Research Article

Appliance Recognition in an OSGi-Based Home Energy Management Gateway

Federica Paganelli,[1] Francesca Paradiso,[2] Stefano Turchi,[2] Antonio Luchetta,[2] Pino Castrogiovanni,[3] and Dino Giuli[2]

[1]CNIT, University of Florence, Via Santa Marta 3, 50139 Florence, Italy
[2]Department of Information Engineering, University of Florence, Via Santa Marta 3, 50139 Florence, Italy
[3]Innovation & Industry Relations Research & Prototyping Telecom Italia, Via Guglielmo Reiss Romoli 274, 10148 Turin, Italy

Correspondence should be addressed to Stefano Turchi; stefano.turchi@unifi.it

The rational use and management of energy is considered a key societal and technological challenge. Home energy management systems (HEMS) have been introduced especially in private home domains to support users in managing and controlling energy consuming devices. Recent studies have shown that informing users about their habits with appliances as well as their usage pattern can help to achieve energy reduction in private households. This requires instruments able to monitor energy consumption at fine grain level and provide this information to consumers. While the most existing approaches for load disaggregation and classification require high-frequency monitoring data, in this paper we propose an approach that exploits low-frequency monitoring data gathered by meters (i.e., Smart Plugs) displaced in the home. Moreover, while the most existing works dealing with appliance classification delegate the classification task to a remote central server, we propose a distributed approach where data processing and appliance recognition are performed locally in the Home Gateway. Our approach is based on a distributed load monitoring system made of Smart Plugs attached to devices and connected to a Home Gateway via the ZigBee protocol. The Home Gateway is based on the OSGi platform, collects data from home devices, and hosts both data processing and user interaction logic.

1. Introduction

Energy conservation is considered a main challenge to be faced at national and international level. Several factors, such as climate change, the growing resource consumption rate, and availability of energy resources (raw materials), are making this challenge a priority. Riveiro et al. argued that "the challenge lies in finding technologies that reduce the energy consumption, while guaranteeing or even improving customer comfort levels and economic activity" [1]. Indeed, the technological progress in power efficiency is expected to produce a remarkable reduction in energy consumption, in both industrial and private home domains. In this context, the residential sector plays a significant role. For instance, in the European Union, it accounts for 30% of electricity usage. Some studies [2, 3] report that significant results in the energy consumption reduction in private households can be achieved through fine-grained monitoring of energy consumption and accurate provisioning of this information to consumers.

Home energy management systems (HEMS) have been introduced especially in private home domains to support users in managing and controlling energy consuming devices. Most adopted home energy monitoring techniques provide information on the whole energy consumption profile. Although this approach has the advantage of minimizing technological requirements and costs, it is hard to extract significant information for recognizing and predicting user habits and propose solutions for reducing energy wastage.

Fine-grained monitoring can provide more useful information, such as the consumption profile of specific appliances. In this way, customers could be kept informed of how much the total energy consumption is affected by the usage of a specific device and decide whether to replace it with a

more efficient one or just postpone its usage to a time with a less expensive fare. Moreover, the analysis of an appliance consumption for detecting anomalies could also help in recognizing possible malfunctions and undertaking actions to prevent additional appliance deterioration.

Fine-grained monitoring can be achieved through three main approaches.

(1) *Nonintrusive load monitoring (NILM)* [4]: NILM refers to a family of techniques that aim at recognizing the power consumption of a specific device from the whole-house consumption profile.

(2) *Hardware-based submetering*: this method consists in attaching a metering hardware module (e.g., a Smart Plug) onto each household appliance so that its energy consumption can be easily collected. This implies that a distributed system of low-cost metering devices has to be deployed in the house and data should be collected via the proper (wireless and/or wired) network infrastructure.

(3) *Use of smart appliances*: this approach relies on the adoption of household appliances equipped with sensing, processing and communication resources. Common examples are the last generation of so called "white goods" (e.g., refrigerator, washing machine, and oven) that can be remotely controlled and configured.

Although the adoption of smart appliances is expected to boost the effective and efficient implementation of energy saving policies, this is not likely to take place in the short term. In any case, regardless of the chosen technique both appliances classification and identification are essential for understanding their consumption characteristics and producing convincing energy saving applications.

As highlighted in two recent studies [2, 3], only very few products are currently present on the market and consumers tend to consider their prices too high. On the other hand, NILM approaches which are based on whole-house consumption information can be easily deployed by leveraging existing and widely adopted smart meters. Several load disaggregation and classification algorithms have been proposed in literature [5] to extract more meaningful information from the house aggregated load profile. Most of them need medium or high frequency monitoring data (at least 1 Hz frequency) to obtain accurate results. In real-world scenarios, this assumption may be resource demanding. Indeed, only few works have experimented the adoption of NILM algorithms in real home settings [6]. Therefore, the second approach (i.e., distributed load metering through Smart Plugs) can be considered a good candidate for gathering accurate per-device consumption profiles while keeping costs and resource requirements as low as possible [7].

