UNIVERSITÀ
DEGLI STUDI
DI PALERMO

NDS
NETWORKING AND DISTRIBUTED SYSTEMS
RESEARCH GROUP

User Activity Recognition for Energy Saving in Smart Homes

Article

Accepted version

Pietro Cottone, Salvatore Gaglio, Giuseppe Lo Re, Marco Ortolani

In Journal of Pervasive and Mobile Computing

Publisher: Elsevier

User Activity Recognition for Energy Saving in Smart Homes

Pietro Cottone, Salvatore Gaglio, Giuseppe Lo Re, Marco Ortolani*

DICGIM, University of Palermo, Viale delle Scienze, ed. 6 - 90128 Palermo, Italy

---

Abstract

Energy demand in typical home environments accounts for a significant fraction of the overall consumption in industrialized countries. In such context, the heterogeneity of the involved devices, and the non negligible influence of the human factor make the optimization of energy use a challenging task; effective automated approaches must take into account basic information about users, such as the prediction of their course of actions.

Our proposal consists in learning customized structural models for common user activities for predicting the trend of energy consumption; the approach aims to lower energy demand in the proximity of predicted peak loads so as to keep the overall consumption within a predefined range, thus minimizing the impact on the end users. In order to build the models, the inherent recursive structure of user activities is abstracted from raw sensor readings, via an approach based on information theory. Experimental assessment based on publicly available datasets and synthesized consumption models is provided to show the effectiveness of our proposal.

Keywords: Activity discovery, peak load avoidance, structural modeling.

---

*Corresponding author.

Email addresses: pietro.cottone@unipa.it (Pietro Cottone), salvatore.gaglio@unipa.it (Salvatore Gaglio), giuseppe.lore@unipa.it (Giuseppe Lo Re), marco.ortolani@unipa.it (Marco Ortolani)

1. Introduction

The ever-increasing energy demand in recent years is becoming a major issue as it represents a possible drawback in our society's future development, where energy is arguably the single most valuable good. Current consumption trends are unsustainable from an environmental point of view, and efficient usage and overall energy demand reduction have become two major concerns of the international community and most governments, due to both economic and environmental motivations [1]. Namely, according to the classical market laws, those trends have caused a burst in energy price which eventually has attracted greater attention to the energy problem.

The periodical shortages in energy supply during the last century, led to the birth of new research areas, and considerable effort is being carried out to devise viable solutions to the energy issue, ranging from discovering new energy sources to raising people awareness. In this context, a steady attention has been devoted to energy saving in buildings, starting from the energy crises of the 1970s [2, 3].

User habits play a central role in household energy demand: an inefficient control of electric appliance and heating systems is a major energy waste source. Current literature about building automation, however, shows that building control is still mainly performed manually, as in the case of artificial lighting setting, powering appliances, or seasonal control of heating systems; additionally, automation in buildings has historically focused on narrow-scope tasks, such as lighting control with simple motion detection and a fixed timeout, or indoor climate control based on temperature and CO_2 level. On the other hand, user activities and behavior have considerable impact on the amount of consumed energy in all kinds of buildings (i.e., residential, office, and retail sectors). Thus, the design of *Building Energy and Comfort Management* (BECM) [4] systems has grown to become a self-standing research area, in order to optimize energy use in home scenario. A significant amount of the energy dissipated in these areas can be saved by fine-tuning deployed devices and appliances according to actual user needs; for instance, many research efforts have been focused on proposing "smart thermostats" based on occupancy prediction, or on maximizing user comfort by providing appropriate artificial lighting, based on the activity carried on at a given moment.

This research area belongs to the greater field of *Ambient Intelligence* (AmI), which encompasses different topics, ranging from environmental monitoring [5] to healthcare [6]; however, while the general scope of AmI is to apply artificial intelligence techniques to transparently support users in their everyday activities, a BECM system can be defined more specifically as a control system that uses artificial intelligence and a distributed sensor network for monitoring a building in order to ensure efficient usage of the available energy sources. A system implementing this approach must be able to predict the users' course of actions, in order to cope with the issue of reducing energy consumption without negatively affecting the user experience. Keeping intrusiveness at a minimum is essential to promote this kind of systems and to allow acceptance by a broad target of users; in fact, their impact on energy consumption will be significant only if they are used at a large scale. Several studies (e.g. [7]) have shown that a user-centric optimization of energy consumption, with no perceivable effects on user comfort, can lead to significant energy saving. In other words, the primary goal of energy saving systems is to automatically adapt to user preferences; this motivated us to follow the AmI paradigm, which requires minimizing user intervention, by "hiding" the system within the surrounding environment, while still enabling support to the users for their everyday-life activities.

Our main focus has thus been on adaptiveness, and our efforts have been specifically directed toward learning and prediction of user activities, as a first step towards an effective approach to energy saving. The present work is an extension of a previous paper, whose focus was the minimization of energy consumption in a home setting by preventing peak demands from exceeding a given threshold [8]. Our approach is based on discovering everyday activities performed by the occupants, and building predictive models for them; while retaining the fundamental focus of the original work, this paper provides a deeper insight into the issue of activity representation and of reliable simulation, and describes a new perspective to modeling user habits by a more efficient coding. In our vision, activities are inherently complex recursive structures; activity recognition may thus be formulated as a data mining task, so that the hidden structure may be identified and reconstructed by means of an unsupervised bottom-up technique based on frequent pattern analysis. This allows us to characterize the features of elementary constituting blocks for an activity, and to hierarchically combine them into more complex objects. A potential drawback of this approach is the computational cost related to building activity models from raw sensor data, and to matching them to newly acquired ones. Our current proposal processes sensory data by considering a description explicitly revealing their hidden structure, by way of a tree-like coding; moreover, we analyze the issue of producing a reliable simulation of the energy footprint of user activities, in order to obtain a more realistic model of energy consumption, and we assess our proposal by considering a new method for energy saving, in a realistic domestic environment.

The remainder of the paper is organized as follows. Section 2 summarizes some of the approaches presented in literature, as regards energy management, activity recognition in smart homes, and energy profile simulation. Sections 3 and 4 present our improved approach to energy saving, highlighting the novelty with respect to [8]. Section 5 provides experimental assessment of our system in comparison to the previous one. Finally, some final considerations are provided in Section 6.

2. Related Work

Substantial research effort has been devoted to address the complex issues related to the design of a BECM system, and most proposals agree on the need for automated approaches to energy demand optimization; the presence of peaks in energy demand is often regarded as a symptom of a suboptimal scheduling of the use of electric appliances and the authors of [9, 10, 11], for instance, point out that even straightforward approaches, such as turning off unused devices, can be very effective in terms of energy saving. The challenging aspect of those proposals is their potential impact on user perception: if automated energy saving policies are so intrusive as to become a hindrance to the overall user experience, they might hardly be accepted from householders.

The key to designing a system capable of adapting to its users' needs is to correctly identify their activities. This is in fact a widely discussed topic in scientific literature, and common proposals include methods based on the use of logic, probabilistic methods, methods based on common sense reasoning, and, finally, data mining approaches; a detailed taxonomy is reported and discussed in [8]. Several state-of-art proposals assume the availability of considerable *a priori* knowledge, which makes them often prone to overfitting. Results obtained by these systems depend on the particular features of the application scenario, and their activity models are fitted onto data, as opposed to "emerging" from data itself [12]; this may be a major issue, if the goal is the design of a fully adaptive and generalizable system. Our work is partly inspired to the key ideas presented in [13] and [14]. The authors of [13], in particular, proceeding from a scenario characterized by scarcity of labeled data and uncertainty about activity granularity, showed that *formal grammars* are suitable to capture the inherent structure of activities. Their system, called Helix, initially generates a vocabulary combining unlabeled sensor readings, and attempts to incrementally merge them, by grouping similar activities into high-level ones. Grammar induction is used as a tool for heterogeneous sensor fusion in order to build up the structure of activities; each activity is regarded as a cluster in a multi dimensional space where the data streams coming from the different sensors present in the monitored area are represented; a hierarchical structure is then induced on this space, through statistical analysis. The authors of [14] focused on formalizing computational models for every-day human activities; they claim that global structural information about activities can be encoded by using a subset of their local event subsequences; hence, an activity is defined as a finite sequence of events, expressed in terms of the objects present in the observed environment, whose functionalities may be needed for the execution of a particular activity. An event is defined as a specific interaction between two or more objects in a finite duration of time, and a list of key objects for each environment needs to be provided as *a priori* knowledge. This approach does not need to rely on predefined activity models, whose creation is typically very challenging, rather it is pointed out that an analysis of continuous event subsequences suffices to discover and track every-day activities.

