Machine Learning for Smart Building Applications: Review and Taxonomy

DJAMEL DJENOURI, ACM Senior Member, CERIST Research Center, Algeria
ROUFAIDA LAIDI, CERIST, Algeria and Ecole Superieur d'Informatique, Algeria
YOUCEF DJENOURI, Southern Denmark University, Denmark
ILANGKO BALASINGHAM, Norwegian University of Science and Technology (NTNU), Norway

The use of machine learning (ML) in smart building applications is reviewed in this paper. We split existing solutions into two main classes, occupant-centric vs. energy/devices centric. The first class groups solutions that use ML for aspects related to the occupants, including (1) occupancy estimation and identification, (2) activity recognition, and (3) estimating preferences and behavior. The second class groups solutions that use ML to estimate aspects related either to energy or devices. They are divided into three categories, (1) energy profiling and demand estimation, (2) appliances profiling and fault detection, and (3) inference on sensors. Solutions in each category are presented, discussed and compared, as well as open perspectives and research trends. Compared to related state-of-the-art survey papers, the contribution herein is to provide a comprehensive and holistic review from the ML perspectives rather than architectural and technical aspects of existing building management systems. This is by considering all types of ML tools, buildings, and several categories of applications, and by structuring the taxonomy accordingly. The paper ends with a summary discussion of the presented works, with focus on lessons learned, challenges, open and future directions of research in this field.

CCS Concepts: • **Computing methodologies** → **Machine learning**; • **Computer systems organization** → **Embedded and cyber-physical systems**; • **General and reference** → **Empirical studies**; • **Hardware** → *Sensor applications and deployments*;

Additional Key Words and Phrases: Smart buildings, Smart cities, Internet of Things

ACM Reference Format:
Djamel Djenouri, Roufaida Laidi, Youcef Djenouri, and Ilangko Balasingham. 2018. Machine Learning for Smart Building Applications: Review and Taxonomy. 1, 1 (August 2018), 42 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Recent advances in mobile computing, wireless sensing and communication technologies, consumer electronics have modernized our cities and living environments. Buildings, roads, and vehicles are now empowered with a variety of smart sensors and objects that are interconnected via machine-to-machine communication protocols, accessible via Internet, to form what is known as the Internet

Authors' addresses: Djamel Djenouri, ACM Senior Member, CERIST Research Center, Rue des freres aissou, Ben-Aknoun, Algiers, Algeria, ddjenouri@acm.org; Roufaida Laidi, CERIST, Rue des freres aissou, Ben-Aknoun, Algiers, Algeria, Ecole Superieur d'Informatique, Algiers, Algeria, rlaidi@cerist.dz; Youcef Djenouri, Southern Denmark University, Odense, Denmark, djenouri@imada.sdu.dk; Ilangko Balasingham, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, ilangkob@iet.ntnu.no.

XXXX-XXXX/2018/8-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

of things (IoT). This makes our cities and premises smarter, with augmented capacities through cyber-physical systems where machines and humans interact and act on the environment. The term "smart environments" covers a variety of domains such as smart transportation, infrastructure and resource management, precision agriculture, smart buildings, etc. In this paper, we focus on smart buildings. IoT solutions have revolutionized the energy management systems in buildings and endowed them with the capability to dynamically adapt automation and energy supply, which contributes to reduce the wasteful power drain due to suboptimal management and irresponsible human behaviors. A variety of IoT-based automation systems are already in the market [Hossain et al. 2017], e.g., Smartthings, Twine, Vera, openHAB, Ninjablocks, Microsoft Lab of Things, etc. [Perera et al. 2015].

While the most appealing benefit of smart building technologies is this revolution in the building management systems (BMS), they are also significantly influencing other sectors such as retailing (smart shopping centres), health care (in smart hospitals and homes), security and safety (intrusion/anomaly detection systems), etc. The term *smart buildings* in this paper should be interpreted holistically, as potential applications to all categories of buildings are considered, including commercial (e.g., offices, retailing), residential (smart homes), and public buildings (hospitals, schools, etc.). Earlier researches on the use of the information and communication technologies (ICT), and particularly IoT solutions, for smart buildings (mostly for energy management) have been concentrated on solutions for monitoring, dynamic automation and real-time actuation, e.g., [Caicedo and Pandharipande 2015], [Reppa et al. 2015], etc. Technical and scientific challenges that have been dealt with include presence-adaptive and daylight harvesting when using dimmable luminaries [Caicedo and Pandharipande 2015], sensor node deployment for occupancy detection and optimal sensing coverage [Ouadjaout et al. 2016], [Fanti et al. 2018], for optimal communication in indoor environments [Bagaa et al. 2017], indoor localization [Xiao et al. 2016], time synchronization [Djenouri and Bagaa 2016], etc. In most of these solutions, traditional optimization models have been used such as linear/convex programming, dynamic programming, meta-heuristics, game theory, stochastic models. However, studies show that solutions limited to dynamic automation are insufficient [Molnar et al. 2015]. For example, a survey has been carried out over 11 United Kingdom householders for one year time period [Hargreavesn et al. 2013], and the results reveal that beyond a certain level of energy consumption by the householders (up to couple of kilowatts), the latter quickly adopt the reported consumption as normal and often find no motivation to use the monitoring systems that might compromise their comfort. Considering users' comfort is thus vital to motivate for the use of these technologies and to get the consumer in the energy saving loop. The current trend in smart building applications is to explore approaches derived from machine learning for inferring the users' preferences, behavior, comfort, etc., and then accordingly pursue the targeted optimization in accordance with the users' perspectives (e.g., their preferences and comfort).

Machine learning (ML) is training computers to *learn* from data collected through past experience. Learning is the most appropriate alternative in cases where it is not possible to directly write programs to solve problems, i.e., when the solution is not a priori known, but can only be developed using data or experience. This is typical in problems where human expertise does not exist, or when it is difficult to express it. Traditional domains where ML has largely been used include speech/face recognizing, language processing, spam filtering, etc. In the context of buildings, fundamental problems such as predicting occupants behavior and preferences, forecasting energy demand and peak periods, etc., are difficult to be solved with traditional programming but potential solutions can only be learned from data. The use of ML tools for emerging domains such as smart buildings is amongst the research trends that has recently been attracting the research communities in several disciplines, including computer science and electrical engineering, power engineering, civil engineering and architecture. This paper provides a comprehensive and holistic review on works related to smart buildings from the perspective of ML methods that are used and the fundamental problems dealt with.

Figure 1 presents a general framework that is conceptually shared by most solutions presented throughout the paper. It is composed of four steps: (1) Data collection, where data is harvested from different sources, including environmental sources such as sensors, archive sources such as events log databases, or other data sources. The data collected from heterogeneous sources are stored in a single database. (2) Preprocessing of data stored in the previous step before processing using ML techniques. This step includes i) data enrichment, e.g., adding statistical data such as the mean value of the samples, standard deviation, etc. ii) Data cleaning, e.g., of textual data. NLP (Natural Language Processing) techniques [Yi et al. 2003] is typical example of cleaning methods that could be used in this stage. iii) Selection of the appropriate features from all the data, which depends on the task used in the learning step. PCA (Principal Component Analysis) [Hyvarinen 1999] is one of the most used methods for feature selection and dimensionality reduction. iv) Normalization of the data, which is needed for some ML operators such as similarity computation for clustering, or propagation in the neural network (these ML concepts will be explained in the next section). At the end of this step, input data is created for ML approaches. (3) Learning step where the ML techniques are used to learn functions and models. (4) Interpretation of the learning from the previous step, which largely depends upon the application used.

A list of acronyms used in the remaining of the paper is given in table 1.

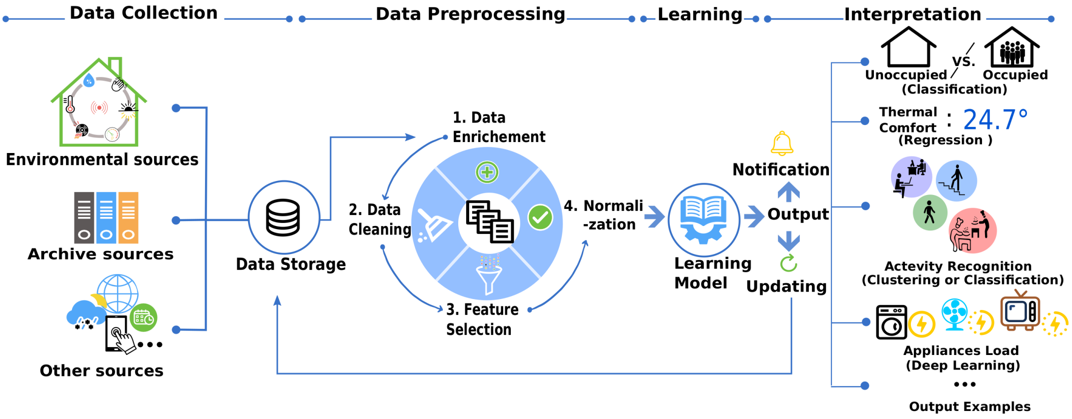

Fig. 1. General framework of ML solutions

1.1 Previous Review Papers

This section summarizes the relevant literature in comparison with the current paper. It clarifies the differences and positions the contribution. ML concepts referred to in this part are presented in Sec. 1.2. Many review papers dealt with the use of ICT for smart cities and environments. [Khatoun and Zeadally 2016] discussed concepts related to architectural issues and challenges of ICT in smart city applications. [Perera et al. 2017] reviewed the use of fog computing technologies in smart city applications. ICT solutions for energy management in smart buildings have largely been reviewed. [Kazmi et al. 2014] reported on solutions based on wireless sensor networks for BMS. [Labeodan et al. 2015] concentrated on occupancy measurement in offices. [Lazarova Molnar et al. 2017] considered Fault Detection and Diagnosis (FDD) in BMS and introduced the use of crowdsourcing

Table 1. Acronyms most used in the paper

ABC: artificial bee colony	ANN: Artificial Neural Networks	ARIMA: Auto Regressive Integrated Moving Average
AUC: Area Under the Curve	BN: Bayesian Network	CPLC: Convex Piecewise Linear Classifier
DAG: Directed Acyclic Graph	DL:Deep Learning	DNN: deep neural networks
DT: Decision Tree	ELN: elastic net	FP, FN : False Positive, False Negative
FSD: fast state decoding	GA: Genetic Algorithm	GBM: Gradient Boosting Machines
GMM: Gaussian Mixture Models	GNB: Gaussian Naive Bayes	HMM: hidden Markov model
HVAC: Heating Ventilation Air Conditioning	K-NN: K-Nearest Neighbor	LLC: Locality-constrained Linear Coding
LR: Logistic Regression	LDA: Linear Discriminant Analysis	MAE: Mean Absolute Error
MAPE: Mean Absolute Percentage Error	MaxAE: Max Absolute Error	MIQP: Mixed Integer Quadratic Program
MLC: Multi-Label Classification	MLR: Multiple Linear Regression	MMPP: Markov Modulated Poisson Process
MPC: Model Predictive Control	PCA: Principle Component Analysis	PMV: Predicted Mean Vote
QDA: quadratic discriminant analysis	R: Correlation Coefficient	RBF: Radial Basis Function
RBM: Restricted Boltzmann Machine	RF: Random Forests	RT: Regression Tree
RMSE: Root Mean Square Error	SD: Standard Deviation	SOA: Service Oriented Architecture
SRC: Sparse Representation Classification	SVM: Support Vector Machine	SVR: Support Vector Regression
WPCA: Weighted Principle Component Analysis	XGB: Extreme Gradient Boosting trees	

for FDD in buildings, with an implementation of the concept in a mobile application. The authors highlighted the potentiality of using the data collected through crowdsourcing by ML algorithms. The use of ML tools for smart environments has also been considered in some recent surveys. [Ota et al. 2017] provided a comprehensive survey on the use of DL in the mobile multimedia domain. The authors presented existing software frameworks that enable the implementation of deep network architectures without coding from scratch, as well as hardware acceleration technologies that allow to overcome the computation complexity of the DL algorithms. Motivated by the fact that the learning components in mobile applications should be run on an external hardware, the authors focused on the inference models for DNNs and approaches reducing the computation complexity. They summarized the use of DNNs in, (1) health care applications, e.g., estimating the amount of calories from food through smart phone pictures and contextual information, human activity monitoring to prevent chronic diseases, estimating the level of stress by voice analysis, etc., (2) security, e.g., detection of malware on mobile devices, biometry, (3) ambient intelligence, e.g., recognizing places of interest and localizing garbage, (4) translation and speech recognition.

The most related review papers on ML and smart buildings are [Millera et al. 2017], [B.Yildiz. et al. 2017], [Karvigha et al. 2017], [De Paola et al. 2014]. However, they are all specific to a category of applications and/or ML tools. [Millera et al. 2017] reviewed works that consider the application of unsupervised ML approaches to non-residential buildings for smart-metering, portfolio analysis [1], anomaly detection, operation and control optimization. The authors also analyzed the reviewed literature from the perspective of publication venues and authors' domains and disciplines, and they noticed high multi-disciplinarity. [B.Yildiz. et al. 2017] reviewed works that use regression models for electricity load forecasting in commercial buildings and discussed the relevant applications. They provided empirical comparison between some models using dataset from real buildings. The authors concludes that the regression models performed fairly well in comparison to other more advanced ML models. [Karvigha et al. 2017] considered user preferences for BMS and discussed how preferences for level of automation vary by contexts, individuals'

[1] of a large group of buildings that share the geographical area, managed or owned by the same entity

personalities and demographic characteristics. The contexts investigated in this study includes rescheduling an energy consumption activity, activity-based appliance state control, and light control. Collected data from 250 respondents have been analyzed using a logistic regression-based approach.

[De Paola et al. 2014] provided a structured presentation of the existing literature on intelligent BMS while supporting a vision that transcends the well-established smart home domain into what is called, the "ambient Intelligence paradigm". The authors discussed the main energy saving approaches in buildings, the requirements of a BMS, and some proposed architectures. Methodologies for occupancy detection are also presented, as well as those for learning the user's preferences. This paper has some limited overlapping with our survey in discussing occupancy detection, activity recognition and user preferences, but it focuses on technological and architectural aspects rather than ML aspects. Further, it was narrower in the sense of being limited to the energy management aspects, and wider in describing all the aspects related to energy management (basically technological and architectural aspects). Although application of ML tools was not amongst the motivations, some solutions using basic ML tools have been shallowly described, and the authors have reported some futuristic vision on the integration of such tools in BMS. Posterior to [De Paola et al. 2014], many ML-based solutions have been proposed in that direction for intelligent BMS, but also for other building related applications as well. Our work is motivated by the lack of a comprehensive review on these solutions in the current literature.

Table 2 summarizes some features of the survey papers discussed in this section. Compared to those reviewing ML aspects in related domains, the current paper is the first that does it holistically, while all the other works are limited to only some categories of ML (LR, DL, unsupervised learning, etc.), or even to some category of buildings (commercial, non-residential). Compared to those dealing with smart buildings, it differs by focusing on categories of applications targeted by ML tools (occupancy, behavior, preferences, energy profiling, etc.) rather than the architectural and technological aspects of BMS. The only survey dealing with smart buildings (in a holistic way) and ML as well is [Karvigha et al. 2017], but it is limited to analyzing solutions dealing with user preferences and using LR.

Table 2. Related review papers.

Reference	**Reviewing ML**	**Topic**	**Purpose**
[Khatoun and Zeadally 2016]	no	smart city	ICT architectural issues
[Kazmi et al. 2014]	no	smart buildings	WSN for BMS
[Lazarova Molnar et al. 2017]	no	smart buildings	crowdsourcing for fault detection
[Labeodan et al. 2015]	no	smart buildings	occupancy
[De Paola et al. 2014]	no	smart buildings	ICT architectures for BMS
[Ota et al. 2017]	DL	mobile multimedia	soft/hard frameworks+applications
[Millera et al. 2017]	unsupervised learning	non-residential buildings	energy analysis and optimization
[B.Yildiz. et al. 2017]	regression models	commercial buildings	electrical load forecasting
[Karvigha et al. 2017]	logistic regression	smart buildings	user preferences
current paper	holistic	smart buildings	several categories of applications and ML tools

1.2 Background and Basic ML Concepts

In general, ML may be used either (1) for data analytic and deriving knowledge from past experiences, (2) for predictive modeling and applying the knowledge to predict new instances, or (3) for decision making. Two major categories of ML algorithms may be distinguished, *supervised learning* vs.

unsupervised learning. In supervised learning, the task is to learn the mapping from a set of features as input to their appropriate output through labeled dataset, i.e., the output's correct value for each datum is known and provided by a supervisor. However, in unsupervised learning there is no supervision but only "unlabeled" input data. Unsupervised learning searches for possible regularities (or for a structure) in the input space that causes certain patterns more frequently occurring than others. There is also a category that falls between the supervised and unsupervised learning, called *semi-supervised* learning, where generally a small amount of labeled data is used jointly with a large amount of unlabelled data. *Active learning* is a typical example of semi-supervised learning, where the user is interactively queried to obtain the desired outputs at new data points.

Classification is a typical example of supervised learning problems. Given a finite set of known categories (classes), the problem is to find a solution that allows to identify (predict) to which category (categories) a new observation belongs to. Instances with know category (labeled data) are used as a training set input. Classification is not limited to assigning a single label to every observation, but multiple labels may be assigned to observations. This is known as *multi-label classification* as a generalization of multiclass classification. *Regression* is another example of supervised learning that is related to prediction, i.e., predicting a numerical value from a continuous set by learning a numeric function that relates the output to the inputs. *Linear regression* is the simplest and most used form of regression. The most common supervised learning tools are ANN, SVM, LR, DT, RF, (we refer to Table 1 for these acronyms, and to the appendex for their definitions).