In this paper we propose an approach for identifying the appliances in use within a home environment that analyzes energy consumption data collected by a distributed load monitoring system which is part of a home energy management system (HEMS) proposed by the Energy@Home Association and deployed in some real households in Italy. The

monitoring system is made of Smart Plugs attached to devices and connected to a Home Gateway via the ZigBee protocol [8]. The Home Gateway, based on the OSGi platform, collects data from devices connected to the home wireless network and hosts both data processing and user interaction logic. The algorithms described in this paper aim at simplifying the manual configuration phase of such a system, where the end user usually needs to add a label and a description to each Smart Plug to associate energy consumption data to a specific monitored device (e.g., washing machine, refrigerator, and dish-washing machine). Automatic appliance classification techniques can be used when a smart plug is moved from one device to one another to monitor different appliances in the house: in this case the system could ask the user to provide a description of the monitored device the first time it gets connected to a Smart Plug and then use data collected from the Smart Plug to automatically recognize the same device in a second time. The initial training phase where the user needs to add or confirm the label associated to a monitored device can be reduced and sometimes totally avoided if a database with energy consumption data associated to specific device categories or models is provided as an input to the system.

While existing works dealing with appliance classification delegate the classification task to a remote central server [7, 9], here we propose a distributed approach where data processing and appliance recognition is performed locally in the Home Gateway. Although this approach has the advantage of minimizing the communication requirements between local home premises and a central server, preserving privacy on user sensitive data and simplifying the user profile management, design and implementation choices should carefully take into account the peculiarities of the Energy@Home HEMS, especially in terms of storage and processing constraints. A major constraint of our application is the low frequency of data (1 sample collected every 2 minutes) which limits both type and number of features that can be extracted for analysis purposes. Thus, the original contribution of this work consists of a supervised classification algorithm based on artificial neural networks (ANN) [10] conceived for processing low-frequency monitoring data for recognizing appliances in use and their consumption profile. We also present a Java-based prototype implementation running as an OSGi bundle in the local Home Gateway.

This paper is structured as follows. In Section 2 we discuss related work and motivate our contribution. Section 3 introduces the architecture of the home energy management gateway, called Energy@Home. In Section 4 we describe the proposed ANN-based appliance classification algorithm and in Section 5 we describe its implementation in the OSGi-based Energy@Home system. Section 6 describes the test activities carried out to evaluate the performance of the implemented algorithm as well as its possible adoption in real-world settings.

2. Related Work

In the context of Smart Home, energy consumption monitoring is one of the most challenging topics. Basically,

the objective is to track individual appliances consumption to gain awareness of actual energy demands of household devices. This knowledge is found to be critical for strategic energy management while aiming at improving consumption efficiency.

In literature, two strategies are mainly adopted for pursuing load monitoring: intrusive and nonintrusive. Intrusive load monitoring, or ILM, requires the installation of a meter in association with the appliance to be monitored. An example is the so-called Smart Plug, which is placed in between the appliance plug and the wall socket. This strategy requires some kind of user involvement because household appliances must be properly equipped with meters. A different approach is nonintrusive load monitoring, or NILM, which is based on a meter that detects the whole-household consumption. Because of this, NILM techniques focus on discriminating the individual contributions by leveraging disaggregation strategies.

Nonintrusive load monitoring techniques have been theorized since the late 80s when Hart [4] proposed to measure the total power consumption of a household through the use of an electricity meter and disaggregate the measurement into partial consumptions due to various devices in use. In the last decade, several NILM techniques based on the use of machine learning approaches have been proposed, such as artificial neural networks (ANN) [6, 11], k-Nearest Neighbor Clustering, and Support Vector Machines [12]. In 2012 Kolter and Jaakkola [13] proposed a disaggregation technique based on unsupervised learning. This technique combines the use of Additive Factorial Hidden Markov Models for modeling the appliances behavior and the authors proposed a new inference algorithm, called additive factorial approximate MAP (AFAMAP), to separate appliances traces from the aggregated load data.

The advantage brought by the nonintrusiveness of NILM techniques methods decreases as the devices to be identified increase in number and type, due to several factors: low-consumption devices cannot be detected correctly [7], the proposed algorithms become increasingly computationally expensive, and devices with similar behavior are difficult to distinguish [5]. Thus, we have focused our research on the ILM category which, in our opinion, can suitably deal with the home energy management topic by achieving a compromise between satisfying classification accuracy results and low installation costs.

ILM approaches require the deployment of distributed sensors, such as the Plug [14], Plugwise [15] and Smart-Meter.KOM [16] platforms. These distributed sensing platforms support the fine-grained monitoring and control (e.g., switch on/off) of the connected devices but do not offer capabilities for identifying the type of attached device [7].

Reinhardt et al. [7] support autonomous configuration by presenting means to identify appliances based on their electric current consumption. Their distributed load monitoring system is based on embedded current monitoring devices, which collect current readings at a sampling rate of 1.6 kHz and extract ten features from them. A machine learning implementation maintains a model which matches these fingerprints to the learned appliance types. Results of the evaluations show that a very high classification accuracy of more than 98% can be achieved when the fingerprint of the in-rush current is regarded in addition to the features extracted from the steady state current waveforms. In a contemporary work [17] the same authors evaluate the accuracy of appliance identification based on the characteristic features of traces collected during the 24 hours of a day. They evaluate nine different classifiers using more than 1,000 traces of different electrical appliances' power consumptions achieving up to 95.5% accuracy rate for the Random Committee algorithm.

In 2011 Chen et al. [18] propose a statistical framework for disaggregation on coarse granular smart meter readings by modeling fixture characteristics, household behavior, and activity correlations. This framework has been implemented into two approaches for different application scenarios and has been deployed to serve over 300 pilot households in Dubuque, IA.

Ridi et al. [9] adopted a system based on low-cost Smart Plugs periodically measuring the electricity values and producing low-frequency (10^{-1} Hz) time series of measurements. They propose to use dynamic features based on time derivative and time second derivative features and they compare different classification algorithms including k-Nearest Neighbor and Gaussian Mixture Models. The best combination of features and classifiers shows 93.6% accuracy.