In order to test the effectiveness of activity recognition for energy saving, a data set including both power consumption and sensor measurement would be needed; however, despite the fact that data sets about activity recognition, as well as about power profiling have been independently collected, to the best of our knowledge none is available that encompasses both aspects. One of the data sets of the *Center for Advanced Studies in Adaptive Systems* (CASAS) project [15], for instance, contains readings from a power meter; however they provide information only about the overall consumption, which is not very useful in the context of activity recognition, where fine-grained energy monitoring is needed. Namely, aggregated information about energy consumption often leads to non optimal consumption control. Indeed, new systems have been developed to produce fine-grained energy reports, at an individual-device scale [16], although in the context of the new research area of "energy reporting", whose aim is that to guarantee a higher resolution in monitoring energy consumptions. In this context, a very promising data set, provided by the Smart* project [17], was collected by continuously gathering measurements from a wide range of sensors and meters placed in three different households; however the sensor set should be significantly enriched before it may be profitably used for activity discovery and recognition. A natural alternative to gathering actual measurements consists in resorting to use synthesized ones; energy demand simulation, in particular, has been widely discussed in scientific literature. The authors of [18] discuss the use of models for end-use energy consumption; they point out that residential consumption represents a substantial part of energy demand in every countries, and suggest a partition of modeling techniques for residential energy consumption into two major classes; top-down, and bottom-up approaches. In the former case, no individual house energy profile is built, rather historic data is aggregated and analyzed to regress the energy model of the whole housing stock; on the contrary, in the latter case, energy consumption is estimated for a representative set of individual houses, and is later generalized to form the residential consumption model. For our purposes, the bottom-up approach is more interesting; its main drawback is the need for detailed information about the home environment (the trend of common environmental measurements might need to be estimated, or simulated [19, 20]; supplier billing data, for instance, is private information, and typically it may be obtained only by disaggregation on the overall consumption); on the other hand, bottom-up techniques are often the only means to evaluate the impact of new systems or technologies, which are likely to lead to more effective power usage optimization.

Some modeling techniques for residential power consumption simulation are reviewed in [21, 22, 23]; those proposals share the idea that realistic energy usage simulation depends on three main factors: occupant behavior (i.e., activities), appliance models, and a model of energy consumption per activity. A slightly different approach, highlighting the importance of user activity simulation, was proposed in [23], where a Markov chain is used to simulate user presence and habits, modeled in terms of nine energy-hungry activities, such as for instance cooking, using a personal workstation, or simply being absent. The work presented in [22] performs energy demand simulation by summing up the contribution of each appliance in a dwelling, in a bottom-up fashion. The authors specifically focus on modeling user "active occupancy" and characterize an activity through a profile, storing its inception time, and duration; each activity profile is assigned to an appliance, strictly tying user presence to energy consumption; this choice also allows to model dependence and time correlation between appliances.

Besides detecting user activities, and linking them to a consumption profile, the ultimate task of a BECM system is the achievement of significant energy saving. In the past years, particular attention has been devoted to the specific issue of avoiding peaks in energy demand, which is a very complex issue, due to the high variability in user consumption demand and to the limited flexibility in scheduling in order not to negatively affect user experience; moreover, price policies adopted by providers are often insufficient to modify user habits and lower peak energy demand. In [24, 25], a demand-side load management system is proposed, suitable to be integrated in the future Smart Grid technology. The proposed system acts in real time, interacting with appliances and users, and adopts a layered structure, processing data coming from actual on-line consumption and schedule user requests, in order to balance electricity demand. Each appliance is modeled as a finite state machine, triggered by events generated by user or the balancing system. The core of the system is the admission control, that manages accesses to power resource and controls appliances. Its scheduling algorithm is heuristic-driven and finds a greedy solution; so the optimality of the solution is not guaranteed. The requests set is checked and, based on the state of appliances and the requested power, the system decides about its delivering. In [26], the authors propose a system to schedule only the so-called background loads, that is refrigerators, dehumidifiers, and so on. An algorithm inspired to the well-known Earliest Deadline First is used; the authors claim that scheduling non background loads may have an impact on user comfort, so they opt against controlling them. Finally, they introduce the concept of *slack*, that is the maximum amount of time a device can be disconnected from power, while still guaranteeing its performance; each load is assumed able to maintain an estimate of its remaining slack time. At fixed intervals, the algorithm checks the slack of each background load and gives priority to the one with the smallest slack; if a load reaches zero slack, then it is powered on, regardless of the increase in energy use. When the aggregated sum of background loads power reaches a prefixed threshold, no other loads are powered. Finally, in the approach presented in [27] the problem of shaving peaks in energy demand is formulated as a mixed integer linear program, in a mixed (i.e., renewable and non renewable) power source scenario. Authors aim at investigating the potential of a combined optimization approach that takes into account every possible kind of loads, namely shiftable, sliceable, stretchable ones, and so on. Each energy-demand task is characterized by a completion deadline, while each day is divided into equal time slices. The goal is to minimize the combined power of all slices. Some constraints are to be met; for instance, each device may be powered by only one source, and the amount of power needed by shiftable loads in each period is constant.

3. Learning User Habits for Energy Saving

The ultimate goal of our system is to shave off peaks of energy consumption by tracking user activities in order to modify the functioning period of appliances that are not immediately useful for the current task; the approach aims to lower energy demand in the proximity of predicted peak loads so as to keep the overall consumption below a pre-set threshold. In order for the system to perform effectively and to be generalizable to previously unforeseen scenarios, it needs to capture and formalize the activities that actually account for user habits.

General *a priori* models of activities, appropriate to exemplify the behavior of any possible kind of user, are too complex to be realistically feasible. Designers are typically able to explain what an activity is in terms of the sensor set actually deployed, but they seldom succeed in describing how each activity can possibly be carried out by every user. In our vision, a general high-level description of what may be regarded as an *activity* is all is required, thus bypassing the difficulty of creating a reliable model of an activity in terms of sensory triggers or supposed interactions between users and their home appliances. The recursive structure of activities is used in order to cope with the complexity of building models from data; to this aim, our system identifies the basic elements of an activity (i.e. *events*, in our terminology), and builds more complex models by mixing up these components. Rather than expressing activities directly through the evidence provided by sensor readings, our system tells them apart by making their distinguishing structural features explicit; namely, we consider *causality*, *critical intervals* and *missing components*, as suggested

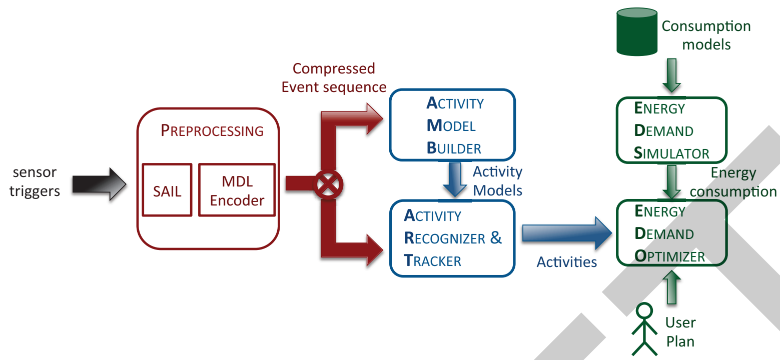

Figure 1: An overview of proposed system.

by the T-pattern model [28]. The system architect is relieved from the need to build a data-driven model for each activity to track, addressing the problem from an *algorithmic* perspective, rather than a *learning* one, involving as little pre-acquired knowledge as possible about the scenario. Accurately discovering user activities and learning reliable models for them is however a very challenging task, so an initial preprocessing step is included, fulfilling two main goals: focusing future computation on the more interesting bits of data, and identifying events; hence, the original undistinguished stream of sensor readings can be translated in a more meaningful stream of events.