Clustering is a typical example of unsupervised learning, where the aim is to group observations such that observations in the same group are more similar to one another than to those in other clusters. Data instances are clustered by maximizing intraclass similarity and minimizing the similarity between different classes. Contrary to classification, the set of clusters are not known a priori but are driven by the training data. The goal is to assign labels according to the features of objects. There are three main categories of clustering algorithms: (1) centroid-based, (2) density-based, and (3) hierarchal. Some references group the first two categories in the same class called *partitional* [Jain et al. 1999]. In centroid-based clustering models, the similarity is measured according to the closeness of a data point to the centroid of the clusters and the number of clusters is introduced as an input, which makes the prior knowledge of the dataset important. These models process iterative searching for local optima. One of the canonical algorithms that is used in many solutions presented in this paper is K-means. In density-based clustering, dense regions of data points separated by low-density regions are grouped as clusters. A region is considered dense if it gathers a certain number of data points within a defined radius. The knowledge of the number of clusters is not needed in density-based clustering and the resulted clusters can be of any shape. *DBSCAN* is a typical density-based clustering algorithm [Loh and Park 2014]. Finally, hierarchal clustering aims to build a hierarchy of clusters. There are two main types for hierarchical clustering, (1) divisive (the "top down" approach) where all observations start in one cluster and the hierarchy is built by recursive splitting, (2) agglomerative (the "bottom up" approach) where every element in the dataset is a cluster and pairs of clusters are merged (or agglomerated).

Association Rules Mining (ARM) is another category of unsupervised learning. It is a method that investigates relations between variables in the dataset based on some measures of interest. Association rules are mainly evaluated using two metrics. (1) Support that measures the absolute frequency, (2) and confidence that measures the correlative frequency.

Reinforcement Learning (RL) represents another category of ML that is based on learning by trial-and-error. *RL* inspires from the human learning process, i.e., by perception from the environment. It uses agents that try in each action to maximize the cumulative reward. Rewards or punishments are perceived in the environment. RL focuses on the goal and learns (over several steps) a complex objective. It differs from both supervised and unsupervised learning in the interpretation of inputs.

RL does not deal with labeling data but on deciding the next action based on short and long-term rewards that the input provides. *RL* is usually combined with DL in decision making, e.g.,. this combination allows to achieve human-level performance in multiple games[et al. 2015].

In some cases, some features may not be informative or repeat a redundant information from other features. *Dimensionality reduction* is important to reduce inputs' patterns and thus reducing the model's complexity and faster the training. This is done using *Features selection and extraction* techniques. Features selection, or variable/attribute selection, consists of automatic selection of the relevant attributes in a dataset. On the other hand, feature extraction combines attributes to create new informative ones. Principal component analysis (PCA) is a common feature extraction technique. In cases where datasets are not available for a particular problem, *transfer learning (inductive transfer)* is used. This is a method of using knowledge gained while solving a problem as a starting point to learn about a different related problem. Detailed explanations of the techniques mentioned in this section can be found in [Alpaydin 2014; Mitchell 1997; Trevor Hastie 2013], while brief definitions are given in the appendix.

1.3 Taxonomy and Paper Organization

Fig. 2 summarizes the general taxonomy of the different works presented in this paper, which are basically divided into two classes, occupant-centric vs. energy/devices centric. The former groups solutions that use ML for services focused on the occupant, including (1) occupancy estimation and identification, (2) activity recognition, (3) estimating preferences and behavior. The second class includes solutions where the ML approaches are used to estimate aspects related either to energy or devices (including appliances and sensors). They are divided into, (1) energy profiling and demand estimation, (2) appliances profiling and fault detection, and (3) inference on sensors.

Following this taxonomy, the remainder of this paper is organized as follows. Solutions that use ML for aspects related to occupant-centric are presented in Sec. 2, followed by those related to energy/devices in Sec. 3. Sec. 2.1 presents works related to occupancy detection/estimation. Sec. 2.2 presents works on recognition of occupants' activity, while Sec. 2.3 presents those for recognition of preferences and comfort. Works dealing with the use of ML for energy profiling and demand estimation are studied in Sec. 3.1, those for appliance profiling and fault detection in 3.2, and those using ML to infer information on sensors are presented in Sec. 3.3. Sec. 4 provides a summary discussion on the presented works, with focus on challenges and future directions. And finally, Sec. 5 concludes the paper.

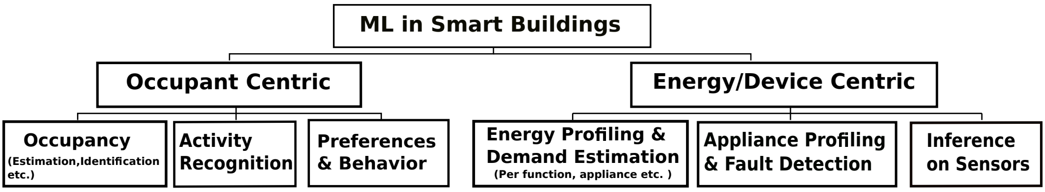
ML in Smart Buildings
Occupant Centric
Energy/Device Centric
Occupancy
(Estimation,Identification etc.)
Activity Recognition
Preferences & Behavior
Energy Profiling & Demand Estimation
(Per function, appliance etc.)
Appliance Profiling & Fault Detection
Inference on Sensors

Fig. 2. Global taxonomy

2 OCCUPANT-CENTRIC SOLUTIONS

2.1 Occupancy

2.1.1 Overview and Problem Statement. Solutions dealing with occupancy of premises in buildings will be presented in this part. This ranges from binary inference on the presence of occupants, to providing advanced estimation (the number, gender, etc.). Earlier works on occupancy monitoring have been focusing on the use of wireless sensing technologies to detect/track occupants in real time. More advanced solutions are not limited to realtime detection but consider the use of information coming from sensors to estimate future occupancy and/or features related to occupancy using ML tools. Those solutions are presented in this section. Contrary to some contexts in smart cities and environments that evolve open/public spaces (roads, streets, etc.), privacy preservation is generally required in many spaces in buildings (offices, meeting rooms, residential spaces, etc.). Therefore, none-invasive sensing technologies (PIR, ultra-sonic, etc.) are more appropriate than the use of cameras or microphones. Only solutions based on such technologies are considered herein.

2.1.2 State-of-the-art. [Khan et al. 2014] developed the bespoke platform that integrates several types of sensors including PIR, acoustic noise (audio level recording without sound sequence), humidity and light, which have been organized in an ad hoc mesh network. Sensor fusion techniques have been applied to combine environmental data (harvested from the sensors) with contextual data (information about meetings schedule and computer activities). A hierarchical analysis method with different levels of granularity have been used, based on standard statistical classifiers that can integrate potential uncertain contextual information (meeting schedule, computer usage, etc.). Three granularities of occupancy estimation have been considered, (1) binary occupancy, (2) categorical, and (3) exact number, where each granularity corresponds to a level in the analysis. The result at each level is used to improve the occupancy estimates at the next levels. Classification is performed at every level, and then the corresponding posterior probabilities are computed for every occupancy density prediction and employed as *additional feature sets* in the next level. The sets of features are calculated at each level using the recurrent formula: $\vec{f}_z^{\,h} = P^{h-1}(\vec{f}^{\,h-1} = z)$, where $\vec{f}_z^{\,h}$ is the new feature added in the level, h, of each object in the class, z, at the previous level $h-1$. $P^{h-1}(\vec{f}^{\,h-1} = z)$ is the posterior probability that the set of features, $\vec{f}^{\,h-1}$, takes the class, z, in the level $h-1$. The hierarchical classification problem has been dealt with using KNN (setting K to 3), SVM, and a standard grid search procedure. For illustration, let us consider the first level (a binary classification), where the KNN classifier is used to predict whether the current state of the building is considered as occupant or not. Historical data is represented by the set of relevant features of the previous states and its corresponding class (occupant or not occupant), and the current state is described by the current features. The similarity between the current state and each previous state is computed, then the three closest states are extracted, and the most frequent class in these states is selected and assigned to the current state of the building. The proposed solution has been evaluated in real-world deployment using a large commercial building with offices, meeting rooms, bathrooms, stairs, hallways, etc., while considering both high-traffic area (large open spaces) and low-traffic area (meeting rooms). The authors used data records of 14 days, and camera pictures taken every 5 minutes to get ground truth on the exact number of occupants. The proposed hierarchical classifier has been compared with non-hierarchical estimation, and the results motivate the use of the former. They showed that the performance of the hierarchical classifier reaches 99% of accuracy, which was higher than the non-hierarchical classifier that does not exceed 86%.

[Sangogboye et al. 2016] considered binary occupancy estimation in rooms with MLC, SVM, and using data from motion sensors that reported the collected data through the KNX protocol [Hersent et al. 2012]. The day has been partitioned into subintervals of equal lengths (10min has been used in the implementation), and then each label (slot) took a binary value. Dataset from three sets of offices has been used, which comprises 77 days of the motion sensor sampled every 30sec. The prediction algorithm proposed in the paper has been compared with Preheat [Scott et al. 2011], where the KNN algorithm with Hamming distance is used to predict occupancy. The formulation with MLC enabled the use of its performance metrics, in particular the micro-averaged F-measure. The results revealed that the proposed approach outperforms Preheat in terms of F-measure.

[Ardakanian et al. 2016] considered approximate occupancy estimation to dynamically optimize HVAC management. They dealt with the occupancy estimation from coarse-grained measurements of sensors that are commonly available through the BMS. They investigated the application of non-intrusive techniques, i.e, techniques that do not require installation of additional sensors at specific locations. They used an existing HVAC system with single-pneumatic control sensors, air flow sensors, and reheat sensors, which are supposed to be largely available in building with state-of-the-art HVAC systems. Schedules at the zone scale might be derived using this occupancy approximation, which improves effectiveness of existing HVAC systems. The authors modelled the problem as a clustering problem and used agglomerative hierarchical clustering with *complete-linkage clustering*. This clustering defines the inter-cluster distance (separating two clusters) as the maximum distance that separates their members. Time series have been used to obtain an estimate on individual-zone occupancy by adapting the Canny detection algorithm [Cannny 1986]. In this adaptation, the occupancy data represented by time series is first mapped to the Gaussian distribution and then convolved with the first derivative Gaussian kernel. This allows to build local minimum and local maximum at each upward and downward edge points, which are used to identify the beginning and the end of the occupancy period. A binary vector is finally produced for the inferred occupancy (occupied vs. unoccupied) at each zone..

[Shih et al. 2016] considered occupancy monitoring in two operation modes: (1) detection of presence vs. (2) estimation of the number of occupants. They used a form of active physical sensing known as ultrasonic response estimation sensor, which is based on the processing of the superposition of the microphone-recorded reflections from a transmitted ultrasonic-signal. Both classification and regression problems have been solved. In the former, binary information on the room occupancy (occupied vs. unoccupied) has been inferred, while in the latter, the number of occupants in each room has been estimated. Within the ML component of the proposed platform called "AURES", the authors proposed an occupancy inferring algorithm based on multilevel classification that operates in two steps. In the first, Doppler shift based classifiers have been used for a binary classification. In the second step, two generic regression trees classifiers have been used in a semi-supervised way to estimate the number of occupants. The Weighted Principal Component Analysis (WPCA) has been used as preprocessing step for feature extraction. This reduces the training data that is required for the estimation of the number of occupants (trained regression model). The results showed no negligible error, notably for large rooms that reach as much as 10%. Further, the system requires the deployment of the sensing platform (AURES) in every room, as well as the labeling of training data.

[Soltanaghaei and Whitehouse 2016] proposed WalkSense, a solution that uses walkway-sensors to classify states of home occupancy. The occupancy detection in this solution relies on the fact that in wlakways, motion sensors are more reliable (as compared to occupancy zones). Therefore, motion sensors have been used and deployed by defining placement rules to optimally cover walkways and avoiding overlapping. Solutions for zone-occupancy state estimation have been proposed where the zones have been partitioned into active sensing zone (where the occupant performs his daily activities), outside walkway zones, and sleep walkway zone (the walkways for exiting home and entering the bedroom, respectively). Consequently, occupancy states are divided into "active", "away", and "sleep". Two strategies have then been proposed for occupancy states estimation. The first is the *Offline WalkSense* that uses historical data and defines a sleep (resp. away) interval to be the duration between a pair of two consecutive detections by the sensor at the sleep walkway (resp. the outside walkway). The second strategy is the *Online WalkSense* which identifies the different states in real time. The module starts in the active state, then if an event is detected by the sleep sensor, it switches to "conditional sleep". A classifier is then used to label the transitions. The module remains in the "conditional sleep" state until the classifier labels transition as *sleep*. The classification is repeated until a decision is made by the classifier or an event is detected by a sensor. Decision trees have been used for the classification. This has been justified by their inherent capability for handling combinations of mix types of data. The used features vary from temporal features, transition features, and mobility features. The same process is performed for detecting active and away states. WalkSense has been evaluated using sensorial data retrieved from six houses observed for a period of 350 days. The results revealed that WalkSense approach outperforms the baseline method (HMM-based), where it reaches 96% of accuracy in offline mode and 95% in online mode.

[Bales et al. 2016] dealt with a more advanced problem. They have not been limited to the detection of the occupants or counting them but were interested in determining their gender (gender classification). This has many potential applications, e.g, security in public buildings, retail sales and advertisement in commercial buildings. They used accelerometer sensors that have been mounted under-floor (the walking surface) for physical sensing. Supervised ML tools have been explored for classification including DT, boosted DT, SVM, and ANN. The authors reported high precision in detections, notably when using SVM that provided less error than ANN, e.g., 88% accuracy for gender classification with SVM vs. only 55% when using ANN. However, the technique used is highly intrusive and requires mounting a high number of sensors underfloor in the walking surface, which complicates installation in existing buildings.

[Khalil et al. 2016] went beyond gender classification and targeted identifying people by sensing their body shape and movement with ultra-sonic sensors. The principle of the proposed solution is to use the variation of the body when moving (in height and width) for feature extraction. From the signals detected upon walking events, the authors were able to extract seven features including the height and the width (maximum and average values), girth, hand-waist distance, and bounce. They used two feature-selection methods: (1) evaluate each feature alone then combine them in pairs and evaluate them, (2) a Recursive Feature Elimination (RFE) [Doak 1992] algorithm combined to PCA, which provides new high level features. Although the width and height are not unique features that identify people, the authors showed that by extracting features from their variations, identification becomes possible. The authors presented their solution as non-intrusive compared to those using cameras, microphones, or badges. However, they used ultrasonic sensors for measurement through a doorway. This might requires additional installation compared to solutions such as [Ardakanian et al. 2016]. In their solution, the selected features are fed to the DBSCAN ([Ester et al. 1996]) clustering algorithm. Occupant identification has numerous applications including customized services related to user comfort and preferences such as analysis of customer behavior in commercial settings, security and intrusion detection, health-care and elderly assistance, etc. However, training is needed for every individual, which might be constrained in applications involving high number of occupants.

2.1.3 Discussion. Different granularities of occupancy estimation have been considered by solutions presented in this section. Binary occupancy is the most elementary, which consists in predicting whether premise are occupied or not. It might be sufficient in some applications such as light control. Binary classification is the relevant ML category for this problem. A more advanced granularity is to estimate the exact number of occupants, generally using advanced regression techniques. This is very useful in applications such as personalized room allocation, but it is challenging and not easy to achieve. Category estimation comes as an intermediate granularity, where the aim is to generalize the binary estimation and define a finite set of categories, and then estimate to which category an observation belongs. This is practical and sufficient for many applications, e.g., HVAC control. Other solutions go beyond estimating occupancy or counting occupants, towards identifying occupants. While challenging, identifying occupants is required in many applications, e.g., gender classification is very useful in commercial buildings and has many applications such as customized advertisement in shopping centers. Some solutions even consider identifying the detected occupants through non-invasive sensor signals, which has applications in security, safety, and advanced health care systems. Most solutions in this category use SVM and decision trees as the baseline ML tool in their models. F-measure and different percentages of accuracy in estimation represent the most used metrics for evaluation. Depending the problem dealt with, the latter include percentage of correct reports (or symmetrically of errors) in presence detection, or that of accuracy in estimating the number of occupants, gender or person recognition. Some solutions also evaluated the energy saving when using their solution for occupancy estimation. Presence (motion) sensorial data is the most used in this category, mostly through PIR sensors. Contextual data has been used by solutions targeting the number of occupants estimation, e.g., meetings schedule and computer activity, while accelerometers and ultrasonic sensorial data has been used by those dealing with gender estimation and identification. Finally, note that DL methods have not been used in the solutions presented in this category. It might be interesting to explore such a tools for advanced estimations, i.e., identification. Table 4 (Appendix) compares the different solutions presented in this section.

2.2 Activity Recognition

2.2.1 Overview and Problem Statement. Solutions of this category do not deal with the estimation of occupancy in premises but with the current activity of the occupant (upon occupancy detection), which is vital for many applications such health care and elderly assistance. Several approaches are presented in the following through some canonical solutions. While most solutions target recognition of daily activities (tasks) such as cooking, eating, sleeping, watching TV, work wit computer, etc., some other solutions use more fine-grained definitions, such as differentiating the types of meals eaten, or associating timestamping of the tasks and considering each tuple (task, timestamp) as a separate activity for recognition. Some works targets less activities, e.g., receiving regular visits vs. irregular visits.

2.2.2 State-of-the-art. [Hossain et al. 2017] Proposed to use active learning and dynamic K-means for activity recognition in residential buildings. The use of active learning has been motivated by the variety of human activities in buildings and the underpinning uncertainty in sensing. This requires the provision of vast amount of labeled data for passive and supervised learning approaches to be effective, which is not always possible. The author proposed to first use a dynamic K-means to cluster the set of unlabelled data. The clusters have been incorporated with unseen activities, and the use of K-means has been adapted accordingly. In the proposed solution, the instances related to the unseen classes (activities) are considered as *outliers*. These outliers makes the clustering algorithm highly sensitive to the number of clusters (k). This has been dealt with by applying an incremental version of K-means in which k is increased at every iteration and the *overall clustering error* is recorded using an error function (J) on the set of clusters $\{C_1, C_2, ..., C_k\}$, which is defined for the set of centers, $\{g_1, g_2 ... g_k\}$, and, n objects, $\{x_1, x_2, ..., x_n\}$, as: $J = \sum_{j=1}^{k} \sum_{x_i \in C_j} ||x_i - g_j||^2$.

