Krumm, J., Meyers, B., Hazas, M., Hodges, S., Villar, N.: PreHeat: controlling home heating using occupancy prediction. Proc. UbiCom. 281-290 (2011)
54. Sethuraman, J.: A constructive definition of Dirichlet priors. Statistica Sinica. 4, 639-650 (1994)
55. Si, H., Saruwatari, S., Minami, M., Morikawa, H.: A ubiquitous power management system to balance energy savings and response time based on device-level usage prediction. Information Processing. 18, 147-163 (2010)
56. Singhvi, V., Krause, A., Guestrin, C., Garrett Jr., J.H., Matthews, H.S.: Intelligent light control using sensor networks. Proc. ACM Sensys. 218-229 (2005)
57. Tapia, E. M., Intille, S. S., Larson, K.: Activity recognition in the home using simple and ubiquitous sensors. Proc. Springer PerCom, 158-175 (2004)
58. Tominaga, S., Shimosaka, M., Fukui, R., Sato, T.: A unified framework for modeling and predicting going-out behavior. Proc. PerCom. 73-90 (2012)
59. Trasarti, R., Pinelli, F., Nanni, M., Giannotti, F.: Mining mobility user profiles for car pooling. Proc. ACM SIGKDD. 1190-1198 (2011)
60. Truong, N.C., McInerney, J., Tran-Thanh, L., Costanza, E., Ramchurn, S.D.: Forecasting multi-appliance usage for smart home energy management. Proc. IJCAI. 2908-2914 (2013)
61. Truong, N.C., Baarslag, T., Ramchurn, G., Tran-Thanh, L.: Interactive scheduling of appliance usage in the home. Proc. IJCAI. (2016)
62. van der Ham, W., Klein, M., Tabatabaei, S.A., Thilakarathne, D.J., Treur, J.: Methods for a smart thermostat to estimate the characteristics of a house based on sensor data. Energy procedia. 95, 467-474 (2016)
63. Van Kasteren, T., Noulas, A., Englebienne, G., Kröse, B.: Accurate activity recognition in a home setting. Proc. ACM UbiComp, 1-9 (2008)
64. Vázquez, F.I., Kastner, W.: Usage profiles for sustainable buildings. Proc. ETFA. 1-8 (2010)
65. Vázquez, F.I., Kastner, W.: Clustering methods for occupancy prediction in smart home control. Proc. IEEE ISIE. 1321-1328 (2011)
66. Ventä, O.: Intelligent products and systems: Technology theme-final report. Technical report, VTT Technical research centre of Finland (2007)
67. Wenninger, M., Schmidt, J., Goeller, T.: Appliance usage prediction for the smart home with an application to energy demand side management. And why accuracy is not a good performance metric for this problem. Proc. smartgreens. 143-150 (2017)
68. Yan, R., Ma, Z., Kokogiannakis, G., Zhao, Y.: A sensor fault detection strategy for air handling units using cluster analysis. Automation in construction. 70, 77-88 (2016)

MSc Yannick De Bock, PhD(c), is a Ph.D. candidate at KU Leuven. His research is focused on user modelling for personalizing products and optimizing the comfort-consumption trade-off through intelligent control. He is also a Data Scientist at Niko NV, a Belgian company specialised in home automation, access control and electrotechnical materials and components.

MSc Andres Auquilla, PhD(c), is a lecturer and researcher at the University of Cuenca in Ecuador. He is also working on his PhD in topics related

to the application of machine learning and statistical techniques oriented to characterize systems that make use of energy to improve their consumption. He is also interested in time series analysis and feature extraction techniques.

Prof. Ann Nowé is full professor at the Vrije Universiteit Brussel (VUB) where she is heading the AI lab. Her main research interest is Reinforcement Learning (RL) including Multi-Agent RL and Multi-criteria RL. More recently she also investigates how to make machine learning models more transparent and explainable to users.

Prof. Joost Duflou is chair professor in the Mechanical Engineering Department of KU Leuven where he among others leads a research group on Life Cycle Engineering with special attention for personalization of product performance as an eco-design strategy for resource efficiency improvement. He has authored over 250 peer reviewed articles and is a fellow of CIRP.