Figure 1 shows the overall architecture of the proposed system; energy consumption modeling is implemented by the *Energy Demand Simulator* (EDS) block, whereas our approach to energy saving algorithm, through peak load shaving, is represented by the *Energy Demand Optimizer* (EDO). The system core is represented by the *Activity Model Builder* (AMB) and *Activity Recognizer and Tracker* (ART), preceded by a *Preprocessing* block. The AMB is devoted to provide models of the most common user activities, which will be used by the ART module for on-line recognition; an optimal energy plan may thus by elaborated by the EDO module, on the basis of the energy demand provided by the EDS, the recognized and predicted activities and a user plan, containing the tasks to be executed in a given time interval.

$$\texttt{Ev}_1 \rightarrow \langle \texttt{2.5 5.0} \rangle\ \texttt{M13}_{\texttt{ON}}\ \langle \texttt{1.25 1.5} \rangle\ \texttt{M14}_{\texttt{ON}}\ \langle \texttt{0.75 2} \rangle\ \texttt{M13}_{\texttt{OFF}}\ \langle \texttt{0.5 1.5} \rangle\ \texttt{M14}_{\texttt{OFF}}$$

$$\texttt{Ev}_2 \rightarrow \langle \texttt{2.5 5.0} \rangle\ \texttt{F1}_{\texttt{OPEN}}\ \langle \texttt{1 1.25} \rangle\ \texttt{F1}_{\texttt{CLOSE}}\ \langle \texttt{0.75 1.75} \rangle\ \texttt{F1}_{\texttt{OPEN}}\ \langle \texttt{0.75 2} \rangle\ \texttt{F1}_{\texttt{CLOSE}}$$

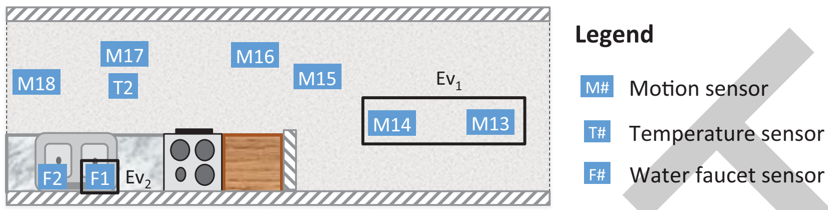

Figure 2: Two sample events extracted by our algorithm: $\texttt{Ev}_1$ captures the user walking toward the kitchen, while $\texttt{Ev}_2$ corresponds to using the kitchen faucet for washing. The maximum duration of events for template abstraction was set to 5s in both cases.

In the following, the detailed descriptions for each of the mentioned modules are provided.

3.1. From sensor data to a compressed event stream

Our basic assumption is that a pervasive deployment of heterogeneous sensors is available over the monitored environment. In order to discover hidden relations between sensor triggers originated by different sources, a preprocessing step is needed; sensor readings can be merged to form templates for the most common events, which can be defined as significant frequently co-occurring triggers. Performing an activity will generate a great number of sensor readings; for instance, breakfast preparation may involve proximity sensors (to the cupboard, to the oven, etc), item sensors (toaster, coffeemaker, taps), and environmental sensors (temperature, water flow), whose state may be represented by a *binary*, *discrete* or *continuous* variable, respectively. We define a *trigger* as the pair composed by sensor ID and sensor state. Representative information must be extracted from a series of raw sensor triggers; to this end, our approach consists in devising a specific language, where an event is defined in terms of triggers according to the following syntax:

$$\texttt{Ev}_{\texttt{ID}}\ \langle \texttt{dur}_{\texttt{min}}\ \texttt{dur}_{\texttt{max}} \rangle\ \texttt{trig}_{\texttt{ID}}\ [,\ \langle \texttt{gap}_{\texttt{min}}\ \texttt{gap}_{\texttt{max}} \rangle\ \texttt{trig}_{\texttt{ID}}\ ]$$

According to this definition, each event is identified by the minimum and maximum expected duration of the whole sequence, an initial trigger followed by an optional sequence of triggers with intervening gaps of duration in the range `[gap`$_{\texttt{min}}$`, gap`$_{\texttt{max}}$`]`. An example of two events extracted by our algorithm is shown in Figure 2.

Discovering event templates enables us to scan previously unseen trigger sequences in order to identify the actual occurrences of the events contained therein. This step is accomplished by a modified version of *String mAtching with wIldcards and Length constraints* (SAIL) [29], an on-line algorithm able to locate patterns as soon as they appear in the sequence, as described in [8]. The use of SAIL transforms the *trigger* sequence into an *event* sequence, ready to be scanned to find frequent and relevant patterns, representing our high-level activities. In order to select instances of the problem that result tractable, thus reducing the complexity of exploring the search space, we exploit concepts of information theory. We aim at compressing the event sequence by lossy optimal coding so that events with low information content will be discarded; in other words, the most relevant patterns will be those that better describe the whole sequence, according to the *Minimum Description Length* (MDL) principle [30]; additionally, the compression of the event sequence allows for a decrease in the computational cost of later processing, thus coping with the exponential complexity of frequent event pattern mining.

Our algorithm for activity discovery is inspired to arithmetic coding and entropy-based compression. In order to find a new, optimal encoding for the event sequence originally produced by SAIL, we regard it as a string of symbols over the alphabet of event IDs. Borrowing the terminology from information theory, we introduce the concept of n-gram, i.e. a subsequence of n contiguous items from a given string: our aim is to translate the original sequence using a new alphabet whose symbols are the most significant n-grams over the event stream. We use a greedy algorithm that returns the n-gram alphabet resulting in a better encoding of the original stream, assessed in terms of the potentially obtainable compression rate. With reference to the pseudo-code shown in Figure 3, the first step of our algorithm aims to extract the frequencies of all the n-grams of size between $n_{\min}$ and $n_{\max}$ from the event sequence. Then, an

Input: string E; int $n_{min},\ n_{max}$

Output: alphabet **a**

nlist $\Leftarrow$ extract_ngrams(E, n_{min}, n_{max})
a $\Leftarrow \emptyset$
while nlist $\neq \emptyset$ **do**
  nlist $\Leftarrow$ sort(**nlist**)
  ngram $\Leftarrow$ getfirst(**nlist**)
  if get_obtainable_compression(**ngram**) $< \theta_{comp}$ **then**
    return a
  else
    a $\Leftarrow$ **a** $\bigcup$ {**ngram**}
    nlist $\Leftarrow$ **nlist** $-$ {**ngram**}
    $E \Leftarrow$ delete(E, **ngram**)
    nlist $\Leftarrow$ update(**nlist**, E, **ngram**)
  end if
end while

Figure 3: Finding a better alphabet for event encoding.

ordered list is obtained from this set of n-grams; the order criterion computes the ability of each n-gram to compress the sequence: each n-gram is viewed as a potential new symbol of a new encoding alphabet, and the length the sequence is re-computed accordingly, using a binary encoding. The head of the list is joined to the new encoding alphabet, the frequencies of the remaining n-grams are updated and the list is newly ordered. The algorithm stops when the compression rate of the head list element falls under a prefixed threshold (θ_{comp}). The algorithm then returns the n-gram alphabet resulting in improved encoding. A detailed description of the algorithm can be found in [8].

3.2. Discovering, modeling, and tracking user activities

We formulate activity discovery as a data mining problem, and we regard frequent recurrent event patterns as instances of the yet unknown activities. Figure 4 depicts the process of information refinement underlying our approach:

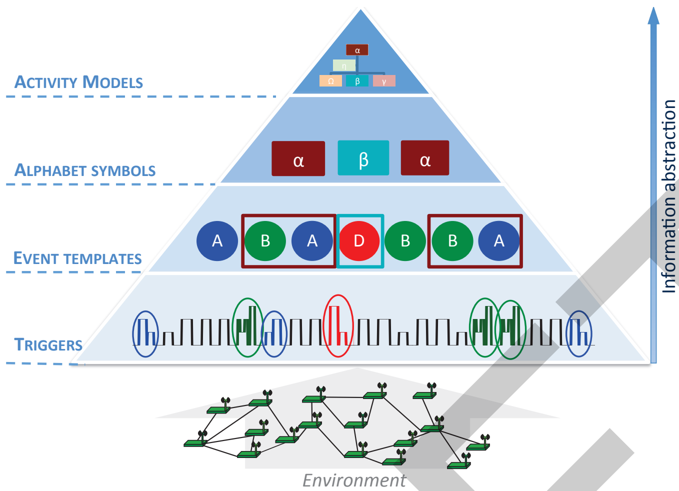

Figure 4: The process of activity discovery as an identification of recurrent structural patterns.

our system attempts to infer models for activities defined as *recursive structures* symbolically expressed in terms of a basic “alphabet”; the process starts by identifying relevant *events*, which, in this context, may be thought of as short and recurrent sequences of triggers, i.e. raw sensor readings. This bypasses the difficulty of creating a reliable model of an activity directly in terms of sensory triggers or supposed interactions between users and their home appliances.

Other proposals adopt a similar approach, but often rely on supervised algorithms, with the aim of looking for a translation of a predefined model of activity into data; however, explaining data through model established in advance implies some constraints and limitations: for instance, all users are supposed to carry out the same activities in a very similar way, and a great amount of data has to be collected and consistently labeled in order to create a sufficiently large training set.