The clusters are then used to fetch the most informative data instances for query. An objective function has been formulated for this purpose, which is based on entropy, similarity coefficient measurement, and it is proportional to the distance between the clusters' points and the centers of the surrounding clusters. It is given by, $f_c(x) = argmax_x\{e_{\theta(x)} S_c^{(x)}\}$, where $e_{\theta(x)}$ is the entropy measurement that indicates the gain by putting x in its cluster (compared to the other clusters), and $S_c^{(x)}$ is the silhouette coefficient that represents the degree of importance of x in the cluster C. It is the difference between two ratios, (1) the ratio between the correlation of x and all elements in the cluster C, and (2) the ratio between the correlation of x and all elements in all the other clusters (excepts C).

The authors also proposed an *annotator selection Bayesian model* in presence of multiple-labelers with varying expertise. This is to deal with the challenging problem of assessing and validating the labels provided by annotators in absence of the ground-truth information. The proposed solutions have been tested in a single bedroom apartment with a kitchen and a living room. Data has been collected from 10 participants (each providing 24 hours of data collection), but with single participant at a time (i.e, no simultaneous presence of participants in the apartment). PIR sensors have been used to collect data, as well as object sensors (compasses and accelerometers) to provide data related to the usage and orientation of some objects (broom, laundry basket, phone, dustpan). Seven activities have been considered, including cooking, brooming, washing, cleaning, eating, sleeping and talking on the phone. The first four activities have been used for passive learning, while the activities have been left for the active learner to discover. The results showed that this approach with the proposed ground truth information model can detect unseen activities. Results also showed diverse performance in the accuracy of recognizing activities. Some activities have been recognized with an accuracy exceeding 80% (e.g., talking to the phone and brooming), while the accuracy was below 60% for other activities (e.g., eating and cooking).

[Chiang et al. 2017] explored the reuse of learned knowledge about occupants' activity recognition from an existing environment into another one (transfer learning). To reduce the complexity of knowledge transfer across different domains, the authors focused on the differences caused by the ambient sensors and the target domain. Only single-resident scenarios are considered, with similar activities of interest in the source and the target environments. They dealt with this as a classification problem and proposed a framework for knowledge transfer that uses standard SVM and RBF. Their contribution was to transfer the results of classification from a source building to a target building, which requires the matching of the different features of the two environments. Depending on the availability of labeled datasets, two scenarios have been considered: (1) when labeled datasets from both source and target environments are available, (2) when they are only available from the source environment and the information from the target environment is limited to background knowledge (sensor deployment information). In the former scenario, the aim is to take advantage of the data from the source environment to improve model learning in the target environment. In the second scenario, the aim is to help learn the activity models for the target environment from the source, and to use sensor readings of the target environment only for test. The first step is preprocessing, where data samples are created using the "start" time and "end" time of sensor events with a 30*sec* time interval. The second step is feature set reformulation. For the first scenario, a general approach with linear transformation using Shannon entropy and PCA has been proposed. The authors showed that this information-theory formulation guaranties no loss of information, and that each reformulated feature provides independent information (no redundancy). For the second scenario, the authors proposed a reformulation using profiles and defined four types of properties (object, location, sensor type, event), and then a set of values for each one (e.g, object has three possible values, microwave, TV, and door). In total, concatenating all possible values for the four properties results in a nine-tuple that defines the profile tuple. After reformulation, feature divergence evaluation is performed using Jensen-Shannon divergence (JSD). A graph matching algorithm has been used for feature set mapping. The authors proposed to reformulate the feature mapping into a graph matching problem, where a complete bipartite graph is defined and a weight value is assigned as the distance between features. Stable marriage algorithm [Gale and Shapley 1962] has been used to map every feature from the source to exactly one feature from the target (one-to-one mapping), with a divergence measure for every mapping. LIBSVM library [Chang and Lin 2011] has been used to train and test the SVM-based activity models. Ambient sensing dataset including a large variety of physical sensor information has been used to evaluate the proposed solution. The results confirm the proposed solution improves accuracy, but under the above mentioned assumptions. Recognizing fine-grained activities have been targeted in the experiment. For example, cooking has been split into different activities (preparing lunch, preparing breakfast, preparing dinner, etc.). 25 separate activities have been enumerated in total, but this is using as much as 70 sensors. The results showed accuracy of more than 70% (in most cases) when using knowledge transfer. A notable drawback of this solution is the fact that the proposed model completely ignores temporal dependence between activities; no feature with temporal information has been extracted.

[Nait Aicha et al. 2017] Considered the problem of modeling regular activity patterns in residence buildings to infer the presence of visitors, which is important in elderly assistance applications (e.g, where it is crucial to know whether the patient is alone or not). The authors modeled this as an unsupervised classification problem and proposed a method based on MMPP, which has been extended to enable incorporating *multiple-feature streams*. The proposed solution has been tested using a nine month dataset of sensor data including pressure (on the bed), toilet flush, motion, opening/closing of doors/cabinets. This dataset has been collected from two different apartments. In the first, the resident received daily visits from a caregiver, weekly visits from a cleaner, and occasional (non-regular) visits from his children. In the second, the resident got two visits per month from the cleaner and rarely other irregular visits. The results confirmed that the modified MMPP improves the performance over the standard MMPP in identifying irregular visits.

[Carolis et al. 2015] used a completely different approach to learn daily activities. They used process mining to learn (from annotated sensor data) the daily routines of the user that have been modeled as a classification problem. First-order logic learning has been explored, which the authors considered as a tool that incrementally adapts to the model and allows to express and learn complex conditions. The daily routines are considered as processes (sequence of events associated with actions) and their modeling is casted as process-mining. The authors used the "*WoMan*" process-mining system [Ferilli 2014] where tasks are modeled with a workflow that specifies how they can be composed to yield a "*valid processes*". This is based on the concept of "*case*" which is defined as a specific execution of actions in the workflow following an ordered set of steps. WoMan is incremental and gradually uses new cases (derived from log files of activity events) to update the workflow model. It allows to predict upcoming events upon detecting events that are consistent with the model. Both artificial and real dataset have been used to evaluate the solution, including high level data (tuples on labeled activities). The authors defined activities as the execution of a task in time, i.e timestamps are used in labeling data. That is, executing the same task at a different time is considered as a different activity, which gives a high number of activities (more than 6000 activities in the dataset of the simulation). The dataset was split into 10 folds, including 22 cases each. The normal 10-fold cross-validation procedure (using each time, nine folds for learning and one for testing) reached more than 99% average accuracy on predicting activities. However, the selected dataset was very simple, involving an elderly person who has occasionally been visited by her children (focused on a single person with routine activities).

[Alhamoud et al. 2015] proposed an approach to collect ground truth on everyday activities. They used a system that includes Plugwise sensors to measure power of individual appliances, and *Pikkerton* sensors to measure brightness, temperature, and motion, a *Raspberry Pi* as a gateway, a control server where ML model is implemented, and a smart phone with a customized GUI to allow the user report his/her current activity and thus provide ground truth information. The ML model is a form of K-means clustering that matches the reported activities with the sensor readings, where the reported data is dynamically clustered while setting the number of clusters as a tunable parameter. The latter would ideally match the number of reported activities. K-means has been adapted to the problem using the "validity based approach", which is based upon the validity metric defined as the ratio of the average intra-distance of all clusters (the average distance between points and their respective centroids) to the minimum inter-distance (the minimum distance between centroids.). The number of clusters that produces the smallest validity (after the occurrence of the first local maximum) is considered as the optimal number. Nine activities occurring in two apartments (deployments) have been considered in the experiment (for up to 82 days, and 62 days, respectively). The optimal number of clusters, i.e., corresponding to the number of activities specified by the user, has been found when testing with a dataset of three weeks for the first deployment, and of four weeks for the second. The results showed that by changing the size of the dataset, the optimal number of clusters varies. The authors justify this by the fact that similarities in the activities, e.g., "sleeping" and "not at home" feature very similar readings. This solution is then highly sensitive to the size of the data-set to be used for training, which is problematic. Rather than using K-means, it would be interesting for this problem to investigate other clustering algorithms that are not sensitive to the number of clusters.

[Zhu et al. 2015] did not rely on the deployment of physical sensors but only virtual sensing from smartphones (3D gyroscope and accelerometer data). This eliminates the need of deploying sensors but requires the occupant to permanently carry smart phones to perform activity recognition. The authors considered feature selection before feeding the data to ML algorithms. The proposed solution starts with feature extraction by segmenting the raw data (obtained with a $20Hz$ sampling frequency) using a $5sec$ sliding window interval, with 50% overlap. This results in a 100 samples per segment, from which a set of time and frequency domain features are extracted. These are typical statistical features used in pattern recognition (including mean, min, max values, etc.). All the features are then clustered using k-means as a dictionary to derive a codebook for LLC that is used as a feature selector. The use of LLC is motivated by the fact that it is highly probable that samples from the same class are neighbors in the feature space. Dictionary learning method is used to represent the features in a way that allows distinctiveness. In the proposed LLC approach, the features are factorized by solving an optimization problem with Fisher Linear Discriminant analysis (Fisher LDA), which has a complexity of the order $O(M^2)$ (where M is the targeted dimension of the selected features). The authors also proposed a simplified approximated LLC that has a lower computation complexity. It is based on the use of S nearest neighbors ($S \ll M$) to approximate the regularization terms, and it has a complexity at the order of $O(M + S^2)$. After feature selection, standard classifiers are applied using SVM, KNN, Kernel-Extreme Learning Machine, and SRC. Five activities have been considered, sitting down, getting up, being static, walking, and running. The results showed that the accuracy of the different classifiers varies from 70% to 95%, and the use of LLC with any classifier gives more accuracy (compared to the use of the classifier without feature selector) and provides about 5% of improvement.

2.2.3 Discussion. Different solutions and approaches for recognizing activities of occupants have been presented in this section. Some solutions use active learning and request the participator for labeling. This has the advantage of enabling the recognition of the variant human activities but requires the provision of vast amount of labeled data. Some solutions use knowledge transfer and apply knowledge acquired from a source environment to another target environment. This is useful to enrich the target environment with labeled data and improve the derived model. However, to enable accurate transfer, the two environments (source and destination) must be similar. Further, most activities are temporary dependent. Temporal dependence among features represented by data instances should thus be considered when applying transfer learning. Models that consider temporal dependence such as dynamic BN, as well as time-based models such as ARIMA might be explored for this purpose. The ML problem dealt with by the solutions presented in this category is either clustering or classification (depending on the considered activities), while SVM, KNN, and different variants of K-means represent the most used tools. The most common metrics that have been used include the percentage of accuracy in identifying activities, F-measure, and different distances on the obtained clusters when running the clustering algorithm (intra-cluster, inter-cluster, average, min and max values, etc.). Table 5 (appendix) summarizes the solution presented in this category.

2.3 User Preferences and Behavior

2.3.1 Overview and Problem Statement. Solutions presented in this section deal with estimating user preference in different ways. Most solutions focus on the thermal comfort, while some solutions also consider visual comfort. PMV has been used in earlier smart building automation systems to reach consensus on thermal preferences in public spaces. This approach is complex and fails to infer personalized comfort factors that vary from a person to another, and even for the same individual over time due to environmental and human related factors. Studies presented in this section explored data-driven approaches from an ML perspective and yielded solutions that capture the preferences either by receiving reports from users, i.e, data labeling, or by monitoring the past behavior of occupants to infer (in a transparent way) their preferences or settings that meet their comfort. Different forms of reporting interfaces have been used, where smartphone interfaces are the most convenient.

2.3.2 State-of-the-art. [Ghahramani et al. 2015] used BN to model and quantify personalized thermal comfort with *an online-learning*. They fitted comfort feeling ("warm and cool") dataset with probability distributions that they combined in a BN to define the global individual comfort. Four random variable have been considered, (1) UWC: Uncomfortable Warm Condition, (2) CC: Comfortable Condition, (3) UCC: Uncomfortable Cool Condition, and, (4) OC: Overall Comfort. The probability condition between OC and the three other variables is given by,

$$P(OC) = \frac{P(CC)}{\omega_1 P(UWC) + \omega_2 P(CC) + \omega_3 P(UCC)}, \text{where } \omega_1, \omega_2, \omega_3 \in [0, 1] \tag{1}$$

The Bayes optimal classifier aims at finding the temperature that maximizes $P(OC)$. The authors modeled UWC and UCC with half normal distribution while a complete normal distribution was considered for CC. The method of maximum-likelihood was used to estimate the distribution of these parameters. To reduce the need of estimating ω_1, ω_2, and ω_3, which requires a high number of samples, the authors considered them all equal. They also defined a hyper-parameter that represents the comfort threshold, and which is estimated using the training dataset. A "*Bayesian optimal classifier*" has been trained through online-learning to determine comfortable conditions. *Kolmogorov Smirnov* test has been used to determine comfort variations over time.

The authors compared the proposed solutions with some standard classification techniques (e.g., SVM, KNN, DT, LR) by applying them to thermal comfort data from an office. A user interface [Jazizadeh et al. 2013] has been used to collect thermal votes from occupants, and physical sensors for ambient conditions (humidity and temperature). The collected data are transferred to a database, and A "*survey-based participatory sensing approach*" [Jazizadeh et al. 2013] has been used to obtain comfort levels of individuals. Results showed superiority of the proposed classifier which reached an accuracy of about 70% in the tested scenarios. One advantage of the approach is that it enables the transformation of the comfort objectives, which prevents the *"pareto optimality problems"*.

[Zhou et al. 2015] dealt with thermal comfort modeling and attempted to bridge the gap between control and comfort learning. They considered augmenting the MPC framework with "*data-driven comfort requirement*" while adopting a "*learning for application*" scheme. Instead of user's comfort points, comfort zones are described with a CPLC that is used by the MPC for optimization. CPLC is a set of linear inequalities that makes classification by searching optimal configuration of multiple hyperplanes. It is a binary classification where elements inside the convex set represent the comfort zone. The intuition behind this is to improve the performance of box constrained approaches (usually used in HVAC systems), while generalizing and avoiding overfitting of nonlinear approaches. The authors considered that the cost of assigning an uncomfortable configuration as comfortable (false positive) is higher than the opposite assignment (false negative). For this, they proposed a *cost sensitive large margin* formulation where they weight false positive rates and false negative rates differently. These weights are introduced as hyper-parameters in the loss function of the classifier. The authors used *"online stochastic gradient descent"* with MIQP initialization to enable an incremental learning. The authors assume online voting to obtain user feedbacks. Numerical analysis using public dataset of thermal comfort preference has been performed to asses the proposed solutions in comparison with existing learning methods in an HVAC system. The used dataset includes data about air velocity, humidity radiant and air temperatures, as well as contextual information including physiological conditions such as metabolic-rate, clothing, and the expressed comfort sensations that have been obtained from online survey tools for comfort voting. The solution has been compared to similar methods from the literature including (1) 2v-SVM that uses Gaussian RBF and linear kernel, (2) one class SVM, (3) DNN, (4) AdaBoost, and 5) Lasso Logistic Regression. They used the testing error as a metric while varying the false positive weight (one of the hyper-parameters) from 1 to 9 and maintaining the false negative to 1. The solution outperformed the other solutions for weights above 4. The authors also carried out experiments on HVAC MPC and compared their solution to a box constrained approach in two weather configurations (cold dry and hot humid).

[Sarkar et al. 2016] used the intuition that *"a particular person can feel comfortable beyond the pre-defined set-points"*. They investigated on larger-range individual preferences for the sake of providing flexible, energy-saving, and dynamic operation of the controllers while ensuring the comfort of occupants. A smartphone application has been developed to enable users make reports and register data on-demand, which allows to learn thermal and visual (luminance) individual preferences. The standard of seven-point scale of ASHRAE[2] has been used (both for light and temperature). After studying data from multiple users, a thermal comfort has been represented with a Gaussian function, and the light preference with a Beta function. These functions use two hyper-parameters whose values differ from a user to another, (1) α for the Gaussian function, and (2) β for the Beta function. In this work, they have been derived based on the comfort indicators (collected using smartphone application) and using the *least square curve fitting*. This allows to generate the comfort functions from a set of limited points. Using an existing energy model for HVACs, the authors measured by simulation the yearly consumption of a building and showed that lower energy consumption may be achieved with set points based on individual comfort preferences, as compared to "fixed set points". This is while satisfying the individual preferences.

[Barbato et al. 2010] deployed a wireless sensor network to control home appliances following users' habits, and to automate the set up of operation parameters with a system that uses the past observed behavior to predict user preferences. The MobiWSN architecture [Laurucci et al. 2009] has been implemented and used in this work, where sensors provide information about light, temperature and presence. The authors studied user presence profiling (temperature and light)

[2] the American Society of Heating, Refrigerating and Air-Conditioning Engineers

using a simple data analysis clustering algorithm based on cross-correlation. The sensor network collected data for 24 hours over the monitoring period (up to a month), which was aggregated and processed to create the three types of profiles that represent users habits. The authors implemented their framework as a demo with a graphical Java application that emulates some visualized devices including lighting systems, air conditioners, WiFi access-points, TV. Further, the authors tested the solution by simulation on a five-room house for a 300-day period of pseudo-sequences of profiles for temperature, light, and daily presence. Three behavioral exceptions have been simulated to asses the proposed prediction algorithm: (1) *"exceptions spike"* (with 20 isolated exceptions), (2) *exceptions burst* (4 contiguous sequences) and; (3) *behavior variation* where the behavior changes twice a year. The results showed correct prediction of more than 85% for the first case, and more than 90% for the others. This allowed (for example) the home temperature manager to reduce up to 28% the working time of the cooling system, while keeping the user comfort (activating the cooling at the right moment prior to user's arrival). Lack of real experiment or simulation with realistic dataset represents the major drawbacks in this work. The authors justified this by the need of a long period of time, which was not possible to ensure.