In [19], Gisler et al. present an open database of appliance consumption signatures populated with the power consumption signatures obtained using plug-based low-end sensors placed between appliance power plugs and electricity sockets. The database has been populated through two acquisition sessions of one hour for 100 different appliances distributed among 10 categories. In addition they propose two test protocols intended to compare the performance of different machine learning algorithms for appliance recognition where researchers can work on common data. In the first protocol, first session instances are used for the training set and the second session instances for the test set. In the second protocol, all instances of both acquisition sessions are successively taken in training and test sets by performing a k-fold crossvalidation.

Our work aims at proposing a classification algorithm capable of returning high accuracy levels taking advantage of very low-frequency metering data (about one real power value every two minutes). These data have been collected by a distributed metering system of Smart Plugs which is part of a home energy management system (HEMS) proposed by the Energy@Home Association [20] and promoted, among other companies, by Telecom Italia.

The low-frequency constraint has led us to design a classification system which finds its originality in the minimization of resources required for managing and storing energy data and in the use of low-cost monitoring devices. Moreover, while the majority of the proposed approaches carry experimentation activities in centralized servers, we present a Java-based implementation of the classification approach that runs directly in a HEMS at a customer home.
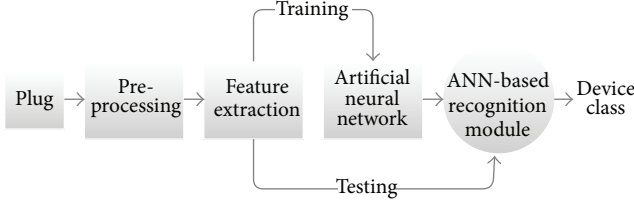
FIGURE 1: Appliance Classification System architecture.

TABLE 1: Feature extraction.

Extracted features
(1) Maximum power value.
(2) Minimum nonzero power value.
(3) Number of samples equal to zero.
(4) Number of samples less than or equal to 30 W.
(5) Number of samples between 30 and 400 W.
(6) Number of samples between 400 and 1000 W.
(7) Number of samples greater than 1000 W.
(8) Number of steady state consumption changes greater than 1000 W.
(9) Number of steady state consumption changes between 10 and 100 W.
(10) Average value of the nonzero power samples.

3. ANN-Based Algorithm for Appliance Recognition

The logical architecture of our classification system is shown in Figure 1. Raw power data are acquired by Smart Plugs—that is, metering devices standing in between the appliance power plug and the wall outlet—and collected by the home energy management system, described in the following Section.

The proposed Appliance Classification System is made of three main blocks: "preprocessing," "feature extraction," and "artificial neural network." An artificial neural network (ANN) is a massively parallel distributed processor that has a natural propensity for storing experimental knowledge [10]. In the context of this work we preferred a neural network approach with respect to other methods because of the low frequency, low numerosity, and jagged typology of available data. The algorithm is implemented in the "artificial neural network" block of Figure 1. The "feature extraction" block is the module where the experimental observation is processed in order to catch its most discriminating features (and labeled during the training phase), while the "preprocessing" block process raw or incomplete data to feed the feature extraction block with a uniform data structure.

The *preprocessing* block is responsible for collecting power data samples and for extracting significant power measures that will be given as input to the classification system. In order to obtain reliable results, we excluded all the power traces containing measuring faults (i.e., missing samples caused by disconnections). In addition, to fully exploit all the remaining traces, we configured an observation window set to the number of samples of the longest nonperiodic trace (100 samples). Consequently, devices characterized by hourly activities such as washing machines or dishwashers will often be represented with a full-length load trace; conversely, devices with a short or highly variable duration, such as microwave ovens or coffee machines, will have their power traces padded with zeros as observed in the original daily trace.

The *feature extraction* block processes the 100-sample trace provided by the preprocessing block to build a vector of features catching the peculiar characteristics of the power trace (e.g., the shape of the consumption profile, maximum peak, ascending or descending consumption steps, and duration). Table 1 shows the features selected by the empirical analysis of available measurement datasets.

The extracted features are provided in input to the ANN-based classification algorithm, which returns the recognized type of device.

The chosen neural paradigm for this application is a multilayer perceptron neural network with backpropagation algorithm (commonly called MLBPNN) [10], a well-known supervised model, used because of its simplicity and guaranteed convergence. This type of network is a universal approximator [21] based on the perceptron elementary neuron, that is, an information-processing unit that takes its origin in the biological counterpart and that can be mathematically described by the following pair of equations:

$$\text{net}_k = \sum_{j=1}^{m} w_{kj} x_j,$$
$$O_k = \psi\left(\text{net}_k + b_k\right) \quad k = 1, \ldots, N_{\text{out}}, \tag{1}$$

where O_k is the output of the kth neuron (if the used neurons are more than one, in the given number N_{out}); $x_1, \ldots, x_m$ are the input signals (with dimension given by m); $w_{k1}, \ldots, w_{km}$ are the synaptic weights of the neuron k; net_k is the weighted sum of the input signals; b_k is an external bias and ψ is the "activation function"; in this application the activation function is the bipolar sigmoid:

$$\psi\left(\text{net}_k\right) = \frac{2}{1 + e^{-\lambda \text{net}_k}} - 1, \tag{2}$$