In order to have activities naturally *emerge* from sensor observations, the AMB looks for recurrent structures; given the event sequence obtained from MDL encoding, the most frequent patterns have to be discovered. Our approach consists in a modified version of *Discontinuous Varied-order Sequential*

Miner (DVSM) [31], which is an *a priori*-based iterative algorithm, relying on four main components: a candidate generation function, a pruning function, a candidate set, and a frequent pattern set. Initially, a candidate set is generated by considering the pruned set of all pairs of consecutive events in the encoded sequence. The idea of the algorithm is that each pattern in the candidate set is iteratively expanded, according to a generation function. New patterns are checked against a pruning function, and only the ones surviving pruning are added to the new candidate set. Only those patterns not allowing any further expansion will be part of the frequent pattern set. The algorithm stops when the candidate set is empty. The candidate generation function expands a pattern by adding the previous and the subsequent event in the encoded sequence, in order to create two new patterns. The pruning function is based on the MDL principle, and discards those sets of patterns unable to produce a sufficient compression rate, according to a predefined threshold.

In order to compute the compression rate, DVSM iteratively creates a hierarchical structure: at each step, variations of similar patterns in terms of the Levenshtein distance [32] are grouped together into general patterns.

The final frequent pattern set returned by DVSM contains the most relevant patterns, which will be clustered into meaningful classes to obtain the discovered activities, by integrating temporal information with other features of interest, such as composition similarity, with an approach similar to [31]. This step is accomplished by k-medoids, a variant of the well-known k-means clustering, where representative points are bound to belong to the initial dataset. k-medoids uses a dissimilarity measure computed over all the possible pairs of points, giving it more robustness than traditional k-means measures with respect to noise and outliers [33]. Similarly to k-means, the number of partitions is a parameter chosen by the user; further details for our case will be provided in the experimental section.

The chosen dissimilarity measure reflects our definition of pattern dissimilarity, according to the T-pattern model, and consists of three components:

- *causality* is expressed by the order of the events in the pattern: earlier occurrences within the pattern may provide an explanation for occurrences found later on; therefore, the more dissimilar two patterns are with respect to the order of their events, the higher the probability that they represent instances of different activities. In our approach, causality is implemented by means of the Levenshtein distance;

- *critical intervals* deal with the relations between the distributions of components of a pattern; in other words, this measure considers the time distances between consecutive components. The corresponding function measures temporal information about the pattern element (time of day, duration, etc) and, clearly, the distance between two different components;

- the so-called *missing components*, i.e. the differences between the events present in two patterns, are determined based on the best pair of corresponding events between two patterns, if any.

In order to choose the best partitioning of the original pattern set, the algorithm is run multiple times with different initial random representative points. In the end, we choose the partition that achieves the best overall dissimilarity measure among the obtained clusters. Such clusters constitute the so-called *discovered* activities, i.e. activities emerging from collected data.

In the last phase, we encode the features of the obtained clusters into models representing the discovered activities. We adopt an approach based on boosting; we use *hidden Markov models* (HMM) [34] to describe activities, and we train an HMM for each activity we discovered, using the corresponding cluster set as training set. In the recognition phase, a window of fixed size is slid over the input events, and an activity label is assigned to the last event in the window, according to the HMM that achieves the higher posterior probability in correspondence to that event.

Once models for activities are available, the ART may process the incoming stream of sensor triggers, convert them into event sequences, and use a sliding window on them in order to recognize the current activity; the label assigned to the last element of the window is that of the activity corresponding to the HMM that maximizes the posterior probability.

4. Optimizing Energy Demand by Peak Shaving

Our approach is based on the assumption that recognizing user activities automatically and non disruptively for the inhabitants of the monitored environment is the key to effective energy demand optimization; to the best of our knowledge, no comprehensive dataset is available to date with details about power profiling and the corresponding information about user activities. However, as discussed in Section 2, a few repositories have been created in the context of pervasive monitoring for activity recognition via simple, off-the-shelf sensors; we opted for making use of such publicly available datasets, and enriched them with synthetic information about energy demand.

For the purposes of the present discussion, we extend our previous work [8], and characterize the overall energy demand of a smart home by identifying its main sources, from a user's point of view; energy consumption may thus be seen as the sum of three different components: a *baseline* demand, the (activity-driven) *user loads*, and what we call the *schedulable loads* (see *Energy Demand Simulator* (EDS) block in Figure 1).

4.1. Simulating energy consumption

The *baseline* consumption is generated by all appliances operating in background, such as heaters, dehumidifiers, freezers, refrigerators, and so on. Most loads belonging to this class can be shifted in time, getting a better execution order from an energy saving point of view; moreover, price forecast could be considered in order to minimize costs. Our definition of baseline loads is inspired to the works by [26], and [24]. All such appliances are somewhat transparent to the end user, who does not perceive their presence and does not make an explicit scheduling plan for them. Moreover, they may be assumed to always have an impact on energy demand, as they account for essential services, or are necessary to guarantee a minimal comfort level. The baseline load profile can be modeled by considering a typical usage in an ordinary house. Once the set of baseline appliances is defined, their consumption is predictable according to the most common consumption profile, which may be obtained by referring to well established references. In particular, we followed the study described in [35] and built the energy profile for a few common appliances, matching their respective loads to the previous taxonomy. For instance, the energy demand profile for baseline loads was inferred from the typical use of the corresponding appliances; a daily demand curve was generated based on the data provided by [35], and was parameterized to produce a set of standard daily usages.

User loads are a byproduct of the current user activity; microwave ovens, TV sets, computers represent typical examples of devices belonging to this class. We follow the approach proposed in [26], where the authors choose to leave out all those appliances that can be scheduled by the user in a predefined fashion (e.g. dishwasher); on the other hand energy demand due to user loads is likely unpredictable, hence very difficult to cope with, in order to prevent a negative impact on peak demands. The energy demand due to each activity is simulated by combining the effects of some randomly chosen devices that can be possibly turned on during its execution. For example, cooking may require the use of different appliances (e.g stove, as opposed to microwave oven), so different instances of the same activity may result into very different energy consumption profiles. In order to account for this peculiarity, we only considered the simulated consumption as due to a random selection of devices from the set of all the appliances related to that activity. The coupling between appliances and activities was defined a priori; moreover, for each device activation we randomly chose a duration, by simulating the use of the same appliance in different executions of the same activity.

Schedulable loads, the third component of our energy model, are obtainable by analyzing a plan provided by the user. It includes all the appliances that are characterized by long-lasting tasks, as compared to normal user activities. Washing machines and tumble dryers are typical examples of this kind of appliances, similarly to "burst loads" in the terminology proposed by [24].

Finally, we considered the user plan, which is a predefined list of tasks; for each of them, the user needs to provide two intervals defining the acceptable ranges for the beginning and ending time for the task; moreover, a priority is associated to every task, expressing its importance in the user's opinion. Time intervals associated with tasks may possibly take into account price forecasting, in order to minimize energy costs. The plan also takes into account dependencies between tasks, thus preventing the execution of meaningless chains of tasks. For example, a user might want the `tumble dryer` task executed only after the `washing machine` one; furthermore if, for any reason, `washing machine` was not executed, then neither `tumble dryer` should be. The idea of including a user plan might be profitable in other contexts as well; for instance, in a scenario where energy cost minimization is required, priorities can be chosen according to dynamic price strategies.

4.2. Peak shaving

Our approach to the optimization of energy demand is implemented by the EDO block in Figure 1; it is focused on peak avoidance, considering the estimates of the baseline, user and schedulable loads. In our formulation, energy optimization is regarded as a variant of the *knapsack optimization problem* (KP) [36], a theoretical approach that has already been applied to several practical fields. KP belongs to the integer combinatorial optimization domain, and encompasses a set of problems in the field of integer linear programming. It is known to be an NP-complete problem, and it has been widely studied due to its possible applications, ranging from financing to resource distribution; it has also found application in the context of energy optimization [11]. Given a set of objects, characterized by a *volume* and a *value*, the KP aims at selecting the best subset of objects that maximizes the total value, while maintaining the overall volume below a pre-set threshold (which is termed the *capacity* of the knapsack).