[Antunes et al. 2013] considered learning thermal preferences of inhabitants from inferring behavior rules. They proposed "*APOLLO*", a platform that infers behavior rules through data collected with physical sensors, by using SOA, along with statistical and ML techniques. The authors' aim was to develop a platform that enables the addition/removal of any type of sensor/actuator without the need of manual reconfiguration. The use of SOA enables several independent services that are based on message passing communications. The platform has been initiated in a home automation scenario where it receives data related to energy consumption, temperature, and it defines the air conditioning system's temperature. The most frequent behavior of users has been used to make inferences and learn the inhabitants' preferences on temperature, which has been set through a remote controller while optimizing the energy consumption. A regression model using SVM has been trained to predict energy consumption based on the temperature and time period. The output of SVM model is used by a Genetic Algorithm (GA) [Banzhaf et al. 1998] that tries to optimize the energy consumption while achieving the preferred temperature. The GA's fitness function includes temperature, period and energy fitness. An anthologies dataset that expresses tendency between variables has been used in the experiments.

[Shoji et al. 2014] adapted a BN to a home energy management system for the purpose of learning the preference of the resident based on his/her behavior vs. electricity pricing and consumption changes, as well as performing appropriate operations of controllable appliances following the electricity price dynamics. The network allows to control energy appliances, e.g., battery energy storage systems (BESS), heat pump water heaters (HPWH), air conditioners (AC) etc., while considering the occupants' comfort. The authors assumed that the BN uses a DAG, and that its topology gives *"an intuitive grasp of the relationships among variables"*. The authors considered the fact that the state of an appliance depends on the environmental variables that affect the comfort and convenience. For example, the authors illustrated their BN with the description of the AC behavior for a "virtual resident". The state of the AC was probabilistically determined, jointly with respect to the temperature and the PMV as the two environmental variables that affect the AC's state. From the graph perspective, the state of the AC is a child node of PMV and temperature setting. To select the adequate probabilistic causal relationship between parent and child nodes, the authors used the "Bayesian Information Criterion", which yields short forms of data description. The authors did not rely on deployment of dedicated sensors but only used power consumption data from a smart meter. They used demographic datasets of 23 months of a continuous reading from a Japanese household databases. The duration of the training set was 14 months, and that of the test set was 9 months. The results showed very high accuracy for controlling many appliances such as AC and BESS, but they were prone to important variation in seasons with a high fluctuation in summer.

2.3.3 Discussion. Two approaches for collecting information about users comfort have been used; (1) the survey based approach where feedbacks about the degree of satisfaction are received from occupants, and (2) the behavior monitoring approach where the desired settings are deduced from past users configurations. Survey based approaches require interaction with occupants, which might be invasive and disturbing. Some solutions use regression ML tools, e.g., curve fitting, to complete the missing data and reduce the interactions with building's occupants. Other solutions correlate comfort levels with environmental variables, such as indoor air temperatures, light level, relative humidity, etc. This is by mapping the satisfaction levels received from occupants into ranges of environmental values, and then retrieving the range that maximizes the satisfaction. Ranges of values are considered instead of a single set point based on the perception that the occupants usually feel comfortable within continuous intervals. This also gives the possibility to satisfy multiple users that share the same space by finding intersections of their comfort zones. It also increases the potential of energy management, small adjustments (e.g., reducing temperature by $1°C$) might contribute in reducing the overall energy consumption. Classification methods such as CPLC and BN have been used. Some works also considered the variations in comfort levels communicated by users over time, which requires the use of online learning approaches to make the system adaptable to the changes. Solutions based on behavior monitoring used past users' interaction with the system to deduce the optimal setting for appliances, and through both supervised and unsupervised learning. Clustering, SVM and BN are the tools used in these solutions. Those solutions based on behavior monitoring are more adapted to individual houses where the occupants have more control on their appliances and can feed the system with their inputs, while those based on surveys are more appropriate for commercial buildings where the setting is generally executed by centralized controller. It is worth mentioning that the behavior monitoring approach is currently used in some commercial solutions, e.g. Nest learning thermostat, to define optimal thermal setting. To evaluate their solutions, some works used prediction estimation metrics, e.g., accuracy, specificity, and testing error to measure the efficiency of their prediction, while other focused on the energy saving potential as a performance metric. Solutions for user preferences and behavior are sketched in Table6 (Appendix), while Fig.3 presents a taxonomy for all the occupant-centric solutions.

3 ENERGY/DEVICE CENTRIC SOLUTIONS

3.1 Energy Profiling and Demand Estimation

3.1.1 Overview and Problem Statement. This class does not include all solutions that deal with energy optimization through ML (many solutions classified in other categories do so), but only those where the use of ML is related to energy profiling and estimation. We consider that the solution of [Barbato et al. 2010] presented previously also belongs to this category (in addition to the "user preferences and behavior" category) as it jointly deals with comfort and energy profiling. Profiles in solutions presented in this section include heating and cooling loads, electricity demand, energy profiles of individual appliances, etc. Including ML methods in energy profiling and demand prediction has several applications such as reducing energy cost in price-based control, fault detection and diagnosis, control optimization, etc.

3.1.2 State-of-the-art. [Zhang et al. 2016] divided the major home appliances loads into three classes: fixed, regulatable, deferrable. Based on this classification, they proposed a "decoupled demand response mechanism" to optimize energy management. For regulatable loads, the authors proposed a "learning-based demand response" strategy, in which they focused on HVACs. The

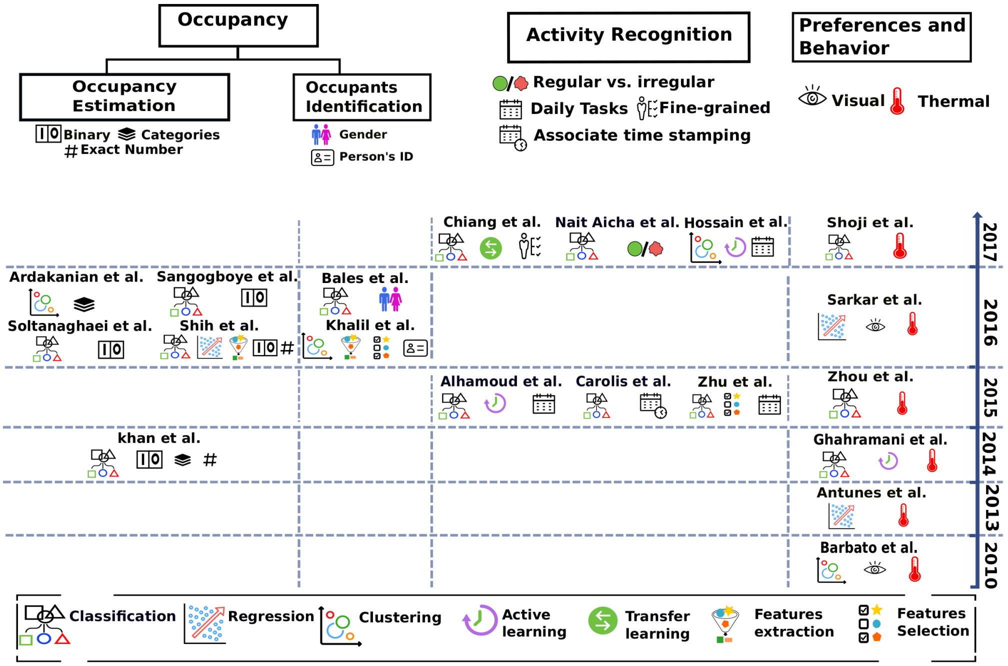

Fig. 3. Occupant-centric ML solutions

authors considered the optimization problem of minimizing the electricity cost function during a day (noted C) such as, $C = \sum_{i=1}^{24} P_i Q_i,$, where i stands for a one-hour timeslot during which P_i and Q_i represent the electricity price and the energy consumed by the HVAC, respectively. The function, Q_i, is learned through the training data by using ANN or regression with a 3rd order polynomial function. For instance, if we consider the function, Q_i, learned by, q_{nn}, and defined by, $Q_i = q_{nn}(T^l_{i+1}, T^l_i, T^O_i, \vec{w})$, then the learning is performed through a neural network that needs multiple-iterations to reach a stop criterion. The input data consists of temperature information including, (1) of the room at hour i, say T^l_i, (2) that in hour $i+1$, T^l_{i+1}, and (3) the outside temperature, T^O_i. The output of the network is the energy consumption, Q_i. The learning function, q_{nn}, aims to adjust the weights vector, $\vec{w}$, in a way that optimizes the energy consumption, Q_i. A "co-simulation system" has been developed to evaluate the proposed solutions. This system includes a simulator for house energy consumption, as well as a mechanism that allows for the simulation of a decoupled and learning-based demand response strategy. The building simulation software "eQUEST" has been used as a virtual testbed to simulate the energy consumption and the a house behavior [3]. Standard commercial building materials are used by the simulator (in its software library), as well as real-life dataset including information on solar conditions and weather [4]. A house featuring a *"generic floor plan for a two story* 2500 *square foot"* was used. The authors compared the proposed learning based solution with conventional demand response policies, and the results confirmed the improvement by the proposed solution in terms of energy consumption. The authors considered

[3] http://doe2.com/download/equest/eQ-v3-63_Introductory-Tutorial.pdf
[4] http://doe2.com/index_wth.html

a single occupant room for five consecutive days as the elementary cost and showed that the consumption cost is reduced from \$4.5 to \$3.64 when using the learning based solution .

In [Sonmez et al. 2015], the heating load (HL) and cooling load (CL) have been considered, and hybrid ML algorithms have been studied and compared by simulation. The methods are variants of KNN and ANN that combine GA and ABC heuristics. The weight of the inputs for KNN and the function to be activated for ANN are considered as hyper-parameters. For KNN, the heuristics (GA and ABC) have been used to assign different weights to the inputs, and for ANN to find the optimal activation function for each node among linear, sign, logistic, sin, tanh, and RBF functions. The authors used existing building architectural information (BAI) without any sensorial data. Eight parameters have been used as input to estimate HL and CL (the output parameters), including "*relative compactness, surface area, wall area, roof area, overall height, orientation, glazing area, and glazing area distribution*". MAE and SD of estimation errors have been used for comparison of the results obtained from the algorithms. 768 data samples have been used from the considered residential buildings, 576 for training and 192 for test and verification. Results confirmed that the proposed hybrid solutions provided better performance compared to the traditional methods (KNN, ANN). The authors justified this achievement by finding the impact of the input parameters on the target parameters (HL and CL), which was used in the estimation process.

[Fan et al. 2017] investigated the use of DL for predicting $24h$ ahead building cooling load profiles. They exploited the potential of DL in both supervised and unsupervised scenarios. As for the unsupervised learning, feature extraction has been carried out to extract meaningful features as model inputs, where four types of methods have been used. Three of the methods (engineering method, statistical method, structural feature extraction method) have been used for comparison with the DNN method proposed by the authors. This method has a symmetric architecture where the input and output layers have the same number of neurons. It tries to reconstruct the input while minimizing the error. It includes 5 layers with 48 neurons in the input and the output layers. This number of neurons represents the size of time series of $24h$, which has been collected at intervals of $30min$ that represents the building cooling load, the outdoor temperature, and the relative humidity. The number of neurons decreases between the input and the middle layer (2^{nd} hidden layer), where it reaches 4, which is the number of the extracted features. The activation of the middle layer is related to the extracted features. The authors used $tanh$ as the activation function in this DNN. They use a sliding window approach, where values from the past $24h$ are used for feature extraction, and the extracted features are then used to predict the load for one hour. This prediction is done by another DNN that is composed of two layers and using $ReLu$ as activation function. The one-hour predicted value is used with its previous $23h$ values to present the new input for the feature extraction for the next hour. This process is repeated until $24h$ cooling load is forecasted. MLR, ELN, RF, GBM, SVR, and XGB have been used for comparison with the proposed prediction approach using different feature sets that have been generated with the proposed feature extraction approach and three others. The authors used data from an educational building that consists of offices, classrooms and a computer data center, which has been collected during one year. The results showed that DL can enhance the performance of building cooling load prediction, especially when used in an unsupervised way.

[Mocanu et al. 2016] considered estimation of electricity demand in individual households and investigated two stochastic time-series DL models to predict energy consumption, (1) the conditional RBM and, (2) the factored conditional RBM. The motivation behind this choice is that the energy consumption might be viewed as a "time series problem". They adapted the architecture of these DL models by merging labels of the features and the style. The new configuration has been used to revisit the equations and the derivatives of the rules. They evaluated the proposed solution using a household benchmark dataset[5] that contains more than $2millions$ measurements gathered during a 47 months period. The period from the first to the third years was used for training, while the remaining period was used for the test. Information from smart meters have been used to get such data (without any additional sensors or intrusive plugs). Both "Aggregated active power" (global consumption) and energy sub metering were used. The authors compared the proposed solutions with traditional ML tools such as ANNs, SVMs, etc. The results showed that FCRBM outperforms all the other solutions for the energy prediction problem solved in this work.

[Chou and Ngo 2016] considered predicting real-time energy consumption of a building connected to smart grid and provided end users with forecast information that enables energy efficient measures during peak times. They developed an optimization-based ML system using a metaheuristic based time-series sliding window. It includes a variant of ARIMA models [Tan et al. 2010]; the seasonable ARIMA model (SARIMA), and the *"metaheuristic firefly algorithm-based least squares SVR (MetaFA-LSSVR)"* model. The proposed model (SARIMA) was fitted to linear components, while the nonlinear data components were captured with the MetaFA-LSSVR model. MetaFA is a metaheuristic used for the selection of parameters ("Hyperparameter") that are plugged into the LSSVR (a variant of SVR) for the training process where a least-squares cost function was used (in a dual space) to derive a linear set of equations. The aim of using MetaFA was to overcome the major drawback of the original LSSVR, whose accuracy highly depends on its Hyperparameter. The latter have then been optimized in the proposed solution through the use of MetaFA. To evaluate the proposed system, the authors used a "building experimental smart grid", from which they extracted realtime data. A family of five members occupied the building to provide learning dataset that has been collected during a four week-period, as well as the test dataset of up to 454 weeks after learning. The smart grid provided physically sensed data information including realtime power consumption data of appliances and electrical devices, temperature and humidity, as well as contextual information, e.g., appliance information, and alternatives of electricity saving, etc. The authors compared the proposed solution (integration of SARIMA and MetaFA-LSSVR) with the separate use of the different approaches integrated in the solution (SARIMA, LSSVR, and MetaFA-LSSVR). The results showed that the proposed solution provides improvement in the accuracy rate.

[Iyengar et al. 2016] analyzed building energy consumption dataset from a utility company. $14,836$ smart meters at a small city scale have been used. The consumption profiles have been clustered with K-means. The authors demonstrated how such analysis of smart meters at a large scale can help to extract very useful information, e.g., *"the impact of weather on energy usage, the correlation between the size and age of a building and its energy usage, the impact of increasing levels of renewable penetration"* etc. They showed that energy usage might be increased in "extreme weather" conditions, up to 36% in hot summer days, and to 11.5% in cold winter days. They diagnosed 700 residential buildings as "highly energy inefficient" due to their high energy demand variability. They also showed that solar penetration rates at a degree beyond 20% of demand increases the risks of generating useless extra energy, which affects the utility operations.

[Lange and Bergés 2016] considered disaggregation of appliances' energy consumption in buildings. They proposed a system that learns the constituent current waveforms from sequences of high frequency current cycles in an unsupervised way and looks up for the combinations of the appliances' subcomponents. An ANN was used to build the aggregate waveforms with a limited number of additive building blocks. These blocks are assumed to constitute "sub-appliance" waveforms. To minimize its training error, the ANN activates a subset of re-occurring patterns for every aggregate waveform. Online binary matrix factorization has been also performed to infer from the current signal, "additive sub-components". A DNN with binary activations has been used to

[5] URL: http://archive.ics.uci.edu/ml.

solve the unsupervised waveform classification problem. One of the solution's feature is the use of a general purpose hardware for data processing. Once the model is trained, realtime inference is performed off-the-shelf on the hardware that avoids transmitting large amounts of data (to a remote central repository). A public dataset has been used to evaluate the proposed method. A cross-validation technique has been used for the DNN hyper-parameters tuning and evaluation, which consists in splitting the dataset (1) randomly into equal training and testing sets, or (2) into non-overlapping sets. The hyper parameters considered in this study are the number of layers and the number of units. The results confirmed that the power consumption of individual appliances can be estimated with the proposed solution, but with different accuracies.

[Jain et al. 2016] explored "data-driven methods" to build "control-oriented" models that reduce power in buildings. They proposed a *"Data Predictive Control with RT (DPCRT)"* algorithm that uses data-driven models based on RT to implements *"finite receding horizon control"*. The method used in the construction of the predictive model replies on optimization (at the node level) of the "variable selection and splitting". The authors evaluated the method using a virtual testbed called "DoE commercial reference". They generated dataset with information on the weather, schedules, building (including different temperatures and light levels, power consumption). The results demonstrated that when using DPCRT, the solution enables 90% accuracy on learning predictive models, and for 48.6% reduction in costs.

[Chandan et al. 2015] explored the use of a pre-cooling method that makes use of data to reduce the operation energy cost and proposed a model for building thermal dynamics. The model targets minimum data requirements from a BMS. In particular, they proposed what is called the "gray box" approach that models thermal dynamics of the building. They illustrated the use of the model for the evaluation of pre-cooling policies in a commercial building. Standard linear regression has been used to solve the regression problem, and sensorial data from the existing BMS (without additional deployment). Results shows that a 30 minutes of pre-cooling from the "default start time" with a setting of the temperature to $26°C$ enables to reduce peak demand by 9.5%.