which is chosen because of a couple of main reasons: (i) sigmoid is a nonlinear activation function that allows to quickly perform a classification of nonlinearly separable problems when used in multilayer structures; (ii) in addition, sigmoid allows to use well-known always convergent backpropagation algorithm as steepest descent [22] or Levenberg-Marquardt [23] during the learning process. The multilayer structure is a three-layer network with a number of neurons 10-30-8, respectively, in the input-hidden-output layers; the number of neurons in input and output layers is imposed by the nature of the problem, whereas the number of neurons in the hidden layer (30) is coming from an empirical choice, motivated by the best "trial and grow" result. In other words, the number
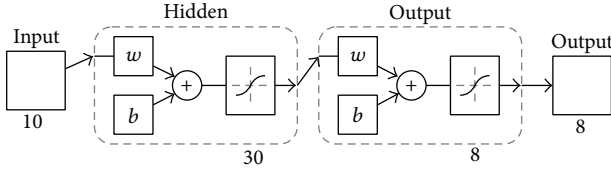
FIGURE 2: The artificial neural network configuration adopted for the experiment.

of neurons starts from a medium low number (around 10) and is increased until no further results improvement is observed. The ANN configuration is represented in Figure 2. The adopted error function is the classical mean squared of errors (MSE). As discussed in Section 5.2, we also performed some tests using Bayesian regularization; in this case the error function specified above is corrected with the mean squared of the network weights as regularization term. In order to avoid the generalization drawback, an early stopping approach is used [24] during the learning phase. Once the ANN block has been trained with the proper knowledge base, the ANN Recognition Model is ready to classify new power consumption traces. As detailed in the evaluation section, we exploited to this purpose the measurement data sets collected during the trials in the Telecom customers' sites.

4. Prototype Implementation

In this work we extended the home energy management system (HEMS) proposed and developed by the Energy@Home Association [25] with device load classification capabilities provided by the ANN-based Appliance Classification Systems. Energy@Home is a no-profit association participated by some major Italian companies (i.e., an electric utility company, telecommunications operators, and manufacturers) and research bodies that aim at developing and promoting technologies and services for energy efficiency at home based upon device to device communication.

4.1. Energy@Home Home Energy Management System. The Energy@Home HEMS (shown in Figure 3) is an OSGi-based software infrastructure for home automation and energy management based on a home area network (HAN) of objects that cooperate and communicate through the the ZigBee protocol [8]. Main cooperating objects are Smart Plugs, which have both sensing and actuating capabilities, and various types of smart devices and actuators, ranging from simple on-off devices (e.g., a TV-set) to configurable devices (e.g., air-conditioner) and smart appliances.

The OSGi [26] standard specifications define a Java-based service platform made by software modules (i.e., OSGi bundles) that can be installed, stopped, started, updated, and uninstalled at runtime. The OSGi Service Platform is made of a service execution framework and a set of standard service definitions that support the development of service-oriented applications in networked environments. This motivates its widespread adoption in the development of Home Energy Manamegement Systems, such as PeerEnergyCloud [27] and Ogema [28]. The OSGi framework provides a simple component model, a service registry, and utilities for dynamic service deployment, while the standard service definitions specify the interfaces and semantics for some reusable services (e.g., a logging service and an HTTP service) [29]. Services are implemented as bundles, which are Java archives that contain code, resources, and a manifest file with metainformation such as dependencies and activation. When the bundle is active, it can publish its services or discover and bind itself to services provided by other bundles through the service registry. The HEMS OSGi Service Platform hosts a set of web applications providing users with home automation and energy consumption awareness capabilities.

A typical home-based deployment of the Energy@Home system is made of the following hardware components.

(1) A *Smart Info* device provides end users with the certified information on electricity consumption managed by the electronic smart meter. It can be plugged in every domestic socket to collect data from the smart meter leveraging powerline communication. The Smart Info can be provided by the distribution system operator (DSO). Published data are a subset of those already made available by the home electricity meter; hence the Smart Info acts like a proxy in respect of the meter.

(2) *Smart Appliances* are white goods (e.g., dishwasher and washing machine) that have local intelligence and networking capabilities. They can provide information on their energy consumption (e.g., used energy and instant power), respond to remote commands, and interact with the user through a GUI.