In our context, the main goal of the system is to estimate the current energy usage, and to predict its short-term trend in order to check that it is compatible

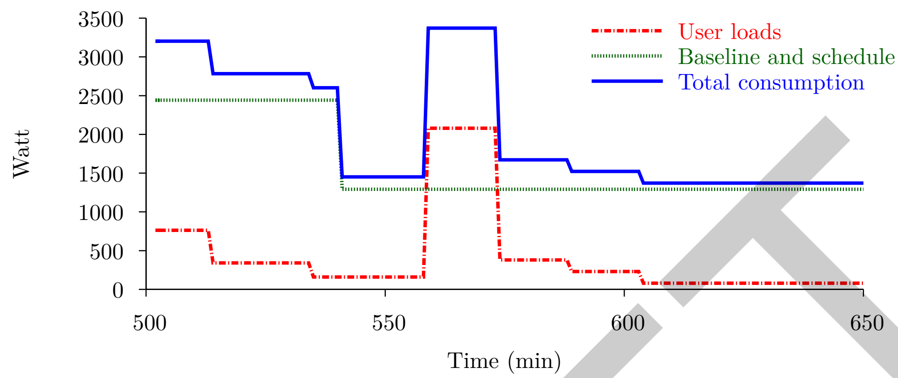

Figure 5: Example of a breakdown of energy demand in terms of baseline, schedulable and user loads.

with the activity the user is performing; the system then tries to rearrange loads generated by the appliances, in order to avoid exceeding a pre-set threshold for the overall demand, while satisfying user requirements, and completing the planned tasks. Our underlying assumption is that the total energy consumption can be parted into two main components, namely the predictable consumption and the unpredictable one. The first component includes all the baseline loads simulated by the EDS module, as well as the schedulable loads due to the user plan; both components are intrinsically predictable. On the other hand, user loads generated by the current activities are hardly predictable, unless we restrict ourselves to a short term prediction by taking advantage of user activity recognition. Figure 5 shows a sample of a breakdown of energy demand in our scenario. The constraint represented by the pre-set threshold is thus further narrowed by an amount corresponding to the estimate of the consumption due to user loads; hence, we focus on optimizing predictable loads energy consumption by rearranging them in order to meet the more restrictive threshold.

This functionality is provided by the block named EDO in Figure 1, which represents a software module accepting the following inputs: the current estimated energy consumption, the predicted user activities, and the user plan. When the predicted short-term energy use exceeds the pre-set threshold, the system attempts to select the minimum necessary amount of devices to be temporarily turned off so as to satisfy the energy use constraint, while respecting the provided priorities. Once the predicted load falls within the limit, the system attempts to restore the device; another option is to look for another device to turn off in order to trade for the reactivation of the old one.

In order to take user requirements into account, we follow the proposals of [26, 27], and provide a *slack* time for each baseline appliance; this piece of information is used to prevent the optimizer from turning a device on and off too quickly, which would cause a degradation in the overall performance, or even a possible failure. In the end, the deactivation time for a device is minimized, causing as little inconvenience as possible for the users.

In our proposal, time is split into fixed-size slices; for each slice, the system selects the optimal set of devices to turn on in order to meet the energy consumption constraint, and match the user plan as closely as possible; as already mentioned, the optimal selection of devices is formulated as a knapsack problem, to be solved at each time slice.

The capacity of the knapsack is defined as:

$$E_k = E_T - \hat{E}_U, \tag{1}$$

where $\hat{E}_U$ is the estimated maximum energy of user load component for the considered time slice, and E_T is the pre-set consumption threshold. The maximum consumption value is stored for each activity instance, together with the timestamps of its beginning and end time; for each new instance, a probability distribution parameterized over the initial time of the activity is recomputed; a similar approach is used for the baseline estimation, based on a whole day prediction.

The function to be maximized is expressed as:

$$\sum_{i=1}^{n} v_i x_i, \tag{2}$$

where the summation is taken over all the appliances generating baseline and schedulable loads. The integer variable x_i is defined as:

$$x_i = \begin{cases} 1 & \text{if the device is turned ON} \\ 0 & \text{if the device is turned OFF} \end{cases} \tag{3}$$

The coefficient v_i indicates the priority of the task, corresponding to the user-defined one for the schedulable appliances and to a function of the slack value for the baseline loads.

The constraint to meet is:

$$\sum_{i=1}^{n} E_i x_i \leq E_k, \tag{4}$$

where E_i is the consumption of the device, according to its consumption model.

5. System Evaluation

In order to assess the performance of our system we considered two reference scenarios, according to activity recognition or energy saving tasks; in the former case, we analyzed events generated by sensors deployed in a smart home environment, where each sequence of triggers was labeled according to the activities performed by the user, whereas for the latter we assumed that the system was able to control a predefined set of appliances, and we simulated an energy consumption demand, according to a realistic energy use profile. In particular, three public datasets were used to measure the accuracy of the system: *adlnormal* [37], and *aruba* [38] (both from the CASAS project), and the one we named *kast* [39] from the *Context Awareness in Residence for Elders* (CARE) project. All datasets are annotated, i.e. their sensor trigger sequences are labeled with the activity the user was performing in correspondence to that portion of data: the so-called *actual* activities; however, the three datasets are very different with respect to the set of employed sensors and to the way the data was collected; their descriptions are reported in Table 1.

Table 1: The datasets used for testing the system.

Dataset	Features	Activities	Sensors
adlnormal	20 users (one at a time), about 6,000 sensor readings, 100 activity instances	5 activities (*Telephone use*, *Hand Washing*, *Meal Preparation*, *Eating and Medication Use*, *Cleaning*)	motion sensors, analog sensors for monitoring water and stove burner use, as well as software sensors (VOIP), and contact switch sensors on phone book, cooking pot and medicine container
aruba	1 user, about 6,000 sensor readings (out of 1,600,000 total), 120 activity instances (6,471 total)	11 activities: *Meal Preparation*, *Relax*, *Eating*, *Work*, *Sleeping*, *Wash Dishes*, *Bed to Toilet*, *Enter Home*, *Leave Home*, *Housekeeping*, *Resperate*	binary sensors: motion sensors and door closure sensors (temperature sensors were also present, but they were not used by the proposed system)
kast	1 user, 2,120 sensor readings and 245 activity instances spanning 28 days	7 activities (characterized by different time duration and different frequency): *Leave house*, *Toileting*, *Showering*, *Sleeping*, *Preparing breakfast*, *Preparing dinner* and *Preparing a beverage*	14 binary sensors deployed in the house, placed on doors, cupboards, refrigerator and a toilet flush.

5.1. Evaluation of the MDL event encoder

The MDL encoder represents the core of the preprocessing step, and its main goal is to reduce the "uncertainty" inherently present in data so that the subsequent modules of the system may focus only on the most significant information. Thanks to the new encoding, dissimilarities among event patterns are magnified, so that they get scattered throughout the ideal representation space, which ultimately results in more easily distinguishable activities.

The effects of user activities are observable by the system only in terms of the effects they produce on the environment, so an activity might be abstractly modeled as a stochastic source of sensor triggers; in our vision, we more specifically regard an activity as a source of alphabet elements (the n-grams selected by the MDL encoder) and compute its emission probability. Telling different activities apart is only possible if each element can be associated to the correct source; this task becomes more manageable as the source emission probability distributions are most different from each other.

The assessment of this module was thus carried out by comparing the statistical properties of the different activities, in terms of probability distribution of their basic elements. We purposely disregarded temporal information at this step, as it does not carry additional significant information in this context. Different instances of the same activity can be very dissimilar in terms of their temporal unfolding, depending on how specific users perform them, but the relevant information content consists in their respective subtask composition, regardless of the exact duration and consequentiality[1]. The statistical properties of an activity, thought of as a stochastic source, might reasonably be considered invariant and distinctive of the activity itself.

We chose the Hellinger distance as a measure of dissimilarity between different activities [40]. This is a f-divergence measure, which quantifies the difference between two probability distributions $P^{(i)}$ and $P^{(j)}$:

$$h(P^{(i)}, P^{(j)}) = \frac{1}{\sqrt{2}} \left\| \sqrt{P^{(i)}} - \sqrt{P^{(j)}} \right\|_2, \tag{5}$$

where:

$$\sqrt{P^{(i)}} = \left(\sqrt{p_1^{(i)}}, \sqrt{p_2^{(i)}}, \cdots, \sqrt{p_m^{(i)}}, \cdots \sqrt{p_n^{(i)}} \right) \tag{6}$$

is a unit vector in 2-norm, $p_m^{(i)}$ is the probability that the i^{th} activity 'emits' the m^{th} symbol, and n is the cardinality of the encoding alphabet.

By definition, the Hellinger distance is symmetric and satisfies the triangle inequality, so it is a proper distance, which induces a metric space. We use this metric space to get a quality measure of our preprocessing; namely, if we compute the Hellinger distance for every pair of activities, both before and after preprocessing, we would expect that a useful encoding imply a larger distance on average in the latter case.

[1] For example, *Cooking* will likely involve a set of tasks such as opening the cupboard, grabbing a pot, and switching on the stove burner, but their duration and exact sequence may vary among different instances of this activity.

Table 2: Comparison of Hellinger distance with original triggers and after MDL encoding.

	adlnormal		***kasteren***		***aruba***	
	Original	**Encoded**	**Original**	**Encoded**	**Original**	**Encoded**
Mean	0.6116	0.6931	0.8360	0.8533	0.9007	0.9024
Max	0.9423	0.9793	1	1	1	1
Min	0.2483	0.1668	0.3190	0.3076	0.1666	0.1600

Table 3: Confusion matrix of the difference of Hellinger distance between original triggers and MDL encoding.