3.1.3 Discussion. Solutions presented in this section dealt with energy profiling either for (1) predicting load per appliance, or (2) predicting load per function such as heating, cooling, and lighting, or (3) predicting the global energy consumption at scales ranging from a single room to a set of buildings in a city. Regression is the ML problem considered in most solutions, where different tools have been used including ANN, LR, DL combined with time series, SVR, RT, etc. RMSE, correlation coefficient, p-value, and F-measure are metrics used for performance evaluation in most works, while some works measured energy cost and the reduced energy. A couple of solutions apply to any isolated building, while other solutions are tailored to buildings enabled with HVAC and/or connected to smart grids and rely on advanced options to acquire information (e.g., available information from the grid, available sensors in the HVAC). Most of the works focused on predicting peak hours, which has a high potential of applications in demand response systems, notably for heating and cooling. Prediction granularity in the presented works varied from minutes to days and weeks, while $24h$ ahead prediction is the most used. Online learning has been used to enable adaptability to long term changes. Besides regression, clustering has been used with analyzing tools to describe large city scale data and answer questions such as, the effect of weather and time on the energy consumption, the correlation between building characteristics and its energy use, the impact of sun penetration, etc. Solutions presented in this section are summarized in table 7 (Appendix).

3.2 Appliance Profiling and Fault Detection

3.2.1 Overview and Problem Statement. Solutions that use ML tools to identify/track appliances, detect anomalies/failures in the different components of the energy management system (airflow, air handling unit, etc.) are presented in this section. Electronic appliances represent a major source for hazardous situations and energy waste. For instance, a study in [Wang et al. 2014] show that 70% of computers and related equipments are left on all the time. Therefore, the automatic identification of appliances, their states, and/or anomalies has several purposes in smart buildings such as better understanding of the energy consumption, appliance maintenance, and indirect observation of human activities.

3.2.2 State-of-the-art. [Katarina et al. 2017] dealt with identifying abnormal consumption behavior in buildings and proposed an anomaly classifier. The latter is based on patterns and uses *"collective contextual anomaly detection sliding window (CCAD-SW)"*, and the *"ensemble anomaly detection (EAD)* framework. CCAD [Capretz and Bitsuamlak 2016] is able to detect anomalies but only with a latency that is proportional to the sliding window, which is unsuitable for services that are delay intolerant such gas-leak detection. CCA-SW addresses this shortcoming by using adaptive overlapping sliding windows to accommodate, (1) fast identification of urgent anomalies, (2) analysis/profiling of building energy consumption in long terms. Some contextual features have been added to the training set, e.g., day, month, season, etc, while ignoring other contextual information such as occupancy and weather. To enrich the training set, statistical values have been derived and used in the learning process such as the *"inter-quartile range value"*, the mean and standard deviation of the data (in every window), etc. A total of 25 heterogeneous features have been selected and used in the overall learning process. These features have different scales, and the large values are more influencing. The training set has been normalized by re-scaling the features to fit in the interval $[0 \cdots 1]$, which equally balances the weights of the features. Different anomaly detection classifiers have been combined (using majority-voting) in a generic framework (EAD), in which k heterogeneous kernels that ensure diversification in the output are launched to learn anomalies from the training data. SVR and random forest classifiers were used for every kernel. The majority voting process has been used from the k kernels to generate the global classifier. The EAD framework has been evaluated using real-world datasets from a company. Results showed that EAD enables 3.6% improvement in the CCAD-SW sensitivity, and a reduction of 2.7% in the false-alarms rate.

Automatic identification of appliances through the analysis of their electricity consumption (signatures) has been considered in [Ridi et al. 2015]. The authors developed a collection of signatures database (ACS-F database) that they made available for the scientific community. ACS-F contains different brands and/or models of appliances with 450 signatures. This database is suitable when using highly-intrusive power load monitoring where smart meters are dedicated to individual appliances with no signal aggregation. This requires high number of smart meters, but the authors justify this choice by the advances in IoT that is providing low cost smart plug meters of improved precision. Despite the constant reduction of smart meters' cost through IoT technologies, the cost of this approach remains an issue for large scale deployments. The authors proposed two protocols for the appliance identification task, where data collection cycles are divided into two sessions. In the first protocol (called intersection protocol), all the signatures contained in the first session are used for training, while the rest of signatures compose the test set. In the second one (called unseen appliance protocol), all instances of both sessions are taken to perform a k-fold cross-validation. The whole duration of the signals (1 hour) is used in both protocols, which is too long and impractical for building applications and services requiring timely actuation. The authors proposed an approach to adjust dynamically the window length and thus *dynamic* versions of both protocols. For the appliance identification, three ML algorithms have been applied on ACS-F: KNN, GMM, HMM. The results showed that most performance accuracy can be gained using up to 30 minutes of analysis window, that the intersection protocol allows to reach better accuracy, and that HMM and GMM provides better performance than KNN (they allow to reach more than 90% for the intersection protocol).

[Wang et al. 2014] considered tracking the states of electrical appliances (ON/OFF) with a minimum number of smart meters, i.e., using less meters than appliances. This is by considering the time correlation of the activation of appliances, where the switching events are sparse in short periods of observation. An entropy-based approach has been used to study the required number of smart meters and derive a lower-bound, which has been used to develop *"a meter deployment optimization algorithm (MDOP)"*. The authors also proposed what they called the FSD (*"Fast Sequence Decoding"*) algorithm to track every appliance and the sequence of states it follows. FSD is based on HMM and uses a *"monometer tree forest"* to model independent meters, i.e., a tree forest where the nodes in every tree are monometers. States decoding in monometer trees are performed in parallel. A monometer tree with N appliances has 2^N possible states, and thus requires $O(2^N)$ comparisons to find the most likely feasible state. The authors proposed "*offline state sorting*" to speed up the online searching, where they sort (offline) the 2^N states according to their energy values and prepare an ordered vector representing the states for online binary search. An HMM-based online state decoding algorithm was then proposed to run in every monometer tree for the search of the best reward path connecting the states. The model needs only limited offline knowledge to initialize the states of HMMs for training. The authors gave an example for initiating the states at midnight as it is known a priori (offline) that all appliances are in an off state. The authors showed that contrary to existing HMM approaches that have a time-complexity of $O(t2^{2N})$ to decode online sequences (where N is the number of appliances), FSD reduces this complexity to a polynomial order, i.e., $O(n^{U_t+1})$, where U_t represents an upper-bound (in a single sampling-slot) on the number of switching events that occurs simultaneously, and $n < N$. MDOP and FSD have been evaluated extensively using both artificial data and a real dataset (PowerNet) [Kazandjieva et al. 2009a]. The results showed more than 80% reduction in the cost of deployment with more than 90% accuracy in state tracking.

[Ferdoash et al. 2015] considered the use of large scale BMS to identify excessive airflow in the HVACs, as well as the calculation of the optimal pre-cooling start-time to reach the desired temperature. The authors deployed temperature sensors to collect data from two buildings. They combined the collected data with weather data from a weather station and used linear regression and SVM to derive simple models. They demonstrated the effectiveness of such models to identify potentials on energy saving for HVAC. The authors evaluated how the models enable efficient dynamic selection of energy policy for HVAC. Additional flow beyond the minimum flow setting at each *"Variable Air Volume (VAV)"* has been analyzed, starting from the time when set temperature is reached. The flows in a zone from all the VAV systems have been combined to calculate the extra flow at the *"Air Handling Units (AHU)"* level. Energy saving possibilities at the AHU level has been calculated by using the obtained *"power-flow relationship"*. The results showed that aggregate savings through appropriate operation of the VAVs enable an elimination of about 5% of the extra flow per month at the facility scale.

[Li et al. 2017] presented a feature selection method (IGFF) for building fault detection and diagnosis (FDD). IGFF selects the subset of features (sensor variables) that maximizes *"mutual information between candidate variables and the fault labels"*. That is, the subset of features of maximum dependence with the fault labels (the targeted random variable). The selected features may help improving the FDD accuracy and guide the operators in the deployment of sensors. They may also (especially in case of limited resources of measurement) serve as reference for sensor configuration. The authors argued that different types of building working conditions can be accommodated by IGFF, independently from the FDD algorithms. They proofed that IGFF guarantees a near-optimal solution in maximizing mutual information and allows to derive upper-bounds. Based on realistic dataset, a case study on the AHU has been performed. Multi-classification techniques have been considered, notably QDA, LR, ANNs, and multiple SVM. The chosen features have been fused together and plugged into these classification techniques. The numerical results showed that the IGFF improves (in comparison to some baselines and state-of-the-art solutions) conventional classification methods in terms of FDD performances. Solutions cited in this section are summarized in table 8 (Appendix).

3.2.3 Discussion. Different solutions for appliance tracking/profiling and anomaly/fault detection have been presented in this section. Classification is the ML problem considered in most of the solutions. Customized classifiers have been applied to electrical signatures that are obtained from continuous power load monitoring either with (1) power sensors and IoT smart plugs integrated to wall sockets, or (2) through global smart meters. The former approach is highly intrusive and costlier than the latter, but more precise in terms of signature analysis. Advances in IoT technologies might pave the way for large deployment of low-cost power control plugs in the future. The second approach has the advantage of easy installation, low cost and privacy protection. A practical solution is to use smart plugs in a limited locations along with disaggregation algorithms on the obtained signals to deduce individual appliance consumption/state. However, this yields a trade-off in sensor deployment between reducing the cost vs. accurate tracking. Signal analysis through data disaggregation is more difficult when multiple equipments are used simultaneously. Information on single sources of consumption can be partially retrieved through the application of disaggregation algorithms (similarly to the solution presented in [Lange and Bergés 2016]), but the performance depends upon the type and the number of appliances that are used simultaneously. Anomalies might be detected by analyzing the power consumption of appliances and/or behavior of some components of the BMS (e.g., temperature variation of airflow). Sensitivity to the detection delay varies from an application to another and might be critical for applications such as gas leak detection. Fast detection might be achieved with some adaptive techniques on the classifiers but with the cost of an increased false positives, which yields a trade-off. To evaluate their solutions, some works used metrics related to classification, e.g., accuracy rate, precision and recall, while others used some specified metrics related to outlier detection such as metering noise, AUC (Area Under the Curve) and cost saving ratio. The baseline methods used in this category include statistical-based and neighborhood-based outlier detection approaches. While these approaches have been used as baseline, advanced outlier detection algorithms such as such LOF (Local Outlier Factor) [Breunig et al. 2000] have not been considered. It is interesting to explore such algorithms for fault detection. Solutions cited in this section are summarized in table 8 (Appendix).

3.3 Inference on Sensors

3.3.1 Overview and Problem Statement. This section presents ML approaches that have been used to infer information on sensors, such as their types, their positions and orientations. Meta-data related to sensors are trained to learn models for sensor's placement and orientation, sensor's type identification. The difficulty of such task lies on to the effective use of the meta-data coming from several sensors in the learning process, i.e., how to deal with data heterogeneity problem.

[Gonzalez et al. 2016] dealt with localization of building-installed spatial sensors (inference of the relative positions and orientations) that are used for object tracking. The proposed approach was tuned to sensors that detect objects without identification. The positions of sensor nodes have been mined from their data using association rules mining (ARM) independently from the underlying

technologies and infrastructure. Walking trajectories within building spaces have been explored by objects' tracking and the creation of link rules between sensors, which enabled to infer relative sensor arrangement. The approach consists in three main steps:
(1) Transition event extraction: This step aims to built the set of transactions from the sensors' data. Each track, *tr*, in a sensor node, *s*, is defined by the set, $\{(r_1, p_1, t_1), (r_2, p_2, t_2), \Delta t\}$, where r_1 (resp. r_2) represents the regions of an object that is detected when it enters (resp. exits) the sensor's coverage area, p_1 (resp. p_2) represents the positions, t_1 and t_2 their respective times. The lifetime of the track, *tr*, of the sensor, *s*, is defined as the difference between these times, i.e., $\Delta t = t_2 - t_1$. The transition event is defined by the tracks of the sensor *s*, and *t*, where Δt does not exceed a given threshold, OW, called observation window. The transaction is defined by the set of transition events that are related to the same objects.
(2) Link rule estimation: This step aims to apply ARM from the set of transactions created in the previous step to define relationships between tracks of sensors and determine the most relevant sensors. At the end of this step, a set of link rules are extracted. A link rule is a rule that links two tracks of two different sensors. For example, the link rule $(s_1, r_1 \Rightarrow s_2, r_2)$ with the confidence of μ means that if the given object is captured by the region, r_1, and the sensor, s_1, then there is μ of chance that entries to the region, r_2, are captured by the sensor, s_2.
(3) Sensor matrix arrangement: The placement of the sensors are determined in this step based on the set of link rules extracted in the previous step. The rules are first sorted in descending order on the confidence values. The link rules are then selected, and the sensors are placed according to the tracks of the selected rules. This process is repeated until all the sensors are deployed.

The proposed solution is general and applies to any type of spatial sensors. For the evaluation, the authors used "thermopile array sensors". Four building-scenarios have been considered with different arrangements in sensor deployment (in position and numbers): (1) corridor, (2) T-crossing, (3) meeting room, (4) foyer. Real dataset has been collected over several weeks in each scenario. The results showed high accuracy in inferring positions for the T-crossing scenario (approaching 100%), which is the most dynamic scenario in terms of occupants' movements. The average accuracy for the other scenario ranged from 50% to 70%. This fluctuation in accuracy represents the major drawback, which makes the solution strongly dependent upon the activity of the monitored area. However, the solution is useful as a tool for the maintenance of *"heterogeneous spatial sensor networks"*, which is assured independently from the communication protocol.

[Hong et al. 2015] considered automatic inference of the type of sensors in a building (without manual labeling). Techniques from transfer learning have been used to learn, from meta data of a labeled building, statistic classifiers. The latter have been adapted for use in another unlabeled building. The proposed techniques allow for mapping independantly from the structure of the meta data. Starting from a building where all the dataset is labeled, the solutions uses time-series to train "multiple statistical classifiers", which predict the types of sensors through the raw data and the patterns of the readings, e.g., "*readings from air temperature sensors are different and change more slowly than those generated by light or CO_2 sensors*". ML methods including RF, LR, and SVM with RBF kernels have been used to design the classifiers, which have been derived from the labeled building to label the points in the target building. Weights of the classifiers are determined according to the consistency of the prediction of the instances in the target building. For the evaluation, the authors used a dataset collected during seven days from 2500 sensors in three commercial buildings. Several types of sensors have been used including temperature (of different types, e.g., ambient temperature, water temperature, etc.), CO_2, and humidity. In the three buildings, true sensor types have been created manually. The proposed techniques have then been applied on every building to "automatically infer" the sensor types for the other remaining buildings. Experimental results showed more than 85% accuracy in labeling about 36% of the points, and up to 96% accuracy for labeling 81% of the points in some cases.

[Gao et al. 2015] considered inferring the sensor type and presented a solution to automat the association between sensor measurements and descriptive tags (from a "standard set"). This method has been used in a semi-automatic meta-data inference framework that enables to learn from measurement data of a building automation system. The labeling problem has been casted as a supervised-learning problem, and the authors trained some classification algorithms for the matching of the measurements' features with a "standard form of tagging" (Haystack tags). The classification algorithms used are, RF, K-NN, DT, GNB , SVM with RBF kernels, LR, AdaBoost and LDA. The framework has been evaluated on two buildings with sensors of several types including temperature, PIR, CO_2, power. For each classifier, 20% the dataset has been used for training and the remaining 80% for tests. The results showed that the approaches providing the best performance are RF, k-NN, DT. The results also indicated an average accuracy of 95% when performing the training and the test in the same facility.

3.3.2 Discussion. Different solutions for inference on sensors have been presented in this section. Association rule mining (ARM) and classification are the ML problems considered in most of the solutions. ARM aims to derive different dependencies between meta-data sensors. These dependencies are used to determine sensor placement. Classification aims to learn from meta-data to predict the type of the sensor's measurement. To evaluate their solutions, some works used metrics related to measuring efficiency of the classification, e.g., accuracy rate, precision and recall, while other works used some ARM related metrics such as support, confidence, and the number of the discovered frequent patterns. The baseline methods used in this category are approaches related to classification such as KNN, SVM , DT., as well as other approaches related to ARM such Apriori[Agrawal et al. 1993]. Table9 (Appendix) compares the solutions presented in this subsection, and Fig. 4 is at taxonomy of all the energy/device centric solutions.

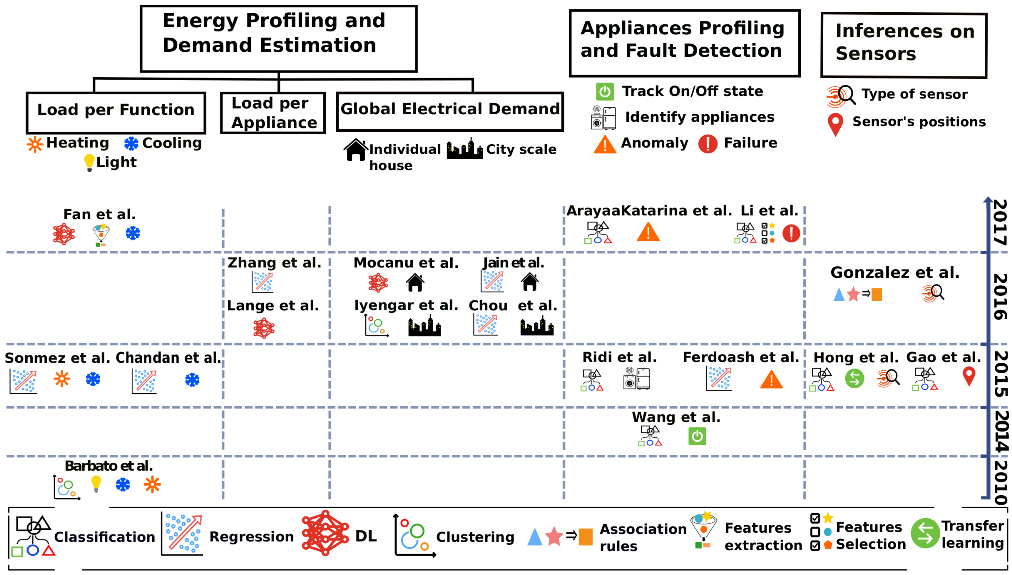

Fig. 4. Energy/device centric ML solutions

4 GENERAL REMARKS AND FUTURE DIRECTIONS

Earlier works in the literature on smart buildings focused on energy performance and proposed solutions based on realtime monitoring and actuation. While being vital, energy represents only one facet of the overall building performance [Molnar et al. 2015]. Further, recent studies indicate that the use of simple energy monitors and automation systems does not spur energy saving actions by the consumers who quickly adopt the reported consumption as normal and often see no reason to reduce it further. This makes most of the current building automation systems either rejected by the users or used in non-optimal ways [Hargreavesn et al. 2013]. Therefore, even from the economic and energy saving perspectives, users' preferences/comfort have to be considered in next generation's buildings. Energy saving solutions should be developed jointly with the stimulation of the occupants involvement. They should improve the comfort of the occupants or at least guarantee there will have no negative impacts on their preferences. Prediction models are needed to accurately estimate occupants' presence, their preferences and behavior, energy consumption and profiles, etc. ML has largely been used for this purpose in the last few years and many solutions have been proposed as presented in this paper. In addition to energy management systems, these solutions are also significantly influencing other sectors such as retailing (smart shopping centres), health-care (smart hospitals and homes), security and safety (intrusion/anomaly detection systems), etc. This requires the development of new metrics in the future that thoroughly combine energy performance and business/comfort performance measures. Separating user comfort and energy saving may lead to wrong decisions. The impact of such wrong decisions in traditional BMS has been investigated in [Lazarova-Molnar et al. 2015; Seppanen et al. 2004; Wyon 2004]. These studies show that an uncomfortable environment may significantly reduce employees' productivity and may engender losses for a business that are higher than the gain in energy saving.