(3) The *Home Gateway* is the core of the Energy@Home HEMS. It is the centralized management component that connects the home area network with external application services via wide area network (WAN) connectivity. It is based on a modular and highly configurable OSGi framework and hosts application logic modules. It offers multiple network interfaces, including a HAN interface to communicate with the abovementioned home devices via the ZigBee protocol, a home network (HN) interface to interconnect additional local devices, such as PCs and TV via wired and wireless LAN, and a WAN interface used to communicate with remote service providers' systems (through xDSL connection). In addition, it provides local service logic and remote services with high-level APIs for discovering, managing, and communicating with HAN devices. The protocol used for the communication between the HAN devices and the Home Gateway is based on ZigBee, since it is a low-cost, low-power-consumption, two-way, wireless communication standard [30]. ZigBee can be used in different application domains (e.g., home automation, healthcare, energy management, and telecom services) and a set of extensions have been designed for the Energy@Home system [31] and integrated in the
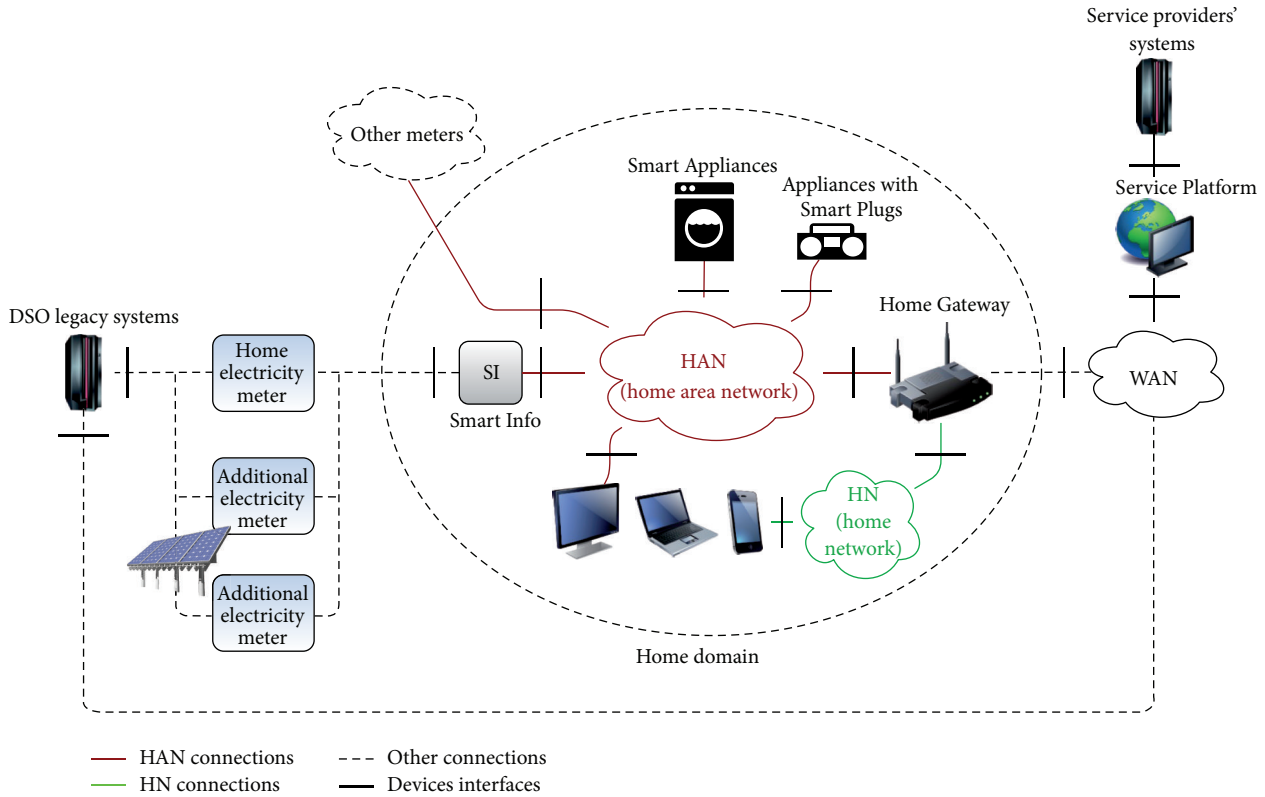
FIGURE 3: Energy@Home home energy management system architecture.

version 1.2 of the Home Automation profile specification, ratified by the ZigBee Alliance in the second half of 2013.

(4) *Cloud Services*: the Home Gateway interacts with a remote Service Platform hosted in the telecommunication operator data centers providing storage and processing capabilities. The Service Platform collects and stores the data sent by the Home Gateway. It can host applications that perform statistical and analytics processing on historical data for providing users with consumption awareness, predictions, and possible suggestions for consumption reduction. The Service Platform can also host downloadable application bundles that can be selected by users and eventually run on a local home service gateway.

This HEMS has been deployed in 20 private homes in Italy and both experimentation and data collection are currently ongoing. Several types of monitoring approaches (whole-house monitoring, real-time monitoring of identified devices through smart appliances, and low-frequency monitoring of unidentified devices) are in place, thus allowing the experimentation of different data analysis and service provisioning approaches.

4.2. Home Gateway Architecture. The Home Gateway is an OSGi-based system made by several bundles, whose interaction is realized through service provision and consumption. It is made of a set of key bundles (i.e., the *Java-Gateway Abstraction Layer*, the *Home Automation Core*, and the *Local Gateway Service Logics* bundles) supporting the development and deployment of energy monitoring and management applications. Hereafter we will briefly describe the features provided by these Home Gateway components and subsequently we will focus on the classification algorithm implementation.

The *Java-Gateway Abstraction Layer* (Java-GAL) implements the ZigBee Gateway Device specifications for managing ZigBee Networks and guaranteeing interworking with IP [32]. Java-GAL implements the features required to perform active nodes discovery in order to keep a constantly updated image of the current ZigBee network. It also provides an abstraction layer that allows applications to control and access ZigBee devices hiding low-level implementation details. To this purpose, it translates the ZigBee product-dependant low level APIs into a set of HTTP/REST and local APIs that can be invoked by other bundles as well as by external applications to control ZigBee devices.

The *Home Automation Core* offers a high-level API exposing attributes and commands defined by ZigBee Home Automation 1.2 Service Clusters. It also implements a basic web-based interface allowing the users to configure and test the home automation system (e.g., for adding and configuring new ZigBee devices).

Local Gateway Service Logics: application bundles that exploit the high-level Home Automation Core APIs for providing users with value-added services for energy management and home automation.

The Home Gateway has been implemented by leveraging the OSGi Equinox 3.5.2 Framework. More detailed specifications of the Home Gateway supported ZigBee Service Clusters can be found at [31] and the Home Gateway source code is available online [33].

4.3. Classification Algorithm Implementation. We implemented the ANN-based classification algorithm as an OSGi bundle, called Classifier bundle, deployed in the Home Gateway as one of the *Local Gateway Service Logics* OSGi bundles.