	Telephone use	***Hand Washing***	***Meal Preparation***	***Eating/med. Use***	***Cleaning***
Telephone use	0	0.1572	0.0370	0.1465	0.0996
Hand Washing	0.1572	0	0.1151	0.1407	-0.0815
Meal Preparation	0.0370	0.1151	0	0.0528	0.1102
Eating/med. Use	0.1465	0.1407	0.0528	0	0.0375
Cleaning	0.0996	-0.0815	0.1102	0.0375	0

In our test, we do get an improvement in Hellinger distance for every dataset, with an increase as high as 8% as compared to the original representation in the case of *adlnormal*, supporting our intuition. For this dataset, the average Hellinger distance computed between all the ten pairs of the five considered activities is 0.6116, when only activation triggers are considered as suggested by [31]; after our MDL encoding, it increases up to 0.6931. Table 2 summarizes the results of our tests. Table 3 shows how much the Hellinger distance matrix differs, for each couple of activities of *adlnormal*, with and without applying the MDL encoding; element (i, j) of this matrix is the difference of the Hellinger distance between activity i and activity j in the two cases; obviously, it is a strictly triangular matrix. The obtained results show a significant improve-

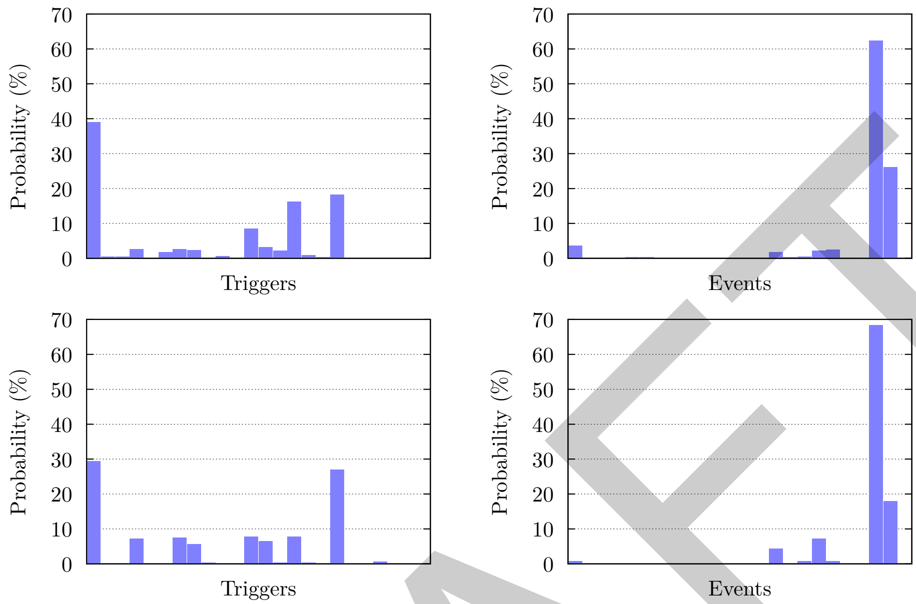

Figure 6: Hellinger distance: difference between original triggers (left column) and MDL encoding (right column) for *Cleaning* (top row) and *Hand Washing* (bottom row).

ment, in terms of a higher Hellinger distance, for most of the activity description dissimilarities. The original encoding outperformed our MDL encoding only for the (*Hand Washing*, *Cleaning*) pair, probably due to the extreme similarity of the two activities. Figure 6 shows a detailed comparison of the two activities, in terms of distribution of triggers and alphabet elements; the *Hand Washing* activity clearly shows how the MDL encoder succeeds in compressing the statistical description of the activity; however, this eventually resulted in an increased similarity to the *Cleaning* activity. On the other hand, the most significant improvement was obtained for the (*Hand Washing*, *Telephone use*) pair, likely because encoding is able to emphasize the difference in terms of the predominant set of subtasks; these two activities indeed involve very different sensor sets, as they are carried out in different areas of the house and with different tools.

Similar results were obtained for the other considered datasets, with an

Table 4: Evaluation of q_1 and q_2 for different values of k.

	k	**q1**	**q2**
adlnormal	**5**	0.80	0.80
	7	1	0.91
aruba	**11**	0.28	1
	22	0.36	1
kast	**7**	0.86	1
	11	1	0.95

overall increase in the Hellinger distance, except for those pairs composed by only a very reduced set of significant elements.

5.2. Testing the activity recognition modules

We tested the performance of our k-medoids algorithm in producing meaningful classes of activities, in terms of the goodness of its clustering. To this end, we used the same metrics as in [31], namely:

- q_1: the ability to identify activities, computed as the ratio between the number of actual labels assigned to the *discovered* cluster representatives, and the total number of *actual* activities;

- q_2: the ability to assign correct labels to the extracted patterns with respect to actual activities, computed as the fraction of patterns actually belonging to the activity assigned to the cluster medoid, per each cluster.

In order to assess the influence of the chosen number of clusters (k) on our metrics, we performed a set of experiments as reported in [8]. We initially set k to the number of actual activities for each dataset, and then increased it. The results show that q_1 is very sensitive to k, and higher values of k cause an increase in q_1, as is particularly evident in *adlnormal* (see Table 4). The worst performance is obtained on *aruba*, due to the presence of many unlabelled triggers, reflecting the fact that actual activities poorly correspond to the user's normal life; this is also highlighted by the results for q_2 on the same dataset,

Table 5: A summary of the best results in activity recognition accuracy.

	k	**N**	**w**	**Actual**	**Discovered**
adlnormal	**7**	**4**	**12**	0.95	0.98
aruba	**11**	**6**	**3**	0.66	0.92
kast	**11**	**6**	**3**	0.55	0.97

which show that when the cluster does represent an actual activity, its patterns are labeled in the correct way. For the other datasets, q_2 confirms the results from q_1, and shows good performance on accuracy in classification.

We also assessed the accuracy of the HMM-based activity recognizer, with respect to discovered and actual activities. We aimed at computing the best values for setting the HMM parameters, i.e. the number of hidden states (N), and the size of the sliding window (w); to this end, we used a grid search, with $N \in [3; 15]$ and $w \in [3; 15]$, and computed the accuracy of the system at each point in the grid. We conducted two separate tests, aimed at recognizing accuracy of actual and discovered activities, respectively. Results for the best configuration of parameters with respect to actual activities are shown in Table 5, where the corresponding value for discovered activities is also shown. As expected, better results are achieved for actual activities in *adlnormal*, due to better correspondence between actual and discovered activities. The achieved accuracy is very high, confirming the capacity of our method of building reliable models. The results obtained for the *aruba* and *kast* show that our recognition system is able to create models of discovered activities with no assumption regarding the particular scenario. On the other hand, results on actual activities in these datasets suffer from the poor correspondence between discovered activities and actual activities. The setting for parameters N and w is also dependent on the specific dataset; such values need to be carefully chosen with respect to the data at hand, as they basically represent how different activity definitions are mirrored into the corresponding datasets.

Table 6: Correspondence between activities and the relative appliances.

Activity	**Appliances**
Meal Preparation	`Hobs`, `Stove`, `Microwave oven`, `Kettle`
Cleaning	`Vacuum cleaner`
Eating and Medication Use	`Coffeemaker`
Telephone use	`Lamp`
Hand Washing	`Instantaneous Water Heater`

5.3. Energy consumption optimization by peak load shaving

The lack of a sufficiently rich dataset to measure the effects of real-time user activity recognition on energy usage optimization motivated us to generate synthetic data to assess the performance of our EDO block. Our simulation takes advantage of the typical home appliance profiles, as documented in [35]; additional profiles were generated by using the models proposed in [22]. With such information fed into the EDS block, we were able to compute two energy demand curves in order to compare the performance obtained without the intervention of the EDO block with the one resulting from the inclusion of the energy optimizer. In the former case, the EDS was tuned to simulate a typical domestic usage, considering the actual sequence of activities in a fashion similar to [22, 23]. In the latter case, the optimized energy demand is computed by following the indications of the EDO block about toggling the appliances on and off.

Our tests were conducted by considering the *adlnormal* dataset so as to build an energy profile for each of the five tracked activities (namely, *Telephone use*, *Hand Washing*, *Meal Preparation*, *Eating and Medication Use*, and *Cleaning*). Table 6 indicates the subset of appliances involved in their execution. *Adlnormal* activities often span a short interval, hence the simulation makes use of time slices of appropriate length (2 minutes in our case).