One of the key scientific challenges that should be dealt with to develop user-centric, personalized services and applications is to exploit the heterogeneous nature of the data stemming from different sources such as sensors, smart phones, existing databases, etc., and through different data transmission schemes such as Internet, cellular networks, wireless personal area networks. This heterogeneity results in a variety level of trustworthiness, frequency, resolution, reliability, and it imposes various types of uncertainties for ML tools such as biased readings, failed sensors, etc. Many ML approaches in different application categories have been analyzed in this paper. For occupancy, most solutions of the literature range from binary classification to estimating the exact number of occupants. Few works have been devoted to more advanced occupancy monitoring services such as gender classification and person identification, which have many applications, e.g customized advertisement in malls and shopping centers, security, safety, advanced health care systems. Some preliminary solutions have been proposed as presented in this paper, but generalizing these solutions and achieving high accuracy with reasonable cost remains challenging. For activity recognition, one of the challenges is the diversity (from an individual to another) of the same activity. For example, *"the differences in speed of walking, gestures, sleep habits, etc., are ambiguous to a general passive learning model"* [Hossain et al. 2017]. It is then important to build adaptive models that can be personalized for individual users. Semi-supervised approaches such as active learning might be useful to find out the data points that are most informative and actively request labels on demand. This will enable the detection of unseen activities, contrary to passive and supervised learning. Ground truth collection is also a difficult task, given the variety and dynamic of human activities [Hossain et al. 2017]. Further, activity monitoring in multi-occupied spaces is challenging, where it is difficult to distinguish individuals activity with the overlapping readings from different sensors. Nevertheless, multi-occupant settings may represent potentials for active learning where the occupants might be asked to label data from one another [Hossain et al. 2017].

Exploring this to overcome lack of data labels and to infer individual activities in such a setting is an open research trend. Transfer learning has been used for activity recognition and to infer information on sensors. Similar approaches can be applied for energy profiling, which will permit to take advantage from existing datasets of similar buildings. Tracking ON/OFF states of appliances is a challenging problem. Current solutions are facing a difficult choice between reducing the meter deployment cost and targeting accurate tracking. Dense smart meters deployment provides high accuracy but has high costs (of deployed meters, maintenance, etc.) while deploying small numbers of meters generally suffers from low accuracy.

The hyper-parameters used in the solutions presented in this paper may be categorized according to two key aspects, (1) ML tool-related hyper-parameters and (2) problem-related hyper-parameters. The first category groups the hyper-parameters defined by the ML model such as the learning rate in LR, the number of hidden units in ANN, etc. The problem-related hyper-parameters are those proposed while solving the appropriate problem, e.g., the threshold used by [Ghahramani et al. 2015] to separate comfort and discomfort votes, the weights of false positive rates and false negative rates used by [Zhou et al. 2015] to penalize false positive classifications. From tuning perspective, hyper-parameters may also be grouped into two categories. (1) empirically tuned hyper-parameters (2) theoretically crafted hyper-parameters. In the first category, the authors use a trial and error process to define the optimal values for hyper-parameters after multiple tests. This category includes also works using cross validation for tuning e.g. [Lange and Bergés 2016]. Examples of the second category include [Sonmez et al. 2015], which used GA and ABC heuristics to define the optimal activation function for each node of an ANN, and [Sarkar et al. 2016], which applied curve fitting to estimate the parameters of a Gaussian and Beta functions.

To evaluate the performance of their solutions, the authors relied either on, (1) their own collected datasets, or (2) on some publicly available datasets, or (3) on generating artificial data by simulation. Table 3 provides details about publicly available datasets used by the works presented in this paper, where each dataset is linked to its citing paper. Besides these datasets, other building datasets can be found in CASAS[6] and UC Irvine Machine Learning [Dheeru and Karra Taniskidou 2017] repositories.

Two types of approaches have been used in works that relied on data collection to construct power consumption dataset: (1) the use of sensors or individual-meters at every part of the electrical network, vs. (2) the use of global smart meters with power desegregation techniques. The first is intrusive and costly, but it has the advantage of providing high granularity in monitoring and more precision in terms of signature analysis. The use of global meters with desegregation techniques has the advantage of easy installation, low cost and preserving privacy. However, signal analysis is more difficult when multiple equipments are used simultaneously. Advances in IoT technologies might pave the way for large deployment of low-cost power control plugs in the future. Another option that has recently been explored is the use of non-intrusive techniques such as the existing sensors (of HVACs), which is useful to eliminate or reduce the need of deploying dedicated sensors.

Privacy is one of the key issues that refrain the implication of potential participants in crowd-sourcing for data gathering in smart building applications (through smart phone applications, online reports, etc.). Few works have been conducted for privacy guarantee in this context [Hamm et al. 2015]. Privacy guarantee when sensing is another critical issue. One of the elementary measures is the use of none-invasive sensing technologies to collect data (e.g., PIR, ultrasonic sensors, etc.) instead of cameras and microphones (except in public areas where it is permitted to install cameras such as corridors, waiting rooms, halls, etc.). This reduces the applicability of existing advanced image processing technologies. When using cameras in public areas, the exploration of video

[6] http://casas.wsu.edu/datasets/

Table 3. Publicly available datasets.

Section	Solution	Dataset	Features	Period	# participants	# Buildings
2.1 / 2.2	[Carolis et al. 2015; Soltanaghaei and Whitehouse 2016]	Aruba [Cook 2012]	motion, temp, and door closure sensors, sensors layout in the home. annotated activities	26 weeks	1 + 2 visiting	1 (8rooms)
2.2	[Chiang et al. 2017]	MIT MAS622J [Tapia et al. 2004]	Binary sensors (PIR, switch) installed in everyday objects, activity labels, time	2 weeks	2	2
	[Carolis et al. 2015; Nait Aicha et al. 2017]	[van Kasteren et al. 2008]	binary sensors	28 days	2	2
2.3	[Zhou et al. 2015]	RP-884 [De Dear and Brager 1998]	temp, humidity, air velocity, clothing, metabolic rate. PMV	NA	21000	160 worldwide
3.1	[Zhang et al. 2016]	DOE-2 based software weather data [Wea 2018]	weather data	NA	/	wide area
	[Mocanu et al. 2016]	Individual household electric power consumption [Dheeru and Karra Taniskidou 2017]	9 power consumption attributes	47 months	NA	1
	[Lange and Bergés 2016]	Blued [Anderson et al. 2012]	voltage, current measurements for 42 appliances	1 week	1 family	1
3.2	[Katarina et al. 2017]	Powersmiths [Pow 2018]	HVAC power	2 years	NA	school
	[Wang et al. 2014]	PowerNet [Kazandjieva et al. 2009b]	power of 134 appliances	+700 days	NA	large office-building

streams to infer comfort information (e.g., thermal) is one of the current trends [Jazizadeh and Pradeep 2016]. We showed that ML are also used for failure detection in buildings. In this context, fast detection is required in some applications (e.g., gas leakage), which might be achieved with some adaptive techniques on the classifiers but with the cost of increased false positives. Since such applications need real time detection with high accuracy, investigating the impact of a wrong decision by the ML model is worth studying. The need for fast detection inevitably raises a trade-off that has not been addressed. Combining advanced ML (such as hierarchal methods and DL) with game-theory approaches might be explored for this purpose. DL has already been explored in smart buildings to analyze heterogeneous data at a larger scale and generate general models. However, DL models might be fooled by misleading examples in the training process, which might have dramatic impacts on the performance [Ota et al. 2017]. Feature selection and data filtering are important steps to reduce such impacts.

The use of ML in commercialized smart building solutions starts to attract investors' attention. The most common way of deploying such ML based applications is by hosting services in the cloud and enabling home smart devices to connect to it via WiFi or Ethernet. These devices aim at detecting and learning usage patterns to anticipate house control based on historical manipulations from users. Viaroom Home[7] is an example of these applications. Their algorithm analyzes 48h habits to control lighting, heating and appliances. Viaroom is compatible with state-of-the-art home automation devices such as Philips Hue[8] and Fibraro lighting[9], Qubino shutters[10], Aeotec

[7] http://viaroom.com/
[8] https://www2.meethue.com/en-us
[9] https://www.fibaro.com/en/
[10] http://qubino.com/products/flush-shutter-dc/

smart plugs[11], Yale door locks[12] and Foscam security cameras[13]. Nest Labs[14] (Google's subsidiary) also uses ML in their products. Their learning thermostat monitors the users thermal settings to preserve energy and provide comfort, while their security cameras use face recognition to detect intrusions. Wireless speakers, such as Amazon Echo and Google home assistant, use speech recognition to analyze different requirements from users. These speakers also combine many smart devices products to provide centralized control. Electronics companies such as LG and Samsung also joined the race to endow smartness in home appliances. LG DeepThinQ is an AI platform that add intelligence to the object depending on its task, e.g. a vacuum robot is able to detect objects while a fridge recognizes missing items. Samsung's Bixby allows voice control of this brand's appliances. The common point between these products is hosting services in the cloud. This raises two problems that are slowing down the large adoption of smart home devices: the privacy of sensitive information communicated to third parties, and the need of efficient communication protocols. The efficiency in communication comes with the cost in high energy usage. This explains why most of available products needs constant source of energy. Researchers and developers are optimistic about the support of Tenserflow by Raspberry pi platform[15] as it opens the opportunity for embedded systems to run DL architectures. Such possibility may allow to avoid (or reduce) the dependency upon cloud support.

We also point out that in real applications, ML services, requirements and challenges may also depend upon the type of buildings. For example, gender classification might be useful in commercial buildings. Offline training is more appropriate for residential buildings, while online training is more appropriate for non-residential buildings. Commercial buildings are potentially sources of collecting important, large amount of data, but some incentives should be provided to attract participation of occupants. This attractiveness is not an issue in residential buildings, but in this category of buildings, the collected data might be limited in time. In addition to the type of buildings, applicability of the solutions presented in this paper also depends of the size of buildings. Some solutions apply to any isolated building, while other solutions have been designed to those enabled with HVAC and/or connected to smart grids.

One of the hot research topic is what is know as "software defined buildings" [Dawson-Haggerty et al. 2012] that inspires from the domain of software-defined networks for application in buildings. This is promising to enable decoupling between applications and the hosting physical building, which will considerably reduce efforts needed to add new functions, applications, and services. It will also extend communication capabilities between buildings and with third parties (e.g., the electrical grid) and will pave the way for the design of autonomous solutions and customized services at a high level of abstraction. Finally, note that that the literature related to the use of ML in smart building applications involves different communities and different disciplines. This implies the use of different terminology, different ways of measuring performance, different angles of looking at the problems, etc. [Millera et al. 2017] provided a good analysis on these issues. The good point here is the multi-disciplinarity of this domain. Several cross-disciplinary venues are already established (e.g., ACM BuildSys conference, ACM E-energy conference, related tracks in many other ACM and IEEE conferences, etc.). These networking opportunities will certainly promote synergy between communities to work on multi-disciplinary projects and achieve goals at large scales.

[11] https://aeotec.com/z-wave-plug-in-switch
[12] https://www.yalelock.com/en/yale/com/
[13] https://www.foscam.com
[14] https://nest.com/
[15] https://www.tensorflow.org/install/install_raspbian

5 CONCLUSION

Model-based solutions with deterministic control algorithms have largely been used in building automation systems for a long time. They have been sophisticated with the introduction of wireless sensing/communication capabilities for realtime optimized actuation. However, such solutions reached their limits due to the increased complexity of buildings and users' demands, and they failed to motivate consumers to spur energy saving actions or even to appropriately use/configure the control system. Learning is the most appropriate alternative in this case, where the optimal policies are not a priori known but can only be developed using data or experience. The current trend in smart building applications is thus to explore approaches derived from machine learning (ML) for inferring knowledge about the occupants, devices and energy profiles. This enables to smoothly take control actions and to provide advanced services that will not only promote users' comfort but attract them to join the loop for the energy saving policy. We have provided throughout this paper a comprehensive survey and taxonomy of ML solutions in smart building applications. The solutions presented have been split into two main classes. The first groups occupant-centric solutions where the ML approaches have been used to deal with features related to occupants. This has been further divided into three sub-categories, (1) occupancy estimation and identification, including different levels of occupancy estimation, from binary information (occupied vs. vacant space) to assessing the exact number of occupants and identification of the gender of occupants or the persons, (2) activity recognition of the occupants, (3) estimating their preferences and behavior. The second main category includes energy/device centric solutions, where ML is used to estimate aspects related either to energy or devices (appliances and sensors). They have been divided into the three sub-categories (1) energy profiling and demand estimation, (2) appliances profiling and fault detection, 3) inference on sensors. Different solutions have been presented in every category, compared, and discussed from the ML perspective, as well as the technical and application aspects of their implementations and/or experimentations. The paper ends in Sec. 4 with a summery of the most relevant lessons we concluded from this survey and the future directions of research trends in this arena. While application of ML has gained high maturity in its traditional domains such as image/speech processing, human languages, spam filtering, etc., the use of ML in smart buildings is in its embryonic age. We have seen only the tip of the iceberg, while much exploration is needed and deep progress is required in all directions to reach mature solutions for large scale deployments.

ACKNOWLEDGMENTS

This work is supported, in part by the Algerian Ministry of Higher Education through the DGRSDT, and by the Melody Project through the Research Council of Norway under a mobility grant no. 225885.

REFERENCES

2018. Powersmiths: Power for the Future. https://ww2.powersmiths.com/index.php?q=content/powesmiths/about-us. (2018). Accessed: 7/12/2018.

2018. Weather Data & Weather Data Processing Utility Programs. http://doe2.com/index_wth.html. (2018). Accessed: 7/12/2018.

Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. 1993. Mining association rules between sets of items in large databases. In *Acm sigmod record*, Vol. 22. ACM, 207–216.

Alaa Alhamoud, Pei Xu, Frank Englert, Philipp Scholl, The An Binh Nguyen, Doreen Böhnstedt, and Ralf Steinmetz. 2015. Evaluation of user feedback in smart home for situational context identification. In *2015 International Conference on Pervasive Computing and Communication Workshops, PerCom Workshops*. IEEE, St. Louis, MO, USA, 20–25.

Ethem Alpaydin. 2014. *Introduction to Machine Learning*. The MIT Press.

K Anderson, A Ocneanu, Diego Benitez, D Carlson, A Rowe, and M Berges. 2012. BLUED: A fully labeled public dataset for event-based non-intrusive load monitoring research. (01 2012), 1–5.

M. Antunes, D.G. Gomes, and R. Aguiar. 2013. Towards behaviour inference in smart environments. In *The Conf. on Future Internet Communications - CFIC*. IEEE, Coimbra, Portugal, 1–8. https://doi.org/10.1109/CFIC.2013.6566324

Omid Ardakanian, Arka Bhattacharya, and David Culler. 2016. Non-Intrusive Techniques for Establishing Occupancy Related Energy Savings in Commercial Buildings. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys '16)*. ACM, NY, USA, 21–30.

Miloud Bagaa, Ali Chelli, Djamel Djenouri, Tarik Taleb, Ilangko Balasingham, and Kimmo Kansanen. 2017. Optimal Placement of Relay Nodes Over Limited Positions in Wireless Sensor Networks. *IEEE Trans. Wireless Communications* 16, 4 (2017), 2205–2219.

Dustin Bales, Pablo A. Tarazaga, Mary Kasarda, Dhruv Batra, A. G. Woolard, Jeffrey D. Poston, and V. V. N. S. Malladi. 2016. Gender Classification of Walkers via Underfloor Accelerometer Measurements. *IEEE Internet of Things Journal* 3, 6 (2016), 1259–1266.

Wolfgang Banzhaf, Frank D. Francone, Robert E. Keller, and Peter Nordin. 1998. *Genetic Programming: An Introduction: on the Automatic Evolution of Computer Programs and Its Applications.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Antimo Barbato, L. Borsani, and Antonio Capone. 2010. A Wireless Sensor Network Based System for Reducing Home Energy Consumption. In *Proceedings of the Seventh Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON, June*. IEEE, Boston, MA, USA, 1–3.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *ACM sigmod record*, Vol. 29. ACM, 93–104.

B.Yildiz., J.I. Bilbao, and A.B. Sproul. 2017. A review and analysis of regression and machine learning models on commercial building electricity load forecasting. *Renewable and Sustainable Energy Reviews* 73 (June 2017), 1104–1122.

David Caicedo and Ashish Pandharipande. 2015. Sensor-Driven Lighting Control With Illumination and Dimming Constraints. *IEEE Sensors Journals* 15, 9 (2015), 5169–5176.