The software logic that implements the classification algorithm contains the following main classes.

 (1) *ILM_Model*: it is the class that handles the information used to configure the neural network. This information can be provided as input as an xml file and specifies the number of inputs, hidden nodes, and outputs of the network and, for each input row, the minimum and maximum values of the allowed range values, which are used in the preprocessing phase to normalize data entering the ANN.

 (2) *ILM_Classifier*: it is the class that implements the ANN-based classification capability. The class offers the methods for training and testing the ANN, saves it as a file, and, finally, classifies input data. The class has been implemented by relying on a Java-based framework for neural networks, Neuroph [34], that we wrapped in an OSGi bundle and deployed in the Home Gateway.

5. Evaluation

In this section we describe the testing activities carried out for evaluating the proposed ANN-based approach for low-frequency distributed load monitoring.

This section is structured as follows: first, we introduce the dataset used for testing and explain how data have been acquired from an experimentation campaign in real households, subsequently, we explain the performance evaluation procedure of the proposed algorithm.

5.1. Data Collection from Home Trials. Data collected from a trial of the Energy@Home system carried out in 2012 have been put together to form the datasets used for the test. The trial included 10 private houses of collaborating italian customers where the Energy@Home HEMS was deployed in, with the following configuration: (i) a Smart Info device connected to the home electricity meter; (ii) many Smart Plugs to collect consumption information from connected loads, such as washing machine; refrigerator; dishwashing machine; smart TV; iron; microwave oven; lighting stuff; coffee machine.

Energy consumption data extrapolated from the activity of these devices are used by the Home Gateway and the

Service Platform to implement use cases with the aim of enhancing customer awareness of energy consumption [35]. In order to meet the trial application requirements, energy and instantaneous power data are collected from HAN devices and stored in the remote Service Platform database so that customers are provided with historical and statistical information on their energy consumption. Stored data include whole in-house consumption from the power meter (Smart Info) and single device energy information coming from Smart Appliances and Smart Plugs.

The HAN devices are connected to the Home Gateway via ZigBee protocol and the Home Gateway uses the reporting strategy defined in the ZigBee Cluster Library specification [36] to receive from each of them the following consumption information: (i) the summed value of energy delivered and consumed in the premise (Smart Info) or by a specific device (Smart Plugs and Smart Appliances); (ii) the instantaneous real power absorbed by the whole house (Smart Info) or by a specific device (Smart Plugs and Smart Appliances).

The reporting parameters configured on each Smart Plug and Smart Appliance provide real-time instantaneous power information: every change in instantaneous power that is greater than or equal to 5 W is notified to the Home Gateway with a maximum configured delay of 2 seconds. Subsequently, the Home Gateway processes these data to provide users with real-time information. Please note that all these measurements are not directly stored in the platform database. Indeed, to avoid storage overloads, only a subset of these data filtered by the Home Gateway is retained. Information is stored on the basis of the reporting of summed energy values sent every 2 minutes by each device: for each of these time intervals, the gateway calculates the device's energy consumption (Wh) and stores this value along with the minimum and maximum instantaneous power values (W) pertaining to the same time interval.

These low-frequency measurements are stored in the remote Service Platform database and are used to test and validate the appliance identification algorithm described in this paper. Table 2 shows an excerpt of the database records whose labels have the following meaning:

 (1) *appliance_id*: a unique identifier of the monitored appliance, associated to a specific appliance category. This information is required in the dataset to test and evaluate the accuracy of the proposed classification algorithm and has been manually provided by the users involved in the trial. In real-world cases it is obviously unknown;

 (2) *start_time*: the start time of the time interval associated to this record, expressed as the difference (in milliseconds) between the current time and midnight, January 1, 1970 UTC;

 (3) *duration*: the time interval duration (in milliseconds);

 (4) *energy*: the energy consumption (Wh) of the device in the time interval;

 (5) *min_power* and *max_power*: minimum and maximum values of the instantaneous power measures (W) reported by the device in time interval;

TABLE 2: Excerpt of a refrigerator's records stored in the database.