As regards the user plan, we assumed that the appliances whose use was suitable to be scheduled were those reported in the first row of Table 7, while the baseline loads were simulated according to the devices reported in the second

Table 7: List of appliances associated to schedulable and baseline loads.

Load category	**Appliances**
Schedulable	`Washing machine`, `Tumble dryer`, `Dishwasher`
Baseline	`Refrigerator`, `Electric heating`, `Freezer`, `Air conditioner`, `Circulation Pump`

row. We also coded the dependencies between different tasks, where applicable; for instance, the use of the `Tumble dryer` is only admissible after the `Washing machine` task has been completed. In our experiments, the pre-set threshold for limiting peaks in energy demand was set to 3 kW; in order to solve the knapsack problem, the capacity may thus be recomputed at each time step by subtracting the predicted energy demand due to the user activity from such threshold (see Def. 1 on p. 21).

Figure 7 shows some significant examples of the outcome of the peak shaving algorithm. The reported charts are representative of the cases where energy demand was successfully maintained below the pre-set threshold; overall, our system managed to reduce the number of unacceptable peaks by about 30%, on our synthetic data; however, there were circumstances when the excess of energy demand could not be avoided due, for instance, to the combined effect of the user plan and the requirements of the current activity or, much less frequently, to a wrong prediction of the activity recognition module.

The two charts shown in the topmost row illustrate a common situation when the operating time of some baseline appliances is delayed until the overall load falls below the given threshold (see the shadowed are in the charts).

A different behavior is shown in the middle row, where the original loads are presumably due to appliances for which a considerable *slack* time was provided; the final effect is that the system is allowed to give priority to the energy constraint at the expense of slightly bending the requirements of the users, who experience a delay in the services offered by baseline appliances; basically over-threshold loads are immediately switched off, and their re-activation (if any) falls beyond the currently shown window.

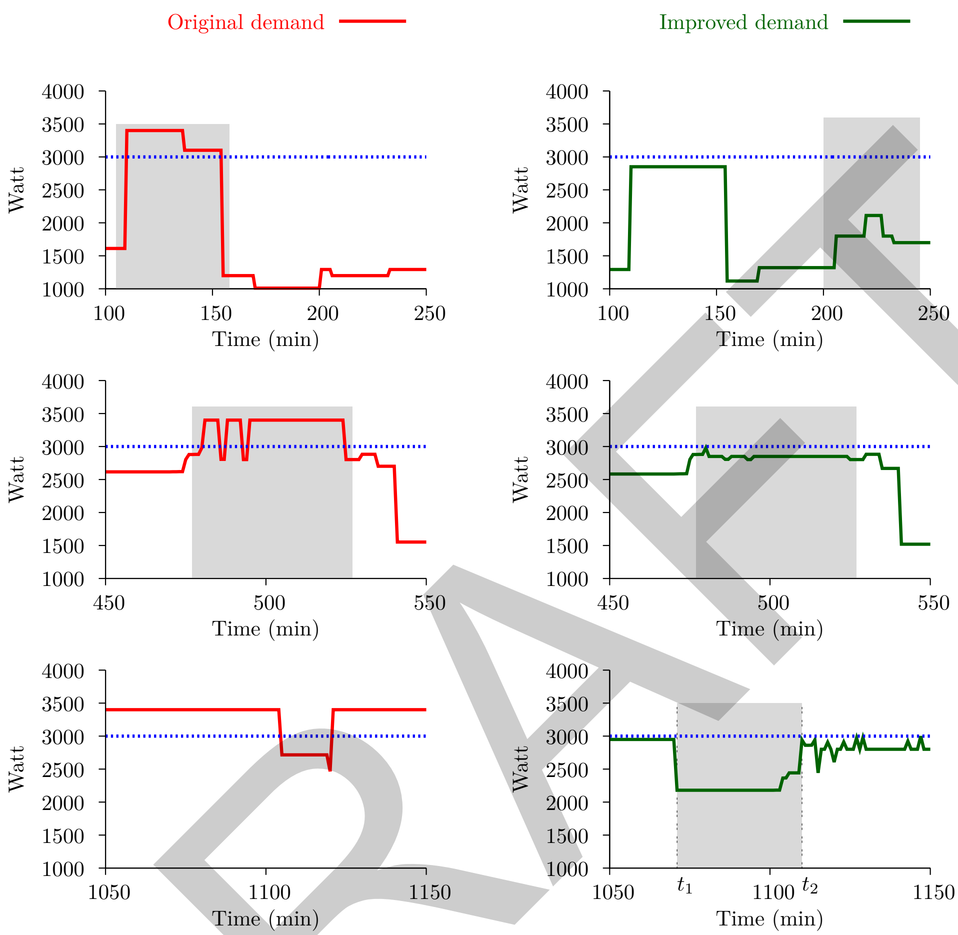

Figure 7: Comparison of original energy demand, and the one obtained after applying our approach.

Finally, the last row shows a specific instance of the action of the optimizer on schedulable loads. Those are typically characterized by long activation times; for instance, one such load is present for about 50 minutes (from time 1050 to 1100 in the left chart). The right chart shows the action of the optimizer resulting in an immediate re-scheduling of the critical loads, which are temporarily removed; this is followed by an additional deactivation at time t_1, and some loads appearing again at time t_2. However, it is evident from the chart that, at time t_2, some loads start "competing" for re-activation, thus producing an oscillation in the optimizer behavior; this is likely due to their relatively similar priorities, or simply to an intrinsic “bursty” consumption (which is typical of some appliances, such as `kettle`).

6. Conclusion and On-going Research

The proposed system implements user activity recognition in order to optimize energy demand of appliances in a smart home by shaving off peaks. User activities have a non negligible impact on energy consumption, however this relationship is hardly predictable due to the typically unsteady course of human habits. Our system builds upon one of our previous works [8], and describes how recursive structural activity models can be inferred, and user activities discovered, in order to predict short-term energy demand, and minimize peaks in energy consumption through ad-hoc appliance scheduling.

The provided results – based on public datasets and synthesized data about energy consumption – demonstrate that two new emerging research areas, namely user-side energy optimization and pervasive monitoring, can be regarded as complementary in the view of the creation of a unified approach.

Our aim was to maintain the system as open to generalization as possible; future developments might include the integration of user-side optimization with smart grids, thus allowing for improved energy saving strategies, possibly through user-centric energy cost policies. On-going research also involves improving models for appliances, in order to take into account for complex and more realistic functioning modes (beyond merely switching them on and off) so as to get fine-grained optimization. Improved energy consumption models will have to be tailored on the different modes, thus resulting into a better estimate of required energy. Moreover, it is conceivable to adapt our system to a scenario characterized by local renewable energy production in a smart grid system; for instance, if the smart home is equipped with renewable energy sources (e.g., domestic wind turbine, solar panels, etc), the focus of the system could shift to meeting the balance between supply and demand, minimizing the electricity drawn from the grid. In the common scenario of an electric price plan that offers discounted energy prices during off-peak hours and higher prices during specific on-peak hours, the system might concentrate high demand period in forecasted high production by renewable sources, and take out electricity in low cost energy period.

Acronyms

AmI	*Ambient Intelligence*
BECM	*Building Energy and Comfort Management*
CASAS	*Center for Advanced Studies in Adaptive Systems*
EDS	*Energy Demand Simulator*
EDO	*Energy Demand Optimizer*
AMB	*Activity Model Builder*
ART	*Activity Recognizer and Tracker*
SAIL	*String mAtching with wIldcards and Length constraints*
DVSM	*Discontinuous Varied-order Sequential Miner*
MDL	*Minimum Description Length*
HMM	*hidden Markov models*
KP	*knapsack optimization problem*
CARE	*Context Awareness in Residence for Elders*

References

[1] B. Bose, Global warming: Energy, environmental pollution, and the impact of power electronics, Industrial Electronics Magazine, IEEE 4 (1) (2010) 6–17. doi:10.1109/MIE.2010.935860.

[2] J. Hollander, J. Harris, Improving Energy Efficiency in Buildings: Progress and Problems, American Council for an Energy-Efficient Economy, 1980.

[3] D. Landsberg, R. Stewart, Improving Energy Efficiency in Buildings, State University of New York Press, 1980.

[4] A. De Paola, G. Lo Re, M. Morana, M. Ortolani, An intelligent system for energy efficiency in a complex of buildings, in: Prodeedings of the 2nd IFIP Conference on Sustainable Internet and ICT for Sustainability (SustainIT), 2012, pp. 1–5.

[5] A. De Paola, S. Gaglio, G. Lo Re, M. Ortolani, Sensor9k: A testbed for designing and experimenting with WSN-based ambient intelligence applications, Pervasive and Mobile Computing 8 (3) (2011) 448–466. doi:10.1016/j.pmcj.2011.02.006.