J Canny. 1986. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8, 6 (1986), 679–698.

Miriam A. M. Capretz and Girma T. Bitsuamlak. 2016. Collective contextual anomaly detection framework for smart buildings. In *International Joint Conference on Neural Networks, IJCNN, July 24-29*. IEEE, Vancouver, BC, Canada, 511–518.

Berardina De Carolis, Stefano Ferilli, and Domenico Redavid. 2015. Incremental Learning of Daily Routines As Workflows in a Smart Home Environment. *ACM Trans. Interact. Intell. Syst.* 4, 4 (Jan 2015), 20:1–20:23.

Vikas Chandan, Arun Vishwanath, Min Zhang, and Shivkumar Kalyanaraman. 2015. Short Paper: Data Driven Pre-cooling for Peak Demand Reduction in Commercial Buildings. In *Proceedings of the 2Nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments (BuildSys '15)*. ACM, NY, USA, 187–190.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.* 2, 3 (May 2011), 27.

Yi-Ting Chiang, Ching-Hu Lu, and Jane Yung-jen Hsu. 2017. A Feature-Based Knowledge Transfer Framework for Cross-Environment Activity Recognition Toward Smart Home Applications. *IEEE Trans. Human-Machine Systems* 47, 3 (2017), 310–322.

Jui-Sheng Chou and Ngoc-Tri Ngo. 2016. Time series analytics using sliding window metaheuristic optimization-based machine learning system for. *Applied Energy* 177 (Sep 2016), 751–770.

Diane J Cook. 2012. Learning setting-generalized activity models for smart spaces. *IEEE intelligent systems* 27, 1 (2012), 32–38.

Stephen Dawson-Haggerty, Jorge Ortiz, Jason Trager, David E. Culler, and Randy H. Katz. 2012. Energy Savings and the "Software-Defined" Building. *IEEE Design & Test of Computers* 29, 4 (2012), 56–57.

Richard De Dear and Gail Schiller Brager. 1998. Developing an adaptive model of thermal comfort and preference. (1998).

Alessandra De Paola, Marco Ortolani, Giuseppe Lo Re, Giuseppe Anastasi, and Sajal K. Das. 2014. Intelligent Management Systems for Energy Efficiency in Buildings: A Survey. *Comput. Surveys* 47, 1 (Jun 2014), 13:1–13:38.

Dua Dheeru and Efi Karra Taniskidou. 2017. UCI Machine Learning Repository. (2017). http://archive.ics.uci.edu/ml

Djamel Djenouri and Miloud Bagaa. 2016. Synchronization Protocols and Implementation Issues in Wireless Sensor Networks: A Review. *IEEE Systems Journal* 10, 2 (2016), 617–627.

J. Doak. 1992. *An Evaluation of Feature Selection Methods and Their Application to Computer Security.* University of California, Computer Science. https://books.google.dz/books?id=S_zhtgAACAAJ

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)*. AAAI Press, Portland, Oregon, 226–231.

Volodymyr Mnih et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015).

Cheng Fan, Fu Xiao, and Yang Zhaoc. 2017. A short-term building cooling load prediction method using deep learning algorithms. *Applied Energy* 195 (Jun 2017), 22–233.

Maria Pia Fanti, Gregory Faraut, Jean-Jacques Lesage, and Michele Roccotelli. 2018. An Integrated Framework for Binary Sensor Placement and Inhabitants Location Tracking. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48, 1 (2018), 154–160.

Afreen Ferdoash, Shubham Saini, Jitesh Khurana, and Amarjeet Singh. 2015. Poster Abstract: Analytics Driven Operational Efficiency in HVAC Systems. In *Proceedings of the 2Nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments (BuildSys '15).* ACM, NY, USA, 107–108.

Stefano Ferilli. 2014. WoMan: Logic-Based Workflow Learning and Management. *IEEE Trans. Systems, Man, and Cybernetics: Systems* 44, 6 (2014), 744–756.

D. Gale and L. S. Shapley. 1962. College admissions and the stability of marriage. *Amer. Math. Monthly* 69, 1 (1962), 9–15.

Jingkun Gao, Joern Ploennigs, and Mario Berges. 2015. A Data-driven Meta-data Inference Framework for Building Automation Systems. In *Proceedings of the 2Nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments (BuildSys'15).* ACM, Seoul, South Korea, 23–32.

Ali Ghahramani, Chao Tanga, and Burcin Becerik-Gerberb. 2015. An online learning approach for quantifying personalized thermal comfort via adaptive stochastic modeling. *Building and Environment* 92 (Oct 2015), 86–96.

Luis I. Lopera Gonzalez, Reimar Stier, and Oliver Amft. 2016. Data Mining-based Localisation of Spatial Low-resolution Sensors in Commercial Buildings. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys '16).* ACM, New York, NY, USA, 187–196. https://doi.org/10.1145/2993422.2993428

Jihun Hamm, Adam C. Champion, Guoxing Chen, Mikhail Belkin, and Dong Xuan. 2015. Crowd-ML: A Privacy-Preserving Learning Framework for a Crowd of Smart Devices. In *35th International Conference on Distributed Computing Systems, ICDCS, June 29 - July 2.* IEEE, Columbus, OH, USA, 11–20.

Tom Hargreavesn, Michael Nye, and Jacquelin Burgess. 2013. Keeping energy visible? Exploring how householders interact with feedbackfrom smart energy monitors in the longer term. *Energy Policy* 52 (Jan 2013), 126–134.

Olivier Hersent, David Boswarthick, and Omar Elloumi. 2012. *The Internet of Things: Key Applications and Protocols* (2nd ed.). Wiley Publishing, Hoboken, New Jersey.

Dezhi Hong, Hongning Wang, Jorge Ortiz, and Kamin Whitehouse. 2015. The Building Adapter: Towards Quickly Applying Building Analytics at Scale. In *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments, BuildSys 2015, Seoul, South Korea, November 4-5, 2015.* ACM, NY, USA, 123–132.

H. M. Sajjad Hossain, Md Abdullah Al Hafiz Khan, and Nirmalya Roy. 2017. Active learning enabled activity recognition. *Pervasive and Mobile Computing* 38 (2017), 312–330.

Aapo Hyvarinen. 1999. Survey on independent component analysis. *Neural computing surveys* 2, 4 (1999), 94–128.

Srinivasan Iyengar, Stephen Lee, David Irwin, and Prashant Shenoy. 2016. Analyzing Energy Usage on a City-scale Using Utility Smart Meters. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys '16).* ACM, NY, USA, 51–60.

Achin Jain, Rahul Mangharam, and Madhur Behl. 2016. Data Predictive Control for Peak Power Reduction. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys '16).* ACM, NY, USA, 109–118.

A. K. Jain, M. N. Murty, and P. J. Flynn. 1999. Data Clustering: A Review. *ACM Comput. Surv.* 31, 3 (Sep 1999), 264–323.

F. Jazizadeh, FM. Marin, and B. Becerik-Gerber. 2013. A thermal preference scale for personalized comfort profile identification via participatory sensing. *Building and Environment* 9 (Oct 2013), 68–140.

Farrokh Jazizadeh and S. Pradeep. 2016. Can computers visually quantify human thermal comfort?: Short Paper. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments, BuildSys@SenSys.* ACM, NY, USA, 95–98.

Simin Ahmadi Karvigha, Ali Ghahramania, Burcin Becerik Gerberb, and Lucio Soibelman. 2017. One size does not fit all: Understanding user preferences for building automation systems. *Energ and Buildings* 145, 15 (Jun 2017), 163–173.

Daniel B. Arayaa Katarina, GrolingeraHany F. ElYamanyac, Miriam A.M. Capretza, and Girma Bitsuamlak. 2017. An ensemble learning framework for anomaly detection in building energy consumption. *Energy and Buildings* 144 (June 2017), 191–206.

Maria Kazandjieva, Brandon Heller, Philip Levis, and Christos Kozyrakis. 2009a. Energy Dumpster Diving. In *Second Workshop on Power Aware Computing (HotPower).* 1–5.

Maria Kazandjieva, Brandon Heller, Philip Levis, and Christos Kozyrakis. 2009b. Energy Dumpster Diving. In *Second Workshop on Power Aware Computing (HotPower).*

Aqeel H. Kazmi, Michael J. O'grady, Declan T. Delaney, Antonio G. Ruzzelli, and Gregory M. P. O'hare. 2014. A Review of Wireless-Sensor-Network-Enabled Building Energy Management Systems. *ACM Trans.actions on Sensor Networks* 10, 4 (Jun 2014), 1–43.

Nacer Khalil, Driss Benhaddou, Omprakash Gnawali, and Jaspal Subhlok. 2016. Nonintrusive Occupant Identification by Sensing Body Shape and Movement. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys '16)*. ACM, NY, USA, 1–10.

Aftab Khan, James Nicholson, Sebastian Mellor, Daniel Jackson, Karim Ladha, Cassim Ladha, Jon Hand, Joseph Clarke, Patrick Olivier, and Thomas Plötz. 2014. Occupancy Monitoring Using Environmental and Context Sensors and a Hierarchical Analysis Framework. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings (BuildSys '14)*. ACM, NY, USA, 90–99.

Rida Khatoun and Sherali Zeadally. 2016. Smart cities: Concepts, Architectures, Research Opportunities. *Commun. ACM* 59, 8 (Aug 2016), 46–57.

Timilehin Labeodan, Wim Zeiler, Gert Boxem, and Yang Zhao. 2015. Occupancy measurement in commercial office buildings for demand-driven control applications A survey and detection system evaluation. *Energy and Buildings* 93 (April 2015), 303–314.

Henning Lange and Mario Bergés. 2016. BOLT: Energy Disaggregation by Online Binary Matrix Factorization of Current Waveforms. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys '16)*. ACM, NY, USA, 11–20.

A. Laurucci, S. Melzi, and M. Cesana. 2009. A reconfigurable middleware for dynamic management of heterogeneous applications in multi-gateway mobile sensor networks.. In *roceedings of the Seventh Annual Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON*. IEEE, rome, Italy, 1–3.

Sanja Lazarova-Molnar, Mikkel Baun Kjærgaard, Hamid Reza Shaker, and Bo Nørregaard Jørgensen. 2015. Commercial buildings energy performance within context occupants in spotlight. In *Smart Cities and Green ICT Systems (SMARTGREENS), 2015 International Conference on*. IEEE, 1–7.

Sanja Lazarova Molnar, Halldor Logason, Peter Grønbæk Andersen, and Mikkel Baun Kjærgaard. 2017. Mobile Crowdsourcing of Occupant Feedback in Smart Buildings. *SIGAPP Appl. Comput. Rev.* 17, 1 (May 2017), 5–14. https://doi.org/10.1145/3090058.3090060

Dan Li, Yuxun Zhou, Guoqiang Hu, and Costas J. Spanos. 2017. Optimal Sensor Configuration and Feature Selection for AHU Fault Detection and Diagnosis. *IEEE Trans. Industrial Informatics* 13, 3 (2017), 1369–1380.

Woong-Kee Loh and Young-Ho Park. 2014. A Survey on Density-Based Clustering Algorithms. In *Ubiquitous Information Technologies and Applications*. Springer, 775–780.

Clayton Millera, Zoltan Nagyb, and Arno Schlueter. 2017. A review of unsupervised statistical learning and visual analytics techniques applied to performance analysis of non-residential buildings. *Renewable and Sustainable Energy Reviews* In press (Jun 2017).

Thomas M. Mitchell. 1997. *Machine Learning* (1 ed.). McGraw-Hill, Inc., NY, USA.

Elena Mocanu, Phuong H. Nguyenv, Madeleine Gibescu, and Wil L. Kling. 2016. Deep learning for estimating building energy consumption. *Sustainable Energy, Grids and Networks* 6 (Jun 2016), 91–99.

Sanja Lazarova Molnar, Mikkel Baun Kjaergaard, Hamid Reza Shaker, and Bo Norregaard Jorgensen. 2015. Commercial Buildings Energy Performance within Context - Occupants in Spotlight. In *Proceedings of the 4th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS), 20-22 May*. IEEE, Lisbon, Portugal, 306–312.

Ahmed Nait Aicha, Gwenn Englebienne, and Ben Kröse. 2017. Unsupervised Visit Detection in Smart Homes. *Pervasive Mob. Comput.* 34 (2017), 157–167.

Kaoru Ota, Minh Son Dao, Vasileios Mezaris, and Francesco G. B. De Natale. 2017. Deep Learning for Mobile Multimedia: A Survey. *ACM Trans. Multimedia Comput. Commun. Appl.* 13, 3s (Jun 2017), 34:1–34:22.

Abdelraouf Ouadjaout, Noureddine Lasla, Djamel Djenouri, and Cherif Zizoua. 2016. On the Effect of Sensing-holes in PIR-based Occupancy Detection Systems. In *SENSORNETS 2016 - Proceedings of the 5th International Confererence on Sensor Networks, February 19-21, 2016.* Rome, Italy, 175–180.

Charith Perera, Chi Harold Liu, and Srimal Jayawardena. 2015. The Emerging Internet of Things Marketplace From an Industrial Perspective: A Survey. *IEEE Trans. Emerg. Top. Comput.* 3, 4 (Oct. 2015), 585–598.

Charith Perera, Yongrui Qin, Julio C. Estrella, Stephan Reiff-Marganiec, and Athanasios V. Vasilakos. 2017. Fog Computing for Sustainable Smart Cities: A Survey. *ACM Comput. Surv.* 50, 3, Article 32 (June 2017), 43 pages. https://doi.org/10.1145/3057266

Vasso Reppa, Panayiotis M. Papadopoulos, Marios M. Polycarpou, and Christos G. Panayiotou. 2015. A Distributed Architecture for HVAC Sensor Fault Detection and Isolation. *IEEE Trans. Contr. Sys. Techn.* 23, 4 (2015), 1323–1337.

Antonio Ridi, Christophe Gisler, and Jean Hennebert. 2015. Processing smart plug signals using machine learning. In *2015 Wireless Communications and Networking Conference Workshops, WCNC Workshops 2015, New Orleans, March 9-12, 2015.* IEEE, LA, USA, 75–80.

David S. Stoffer Robert H. Shumway. 2017. *Time Series Analysis and Its Applications: With R Examples, 4th Edition* (springer texts in statistics ed.). Springer.

Fisayo Caleb Sangogboye, Kenan Imamovic, and Mikkel Baun Kjærgaard. 2016. Improving occupancy presence prediction via multi-label classification. In *International Conference on Pervasive Computing and Communication (PerCom) Workshops*. IEEE, Sydney, Australia, 1–6.

Chayan Sarkar, Akshay Uttama Nambi S.N., and Venkatesha Prasad. 2016. iLTC: Achieving Individual Comfort in Shared Spaces. In *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks (EWSN '16)*. Junction Publishing, USA, 65 – 76. http://dl.acm.org/citation.cfm?id=2893711.2893723

James Scott, A.J. Bernheim Brush, John Krumm, Brian Meyers, Michael Hazas, Stephen Hodges, and Nicolas Villar. 2011. PreHeat: Controlling Home Heating Using Occupancy Prediction. In *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp 11)*. ACM, New York, NY, USA, 281–290.

Olli Seppanen, William J. Fisk, and David Faulkner. 2004. Control of Temperature for Health and Productivity in Offices. 111 (06 2004).

Oliver Shih, Patrick Lazik, and Anthony Rowe. 2016. AURES: A Wide-Band Ultrasonic Occupancy Sensing Platform. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys '16)*. ACM, NY, USA, 157–166.

Tomoaki Shoji, Wataru Hirohashi, Yu Fujimoto, and Yasuhiro Hayashi. 2014. Home Energy Management Based on Bayesian Network Considering Resident Convenience. In *International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*. IEEE, Durham, UK, 1–6.

Elahe Soltanaghaei and Kamin Whitehouse. 2016. WalkSense: Classifying Home Occupancy States Using Walkway Sensing. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys '16)*. ACM, NY, USA, 167–176.

Yusuf Sonmez, Ugur Guvenc, H Tolga Kahraman, and Cemal Yilmaz. 2015. A comperative study on novel machine learning algorithms for estimation of energy performance of residential buildings. In *Third International IEEE Conference on Smart Grid Congress and Fair (ICSG)*. IEEE, Istanbul, 1–7.

Fuchen Sun, Kar-Ann Toh, Manuel Grana Romay, and Kezhi Mao (eds.). 2014. *Extreme Learning Machines 2013: Algorithms and Applications*. Springer.

Zhongfu Tan, Jinliang Zhang, Jianhui Wang, and Jun Xu. 2010. Day-ahead electricity price forecasting using wavelet transform combined with ARIMA and GARCH models. *Applied Energy* 87, 11 (2010), 3606 – 3610. https://doi.org/10.1016/j.apenergy.2010.05.012

Emmanuel Munguia Tapia, Stephen S. Intille, and Kent Larson. 2004. Activity Recognition in the Home Using Simple and Ubiquitous Sensors. In *Pervasive Computing*, Alois Ferscha and Friedemann Mattern (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 158–175.

Jerome Friedman Trevor Hastie, Robert Tibshirani. 2013. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer.

Tim van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. 2008. Accurate Activity Recognition in a Home Setting. In *Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp '08)*. ACM, New York, NY, USA, 1–9. https://doi.org/10.1145/1409635.1409637

Yongcai Wang, Xiaohong Hao, Lei Song, Chenye Wu, Yuexuan Wang, Changjian Hu, and Lu Yu. 2014. Monitoring Massive Appliances by a Minimal Number of Smart Meters. *ACM Trans. Embed. Comput. Syst.* 13, 2s (Jan 2014), 56:1–56:20.

J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. 2009. Robust Face Recognition via Sparse Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31, 2 (Feb 2009), 210–227.

David P Wyon. 2004. The effects of indoor air quality on performance and productivity. *Indoor air* 14, 7 (2004), 92–101.