| appliance_id	start_time	duration	energy	min_power	min_power_time	max_power	max_power_time
746	1361752107117	120258	0	0	1361752167156	0	1361752167156
746	1361752227375	120733	0	0	1361752287691	0	1361752287691
746	1361752348108	120598	0	1	1361752408367	1	1361752408367
746	1361752468702	120754	0	1	1361752529172	1	1361752529172
746	1361752589456	120698	0	0	1361752649774	0	1361752649774
746	1361752710154	120717	0	0	1361752770566	0	1361752770566
746	1361752830871	120682	1	0	1361752891322	0	1361752933525
746	1361752951553	120743	2	58	1361752951649	72	1361752985833
746	1361753072296	120821	2	70	1361753106463	70	1361753106463
746	1361753193117	120585	3	69	1361753227186	69	1361753227186
746	1361753313702	120764	2	69	1361753347915	69	1361753347915
746	1361753434466	120628	2	69	1361753468611	69	1361753468611
746	1361753555094	120866	1	0	1361753603392	68	1361753589449
746	1361753675960	120658	1	0	1361753724209	0	1361753724209
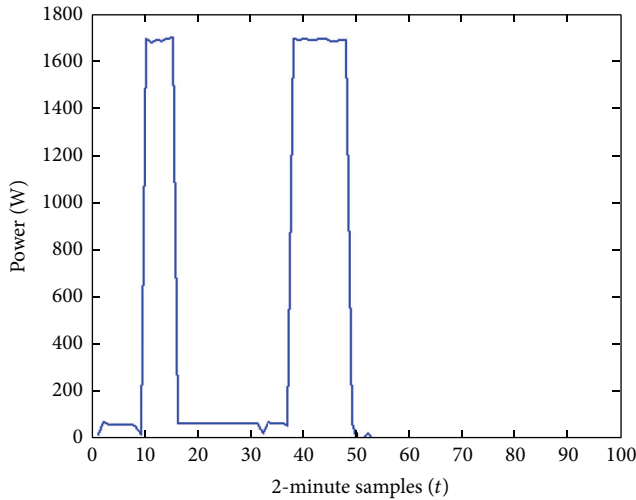
FIGURE 4: A dishwasher power consumption trace.
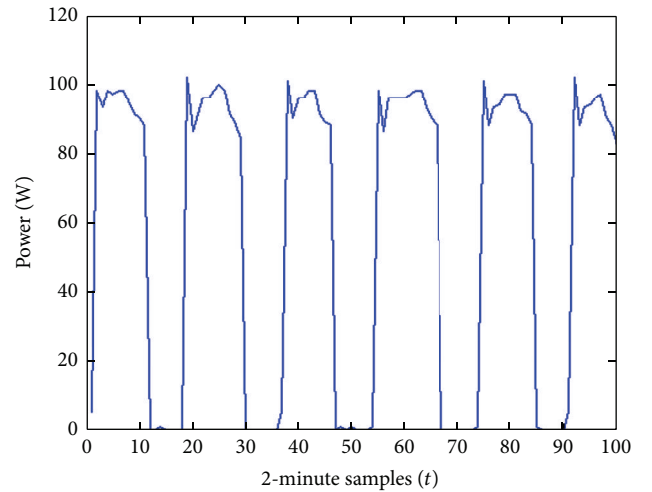
FIGURE 5: A refrigerator power consumption trace.

(6) *min_power_time* and *max_power_time*: the time of sampling of the associated power field (same format as start_time).

As an example, Figures 4 and 5 show the average energy/power consumption evaluated every 2 minutes interval for a dishwasher and a refrigerator, respectively.

5.2. Performance Test. In this section we present the test cases carried out for the accuracy estimation of our classification algorithm. The software used for testing is Matlab version R2012a running on a machine having an Intel Core2 Duo CPU T7500 at 2.20 GHz, 2 GB RAM.

The test cases have been led using a new dataset created from a subset of the data collected during the Energy@Home trials. Of all the available devices, we selected the ones that were present in all of the three houses (washing machine; refrigerator; dishwashing machine; smart TV; iron;

microwave oven; lighting stuff; coffee machine) and for each type of device we extracted the same number of traces.

This new dataset is made of 528 examples (66 for each of the 8 monitored devices) and contains the first 100 samples of each device load curve of power, measured in Watt. Signatures are entered according to the following criterion: given a device, we consider as a signature the trace starting from the first nonzero sample of the daily consumption curve and ending with the hundredth sample, regardless of the duration of the consumption itself. If the consumption trace exceeds the hundredth sample, it is simply trimmed, while shorter traces are expanded by padding them with zero samples. Finally, these signatures undergo the features extraction process.

This criterion has been defined in order to preserve the nature of the original daily traces belonging to the dataset collected in the home trials; in fact, the appliances consumption is represented by power samples separated by

TABLE 3: Classification accuracy for each device type and test iteration.

	Correct classification percentage (true positive)					
	Test 1	Test 2	Test 3	Test 4	Test 5	Total
Washing machine	100	100	100	100	98.48	*99.70*
Refrigerator	92.42	100	100	100	96.96	*97.88*
Dishwasher	100	100	100	100	100	*100*
Smart tv	69.69	84.84	84.84	90.90	81.81	*82.42*
Iron	90.90	89.39	95.45	90.90	86.36	*90.60*
Microwave oven	100	100	100	100	100	*100*
Lighting	90.90	90.90	93.93	93.93	90.90	*92.11*
Coffee machine	98.48	100	100	98.48	100	*99.39*
Overall accuracy	*92.80*	*95.64*	*96.78*	*96.78*	*94.31*	**95.26**

zeros (no consumption detected). Moreover, we considered that a 100-sample window (the longest nonperiodic trace duration) corresponding to a 200-minute time interval would be sufficient for exhaustively representing a device signature of the dataset. Zero-padding has been performed on traces characterized by less than 100 samples for keeping them accordant with the original and isolating them for proper classification. The trimming operation has been basically performed on "always-on" devices traces characterized by a periodical consumption (i.e., fridge and refrigerator); for such devices zero-padding the end of a single signal period could cause a significant loss of information while using a properly set trimming frame contributes to maintain the original signature shape.

This dataset was divided into three parts; the 70% of the examples have been allocated to train the ANN, the 15% have been used for validation, and the last 15% have been used for testing purposes. In order to validate the effectiveness of the algorithm we have used the "overall accuracy" index in terms of percentage.

Figure 6 shows the overall confusion matrix resulted from a single classification test of the abovementioned 528-sample dataset (66 per device). The matrix rows and columns represent the output and the target classes, respectively; in the diagonal the correctly classified samples for each device are reported. Observing the iron (E) as an example, the ANN has correctly classified 56 out of the 66 samples; the remaining 10 samples have been misclassified 7 times with the smart TV (D) and 3 times with the refrigerator (B).

Table 3 shows the classification results for each device and for five test iterations, where each iteration includes an independent training phase and, consequently, a different ANN configuration. The final column shows the average accuracy value for each device. The overall accuracy row shows the algorithm's successful recognition percentage for all the devices.