[6] D. Peri, Body area networks and healthcare, in: S. Gaglio, G. Lo Re (Eds.), Advances onto the Internet of Things, Vol. 260 of Advances in Intelligent Systems and Computing, Springer International Publishing, 2014, pp. 301–310. doi:10.1007/978-3-319-03992-3_21.

[7] J. V. Paatero, P. D. Lund, A model for generating household electricity load profiles, International Journal of Energy Research 30 (5) (2006) 273–290. doi:10.1002/er.1136.

[8] P. Cottone, S. Gaglio, G. Lo Re, M. Ortolani, User activity recognition for energy saving in smart homes, in: Prodeedings of the 3rd IFIP Conference on Sustainable Internet and ICT for Sustainability (SustainIT), 2013, pp. 1–9. doi:10.1109/SustainIT.2013.6685196.

[9] M. Erol-Kantarci, H. Mouftah, Wireless sensor networks for domestic energy management in smart grids, in: Proceedings of the 25th IEEE Biennial Symposium on Communications (QBSC), 2010, pp. 63–66. doi:10.1109/BSC.2010.5473004.

[10] M. Milenkovic, O. Amft, Recognizing energy-related activities using sensors commonly installed in office buildings, in: Proceedings of the 3rd International Conference on Sustainable Energy Information Technology (SEIT-2013), Vol. 19, 2013, pp. 669 – 677. doi:10.1016/j.procs.2013.06.089.

[11] O. Sianaki, O. Hussain, A. Tabesh, A knapsack problem approach for achieving efficient energy consumption in smart grid for end-users' life style, in: Proceeding of the IEEE Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply (CITRES), 2010, pp. 159–164. doi:10.1109/CITRES.2010.5619873.

[12] A. De Paola, S. Gaglio, G. Lo Re, M. Ortolani, Multi-sensor fusion through adaptive bayesian networks, in: AI*IA 2011: Artificial Intelligence Around Man and Beyond, Vol. 6934 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2011, pp. 360–371. doi:10.1007/978-3-642-23954-0_33.

[13] H. Peng, P. Wu, J. Zhu, J. Zhang, Helix: Unsupervised grammar induction for structured activity recognition, in: Proceedings of the 11th IEEE International Conference on Data Mining (ICDM), 2011, pp. 1194–1199. doi:10.1109/ICDM.2011.74.

[14] R. Hamid, S. Maddi, A. Johnson, A. Bobick, I. Essa, C. Isbell, A novel sequence representation for unsupervised analysis of human activities, Artificial Intelligence 173 (14) (2009) 1221–1244. doi:10.1016/j.artint.2009.05.002.

[15] C. Chen, D. J. Cook, A. S. Crandall, The user side of sustainability: Modeling behavior and energy usage in the home, Pervasive and Mobile Computing 9 (1) (2013) 161–175, special Section: Pervasive Sustainability. doi:10.1016/j.pmcj.2012.10.004.

[16] Y. Kim, T. Schmid, Z. M. Charbiwala, M. B. Srivastava, Viridiscope: Design and implementation of a fine grained power monitoring system for

homes, in: Proceedings of the 11th International Conference on Ubiquitous Computing, Ubicomp '09, ACM, New York, NY, USA, 2009, pp. 245–254. doi:10.1145/1620545.1620582.

[17] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, J. Albrecht, Smart*: An open data set and tools for enabling research in sustainable homes, in: Proceedings of the 2nd KDD Workshop on Data Mining Applications in Sustainability (SustKDD), Beijing, China, 2012, p. 6.

[18] L. G. Swan, V. I. Ugursal, Modeling of end-use energy consumption in the residential sector: A review of modeling techniques, Renewable and Sustainable Energy Reviews 13 (8) (2009) 1819 – 1835. doi:10.1016/j.rser.2008.09.033.

[19] A. De Paola, G. Lo Re, F. Milazzo, M. Ortolani, Predictive models for energy saving in wireless sensor networks, in: Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011, pp. 1–6. doi:10.1109/WoWMoM.2011.5986204.

[20] A. Lalomia, G. Lo Re, M. Ortolani, A hybrid framework for soft real-time wsn simulation, in: Proceedings of the 13th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications (DS-RT), IEEE, 2009, pp. 201–207. doi:10.1109/DS-RT.2009.30.

[21] M. Muratori, M. C. Roberts, R. Sioshansi, V. Marano, G. Rizzoni, A highly resolved modeling technique to simulate residential power demand, Applied Energy 107 (2013) 465 – 473. doi:10.1016/j.apenergy.2013.02.057.

[22] I. Richardson, M. Thomson, D. Infield, C. Clifford, Domestic electricity use: A high-resolution energy demand model, Energy and Buildings 42 (10) (2010) 1878 – 1887. doi:10.1016/j.enbuild.2010.05.023.

[23] J. Widén, E. Wäckelgård, A high-resolution stochastic model of domestic activity patterns and electricity demand, Applied Energy 87 (6) (2010) 1880 – 1892. doi:10.1016/j.apenergy.2009.11.006.

[24] G. Costanzo, A. Kosek, G. Zhu, L. Ferrarini, M. Anjos, G. Savard, An experimental study on load-peak shaving in smart homes by means of online admission control, in: Proocedings of the 3rd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies (ISGT Europe), 2012, pp. 1–8. doi:10.1109/ISGTEurope.2012.6465658.

[25] G. Costanzo, J. Kheir, G. Zhu, Peak-load shaving in smart homes via online scheduling, in: Proceedings of the 20th IEEE International Symposium on Industrial Electronics (ISIE), 2011, pp. 1347–1352. doi:10.1109/ISIE.2011.5984354.

[26] S. Barker, A. Mishra, D. Irwin, P. Shenoy, J. Albrecht, Smartcap: Flattening peak electricity demand in smart homes, in: Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom), 2012, pp. 67–75. doi:10.1109/PerCom.2012.6199851.

[27] Y. Xu, D. Irwin, P. Shenoy, Incentivizing advanced load scheduling in smart homes, in: Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings, BuildSys'13, ACM, New York, NY, USA, 2013, pp. 1–8. doi:10.1145/2528282.2528292.

[28] M. Magnusson, Discovering hidden time patterns in behavior: T-patterns and their detection, Behavior Research Methods, Instruments, & Computers 32 (1) (2000) 93–110. doi:10.3758/BF03200792.

[29] Chen, Gong, Wu, Xindong, Zhu, Xingquan, Arslan, Abdullah, He, Yu, Efficient string matching with wildcards and length constraints, Knowledge and Information Systems 10 (4) (2006) 399–419. doi:10.1007/s10115-006-0016-8.

[30] J. Rissanen, Minimum description length principle, in: C. Sammut, G. Webb (Eds.), Encyclopedia of Machine Learning, Springer US, 2010, pp. 666–668. doi:10.1007/978-0-387-30164-8_540.

[31] P. Rashidi, D. J. Cook, L. B. Holder, M. S. Edgecombe, Discovering Activities to Recognize and Track in a Smart Environment, IEEE Transactions on Knowledge and Data Engineering 23 (2011) 527–539. doi:10.1109/TKDE.2010.148.

[32] V. I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, Soviet Physics Doklady 10 (8) (1966) 707–710.

[33] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Third Edition, Academic Press, Inc., Orlando, FL, USA, 2006.

[34] L. Rabiner, B. Juang, An introduction to hidden markov models, IEEE ASSP Magazine 3 (1) (1986) 4–16. doi:10.1109/MASSP.1986.1165342.

[35] R. Stamminger, U. B. L. Fakultät, Synergy Potential of Smart Domestic Appliances in Renewable Energy Systems, Schriftenreihe der Haushaltstechnik Bonn, Shaker, 2009.

[36] Martello, Silvano, P. Toth, Knapsack problems: algorithms and computer implementations, John Wiley & Sons, Inc., New York, NY, USA, 1990.

[37] D. J. Cook, M. Schmitter-Edgecombe, Assessing the quality of activities in a smart environment, Methods of Information in Medicine 48 (5) (209) 480–485. doi:10.3414/ME0592.

[38] D. J. Cook, Learning setting-generalized activity models for smart spaces, IEEE Intelligent Systems 27 (1) (2012) 32–38. doi:10.1109/MIS.2010.112.

[39] T. van Kasteren, A. Noulas, G. Englebienne, B. Kröse, Accurate activity recognition in a home setting, in: Proceedings of the 10th international conference on Ubiquitous computing, UbiComp '08, ACM, New York, NY, USA, 2008, pp. 1–9. doi:10.1145/1409635.1409637.

[40] E. Hellinger, Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen., J. Reine Angew. Math. 136 (1909) 210–271.