Jiang Xiao, Zimu Zhou, Youwen Yi, and Lionel M. Ni. 2016. A Survey on Wireless Indoor Localization from the Device Perspective. *ACM Comput. Surv.* 49, 2, Article 25 (June 2016), 31 pages. https://doi.org/10.1145/2933232

Jeonghee Yi, Tetsuya Nasukawa, Razvan Bunescu, and Wayne Niblack. 2003. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. IEEE, 427–434.

Dong Zhang, Shuhui Li, Min Sun, and Zheng O'Neill. 2016. An Optimal and Learning-Based Demand Response and Home Energy Management System. *IEEE Trans. Smart Grid* 7, 4 (2016), 1790–1801.

Yuxun Zhou, Dan Li, and Costas J. Spanos. 2015. Learning Optimization Friendly Comfort Model for HVAC Model Predictive Control. In *International Conference on Data Mining Workshop, ICDMW*. IEEE, Atlantic City, NJ, USA, 430–439.

Qingchang Zhu, Zhenghua Chen, and Yeng Chai Soh. 2015. Smartphone-based Human Activity Recognition in buildings using Locality-constrained Linear Coding. In *Industrial Electronics and Applications (ICIEA), 10th Conference on*. IEEE, Auckland, New Zealand, 214 – 219.

APPENDIX

A COMPARATIVE TABLES

Table 4. Summary of the occupancy solution.

Solution	Occupancy	Sensors Type	Sensor Data	Problem	ML tool	Eval Metric
[Khan et al. 2014]	binary, categories, exact number	phys, contex	motion, acoustic noise, light, humidity, meetings schedule, computer activity	class	KNN, SVM	accuracy
[Sangogboye et al. 2016]	binary	phys	motion	class	SVM	F-measure
[Ardakanian et al. 2016]	approximate	phys	single-pneumatic control, air flow, reheat	clus	agglomerative hierarchical	occupancy related metrics, energy saving
[Shih et al. 2016]	binary, exact number	phys	ultrasonic response	class, reg	generic classifiers, RT	mean error, FP, FN
[Soltanaghaei and Whitehouse 2016]	zone	phys	motion	class	DT	accuracy, energy saving, comfort
[Bales et al. 2016]	gender	phys	accelerometers	class	boosted DT, SVM, ANN	error (%)
[Khalil et al. 2016]	occupant ID	phys	ultrasonic	clust	DBSCAN	correct & error (%)

Table 5. Summary of the activity recognition solutions.

Solution	Activity Recognition	Sensors	Sensor Data	Problem	ML Tool	Eval Metric
[Hossain et al. 2017]	7 daily act	phys	motion, compass accelerometers	clust, class	K-means	accuracy (%), F-measure, correct class, distances on clusters
[Chiang et al. 2017]	25 daily act	phys	+70 of different types	class	SVM	accuracy
[Nait Aicha et al. 2017]	regular/irregular	phys	states of cabinets and doors, motion	class	MMPP	F-measure
[Carolis et al. 2015]	tasks in time (+600)	phys	high number, time	class	First-order logic learning	min, max, avg error in activities identifying
[Alhamoud et al. 2015]	9 daily act	phys	temp, light, motion	clust	K-means	distances on clusters
[Zhu et al. 2015]	5 daily act	virt	gyroscope, accelerometer	clust, class	LLC+K-means, SVM, KNN KELM	accuracy

Table 6. Summary of the preferences and behavior solutions.

Solution	Comfort Type	Preference Collection	Sensors Type	Sensors Data	Datasets	Problem	ML Tool	Eval Metric
[Ghahramani et al. 2015]	ther	voting	phys	temp, humidity	real	class	BN	accuracy, specificity
[Zhou et al. 2015]	ther	voting	phys, context	humidity, temp, air velocity, physiological conditions	real	class	CPLC	testing error
[Sarkar et al. 2016]	ther, vis	voting	phys	temp, light	artificial	reg	curve fitting	estimation error (lux), saved energy
[Barbato et al. 2010]	ther, vis	behav monitor	phys	temp, light, motion	artificial	clust	proposed	correct profile prediction (%)
[Antunes et al. 2013]	ther	behav monitor	phys	temp, power	real	reg	SVM	system throughput, detection of relevant patterns
[Shoji et al. 2014]	ther	behav monitor	phys	power	real	class	BN	accuracy, electricity consumption

Table 7. Summary of solutions for energy profiling and demand estimation.

Solution	Energy Profiling	Sensors Type	Sensors Data	Dataset	Problem	ML tool	Eval Metric
[Zhang et al. 2016]	appliances load	phys, context	HVAC load, weather, electricity price	real	reg	ANN, LR	error(%), energy, cost ($)
[Sonmez et al. 2015]	heating/cooling	no	BAI	real	reg	KNN, ANN	MAE, SD
[Fan et al. 2017]	cooling	phys	ambient and cooling water temp, humidity	real	reg	DL	MAE, RMSE, CV-RMSE
[Mocanu et al. 2016]	Building power	phys	power	real	reg	RBM	RMSE, R, p-value
[Chou and Ngo 2016]	smart grids connected buildings	phys, context	humidity, temp, power, appliance power, electricity saving alternatives	real	reg	ARIMA, SVR	R, RMSE, MAE, MAPE, MaxEA, TER, CPU time
[Iyengar et al. 2016]	city scale energy usage	phys	power	real	clus	K-means	/
[Lange and Bergés 2016]	appliances load	phys	current wave form	real	class	DL	F-measure, CPU time
[Jain et al. 2016]	building power	phys, contex	temp, light, power, weather, schedules, building data	real	reg	RT	NRMSE, Power consumption, peak reduction (%)
[Chandan et al. 2015]	cooling	phys	temp, humidity, CO2	real	reg	LR	average error, zone temperature
[Barbato et al. 2010]	heating light	phys	light, temp, PIR	artificial	clust	proposed	correct profile prediction (%)

Table 8. Summary of appliances profiling solutions.

Solution	Considered problem	Sensors Type	Sensors Data	Datasets	Problem	ML tool	Eval Metric
[Ridi et al. 2015]	appliances identification	phys	IoT-based smart plugs for every device	real real	class class	KNN, GMM, HMM	Accuracy
[Wang et al. 2014]	appliances' state (on/off)	phys	smart meters	art, real	class	HMM	metering noise
[Katarina et al. 2017]	anomaly detection in energy consumption	phys	power consumption	real	class	ANN, SVR, RF	AUC
[Ferdoash et al. 2015]	excessive airflow in HVAC	phys, virt	temp, weather	real	reg	SVM, LR	Accuracy
[Li et al. 2017]	failure detection of air handling unit	phys	depend upon features	real	class	QDA, LR, ANNs, MSVM	Accuracy

Table 9. Summary of solution for inferences on sensors

Solution	Sensor Inference	Sensors Type	Sensors Data	Datasets	Problem	ML tool	Eval Metric
[Hong et al. 2015]	type	phys	ambient and water temp, CO_2, humidity	real	class	RF, LR, SVM with RBF kernels	Accuracy, F-measure
[Gonzalez et al. 2016]	position, orientation	phys	thermopile array sensors	real	rule mining	association rules	Accuracy, support, confidence
[Gao et al. 2015]	type	phys	temp, motion, CO_2, power	real	class	RF, KNN, DT, SVM, LR, LDA, AdaBoost	Accuracy, re

B LIST OF DEFINITIONS

Linear Regression (LR). It consists in expressing, with a linear function, the relationship between a scalar dependent variable and one or more independent variable. The regression function is found by solving an optimization problem with the normal equations method while minimizing the empirical error. The latter is defined on the labeled data as the average distances between the estimated and empirical values. Mean square of differences or of absolute values of differences are the most used methods [Alpaydin 2014].

Logistic Regression. It aims at predicting a dependent variable value from a set of independent variables. The dependent variable in this case is binary, i.e., two classes, which makes it adapted to resolve binary classifications problems by estimating the probability of the event using the *Sigmoid function. Softmax regression* (or multinomial logistic regression) is a generalization for multi-class classification. The *Softmax* function (also known as the normalized exponential) receives as inputs scores (Logits) that are calculated in a similar way to linear regression and it returns probabilities for each target class. The high probability target class will be the predicted target class [Alpaydin 2014]. Logistic regression may be seen as one layer of ANN.

Artificial Neural Networks (ANN). An ANN is composed of a collection of interconnected nodes (neurons) that interact with each other. On each connection, a neuron can process a signal and transmit it downstream. Neurons states are represented by real numbers, generally between 0 and 1. The output at each neuron is called its activation value [Mitchell 1997]. Common activation functions, that maps neurons' inputs to activation outputs, are *Sigmoid*, *Softmax*, *RElU* (rectified linear unit), and *tanh*. The neurons are organized in layers, with an input, an output, and hidden layers.

There are two categories of topologies in ANN: (1) Feedforward ANN where the information flow is unidirectional throughout the layers with no feedback loops, which formes a directed acyclic graph. This category of ANN is used in pattern generation/recognition/classification. (2) FeedBack (recurrent) ANN, including feedback loops forming directed cycles in the topology graph. They are used in content addressable memories, (i.e., use internal memory to process arbitrary sequences of inputs).

Extreme Learning Machine. is a *feedforward ANN* where input weights and hidden nodes biases are randomly assigned and never updated. Only output weights (between the hidden nodes and the output layer) are learned in a single step. This method makes learning much faster and provides better performance compared to traditional feedforward ANN where all the parameters need to be learned[Sun et al. 2014].

Restricted Boltzmann Machine (RBZ). is a generative stochastic ANN that may be used either in supervised or unsupervised way. In RBZ, the neuron's binary activations are probabilistic. *RBZ* includes two layers, (1) the visible (input) layer, and (2) the hidden layer that performs a binary analysis. To facilitate the learning in *RBZ*, connection between the layers form a bipartite graph, i.e. no intra-layer connection[Trevor Hastie 2013].

Deep learning (DL). DL models use a cascade of nonlinear processing units that are organized into layers, where the output from every layer is plugged as an input into the next one. The problem dealt with steers the layering composition of the nonlinear processing units used in DL algorithms. Most of DL applications may be also viewed as a general form of ANNs, or in particular, the application to learning tasks of ANNs with several hidden layers. The learning may be supervised, partially supervised or even unsupervised (contrary to traditional ANNs).

Decision Trees (DT). They build classification or regression models in the form of a tree-like graph or a flowchart-like structure, where "tests" on attributes are represented by nodes, class labels (numerical data) by leaves, and the outcome of the test by branches. Carefully crafted questions on attributes of the test record are used in a series to feed the tree. The *C4.5* algorithm is the most used in decision trees. It adopts a top-down divide-and-conquer recursive approach. In every node, the algorithm chooses the best split among the features and possible split points. The split maximizing the normalized information gain is selected. This procedure is repeated for every node until reaching a "stop condition", which may be the minimum number of leaves in a node, the tree height, etc. [Alpaydin 2014]. The composition of multiple trees leads to more efficient algorithms such as *Random Forest* or *Gradient Tree Boosting*.

Random Forests. They are collections of independently trained decision trees outputting the mode class of the classes in case of classification, or the mean of the individual trees predictions in case of regression. Decision trees are unstable as they tend to overfit the training data, i.e., small changes in the data lead to largely different trees. This model provide more robust prediction compared to the use of single trees. The forest is called random because a random sample of the complete dataset is used to train each tree, which is known as "bootstrap aggregating or bagging". [Alpaydin 2014].

Support Vector Machine (SVM) and its Variants. It is a supervised ML model mostly used for classification, and sometimes for regression. Binary linear classification is performed by searching the hyper-plane that optimally differentiates two classes, i.e., that maximizes the distance to the nearest data point on each of its sides. Two parallel hyperplanes (that separate the two classes of data with maximum distance) can be selected if the training data are linearly separable. Non-linear classification can be assured by SVM and kernel trick with non-linear kernel functions. SVM may be generalized to multi-class SVM by reducing the single multi-class problem into multiple binary classification problems[Alpaydin 2014]. The are many variants of SVM, such as SVC (support vector clustering) that is used in unsupervised learning clustering problems, and SVR (support vector regression) that is used for regression[Trevor Hastie 2013].

AdaBoost and Gradient Boosting. These are boosting techniques that attempt to generate a strong classifier based on existing weak classifiers. This is done by building a model from the training data, then sequentially creating a new model that attempts to correct the previous errors. This process is repeated until the training set is predicted perfectly or a maximum number of models are added. AdaBoost (Adaptive Boosting) is a boosting algorithm developed for binary classification [Alpaydin 2014].

Stochastic gradient boosting machines are modern boosting methods built on AdaBoost. In Gradient boosting, each added model is trained to minimize the error by searching in the negative "gradient" direction. *Gradient boosting tree* incrementally builds sequences of simple trees by training each new instance.

Gradient Decent Algorithm. It is used to minimize a function defined by a set of parameters. An initial set of values of the parameters used and then the algorithm iteratively moves toward a set of values that minimizes the function. Steps in the negative direction of the gradient function are taken in the iterations. Gradient descent is commonly used to find parameters that minimize linear regression error [Alpaydin 2014].

K-Nearest-Neighbor (KNN) Algorithm. It is a method that does not require a training phase. The classification or the numerical value prediction for a new instance is made by searching through the entire dataset for the K closest instances using similarity measures (e.g., Euclidean distance). The output is summarized in these K nearest instances. For regression, the value of the new instance is the average on its K nearest neighbors. In classification, it is assigned to the class to which belong most of its K nearest neighbors (the mode class)[Alpaydin 2014].

Sparse Representation Classification (SRC). It is motivated by the fact that even though and object is in high-dimensional space, it can be reduced in a lower-dimensional subspace as long as it is presented as a sparse. A sparse is a vector or a matrix in which most of the elements are zeros.

By having few components that are different from zero, the object is decomposed as a linear combination of only few vectors, called atoms. The general framework of SRC is to use the linear combination of atoms to represent the object and calculate the representation coefficients of the linear representation system, and next use it to calculate the reconstruction residuals of each class. *SRC* is widely used for objects recognition in images[Wright et al. 2009]

K-means. It is an algorithm that divides a dataset into K clusters where each observation belongs to the cluster with the nearest center. K is an important input hyper-parameter on K-means[Alpaydin 2014].

Single-Link Clustering or *Single-Linkage Clustering.* It is a type of *hierarchical agglomerative clustering* where two clusters separated by the shortest distance are combined, and the distance between two clusters is defined as the one separating their two most similar members [Alpaydin 2014].

Bayesian Networks (BNs), *Belief Networks*, or *Probabilistic Directed Acyclic Graphical (DAG) Model.* BNs are defined as a DAG whose vertices represent random variables, which may be parameters or hypotheses, observable quantities, etc., and edges represent probabilistic conditional dependencies. Statistical and computational methods are usually used to estimate conditional dependencies. A probability function is associated to vertices that has as input for every vertex a set of values related to its parent variables, and it gives as output the probability of the variable the node represents. BNs combine principles from graph theory, algorithmic, statistics. They are used as underplaying model for many ML algorithms [Alpaydin 2014].

Hidden Markov Model (HMM). It is a probabilistic sequence model that associates a sequence of observations to a sequence of labels. it computes, for a given sequence, a probability distribution over possible sequences of labels for the purpose of select the best one. HMM is a Markov process with unobserved (i.e., hidden) states. Only the token output (that dependents on the state) is visible. HMM generates a sequence of tokens to inform about the sequence of states[Alpaydin 2014].

Time series. It consists in adding an explicit time-based order dependence between observations in a sapce of data. The Addition of the time dimension allows to, (1) identify the temporal nature of the phenomenon, and (2) predict values in future time slots. Generally, time series forecast two classes of components, i.e., trend and seasonality. The former is the appearance of a phenomenon that does not repeat, at the contrary of seasonality where an observation repeats in systematic intervals over time. Two types of *time-based models* may be distinguished, (1) ordinary regression models that use time values as x-variables, and (2) ARIMA models that relate past experience (values and prediction errors) to the present value of a series[Robert H. Shumway 2017].

Autoregressive Integrated Moving Average (ARIMA) Models. Autoregressive models are models that use the relationship between a variable in one period and a number of previous periods. Moving Average models measure the relationship between an observation and residual errors from previous periods. The term "Integrated" is to make the time series stationary (mean and variance stable over time). Both approaches (autoregressive and moving) are combined in ARIMA[Robert H. Shumway 2017].

Gaussian Mixture Models (GMM). It is a probability distribution of multiple normally distributed subpopulations within a larger population. Its density function is the weighted sum of subpopulations' densities. Generally, Expectation-Maximization (EM) algorithm is used to estimate GMM parameters in iterative ways. GMM are used to perform unsupervised clustering or feature extraction [Trevor Hastie 2013].

Genetic Algorithm (GA). It is a metaheuristic that belongs to Evolutionary Algorithms (EA) class. GA presents candidates solutions as genes while trying to maximize a fitness (or objective) function. It uses bio-inspired operators to chose members of the new generation, mutation and crossover to create new generations. This process of creation of new generations and selection is repeated until a stopping condition is reached, e.g., a certain number of iterations [Banzhaf et al. 1998].

hyper-parameters. They represent the prior known parameters that are distinguishable from the parameters learned by the algorithm during the training.

F-measure. is the harmonic mean of *precision* and *recall*:

$$F_1 = \frac{2}{\frac{1}{\mathrm{recall}} + \frac{1}{\mathrm{precision}}} \tag{2}$$

Precision is the fraction of correctly predicted values among the retrieved ones.*Recall* is the fraction of relevant predicted values over the total number of relevant values.

Accuracy. It is calculated by,

$$\frac{\text{Number of correct predictions}}{\text{Total number for predictions}} \tag{3}$$

Specificity and sensitivity. They are metrics used for binary classification, specificity is the true negative rate and sensitivity is the true positive rate.

Area Under the Curve (AUC). It is computed by,

$$AUC = mean_{o \in O_A, i \in I_A} \begin{cases} 1 & if Score(o) > Score(i) \\ 0.5 & if Score(o) = Score(i) \\ 0 & Score(o) < Score(i) \end{cases} \tag{4}$$

where, O is the set of all outliers, O_A is the set of outliers returned by the scenario A, and I_A is the set of inliers returned by the scenario A.