The relevant fact that emerges from Table 3 is the high accuracy achieved for each device mostly regarding the dishwasher, washing machine, and coffee machine. Conversely, devices such as smart TV, iron, and lighting can be misclassified because of their usage duration variability.

The results presented so far have been achieved while trying to classify traces belonging to devices whose signatures

Target class

	A	B	C	D	E	F	G	H	TOT
A	66	0	0	0	0	0	0	0	100
B	0	66	0	1	3	0	0	0	94.3
C	0	0	66	0	0	0	0	0	100
D	0	0	0	52	7	0	4	0	82.5
E	0	0	0	12	56	0	0	0	82.4
F	0	0	0	0	0	66	0	1	98.5
G	0	0	0	1	0	0	62	0	98.4
H	0	0	0	0	0	0	0	65	100
TOT	100	100	100	78.8	84.4	100	93.9	98.5	94.5

(Output class — vertical axis label)

A: washing machine   E: iron
B: refrigerator   F: microwave oven
C: dishwasher   G: lighting
D: smart TV   H: coffee machine

FIGURE 6: A confusion matrix sample resulting from a single classification test.

have been used in the training phase. From an application point of view, this would call for a preliminary collection activity where the user is requested to associate a signature with the proper label. Unfortunately, such procedure cannot be implemented when electric power consumption data are used for autonomous classification of appliances in different categories. Therefore, we tried to evaluate the accuracy of the ANN in recognizing devices whose signatures were not used in the training phase.

A second test case consisted of testing the ANN using both a Levenberg-Marquardt (LM) and Bayesian regularization (BR) configurations with power traces collected from a new house. To this purpose, we selected the devices monitored in all the houses, that is, washing machine, refrigerator, dishwasher, and microwave oven.

Table 4: Classification accuracy for previously unknown devices.

| | Average correct classification % | | | |
| | 10 features | | 6 features | |
	LM	BR	LM	BR
Washing machine	56.93	8.3	98.4	98.32
Refrigerator	4.38	11.5	83.73	99.52
Dishwasher	6.32	1.5	16.13	24
Microwave oven	37.47	11.6	44.15	0

Table 4 shows the test results: although these appliances are completely unknown to the ANN, the washing machine and the smart TV have been detected albeit with medium and low accuracy values, which decrease dramatically for both refrigerator and dishwasher. While comparing results obtained with a LM network with the ones coming from a BR network, it is evident how the latter slightly improves the identification of those appliances whose time of use is considerable, while dramatically worsens the identification of the ones whose consumption has typically a short duration and an on/off behavior and thus cannot be easily characterized at such low monitoring frequency (e.g., microwave oven).

The result of the second test case helped us in improving the features extraction criterion. According to the results in Table 4, the ten features chosen to characterize the load curve appear to be overspecific. Therefore, we reduced their number to six, by choosing the more general ones (i.e., features 1-2 and 7–10 in Section 3). Overly specific features, such as the number of samples between a minimum and a maximum value (features 3–6 in Section 3), could in fact overdescribe a power consumption signal worsening in generalization. In other words, we selected those features that better describe device classes rather than particular usage patterns: for example, two smart TV traces can differ significantly with respect to the usage time, still preserving typical characteristics. If we include the number of zero samples (feature 3) as a feature, we will force the ANN to consider a usage pattern as a possible discriminating characteristic resulting in an unnecessary complication of the classification task.

Since the classification OSGi bundle has been designed to be executed on the Home Gateway, we carried out some tests for estimating its time performance on a typical HEMS deployment.

The Home Gateway used for these tests is equipped with an ARM Processor 400 MHz, 1 GB DDR 200 MHz RAM and 2 Gbit NAND Flash memory. The operating system is Linux 2.6.35 with Java SE for Embedded 6 (Java ARMv5 Linux Headless) and the Equinox 3.5.2 OSGi Framework installed.

Regarding the machine learning part, the ANN has the same configuration as the one detailed above. The tests have been carried out using the measures coming from the Smart Plugs preprocessed by taking 100 samples and trimming or zero-padding the curves as needed. Again, the signatures obtained at this stage undergo a feature extraction process to obtain a vector of 10 discriminating features.

For each type of device (washing machine; refrigerator; dishwashing machine; smart TV; iron; microwave oven; lighting stuff; coffee machine), 10 preprocessed power traces are entered in the system to measure the output time. The average classification time was found to be 4 ms. Moreover, the average time needed to load and run the ANN has been measured to be 269 ms. This is meaningful because the ANN could also be changed when a new neural network trained with more examples is available, and this would require a complete reload.

6. Conclusions

In this paper we presented a neural network based system for appliance classification. With respect to the state of the art, our contribution is focused on the following: (1) the analysis of low frequency metering data (i.e., 1 active power sample every 2 minutes) and (2) the implementation and testing of the algorithm in a Home Gateway. Adopting low frequency data has the major advantages of allowing the exploitation of cheap Smart Plugs and saving resources for storing, processing, and transmitting data. The latter is even more appealing if we consider to run the algorithm in the Home Gateway instead of a central server.

The algorithm was tested using data coming from real households with encouraging results. We experimented good classification accuracy for the ANN trained with examples coming from similar devices (i.e., from same device, producer, and model). From the user perspective, this approach can cause a significant discomfort due to the mandatory manual labeling of household appliances. To overcome this problem we envision a remote repository storing devices load profiles for future uses. In this way, a knowledge base is incrementally built and maintained thanks to users contributions. A possible use of the knowledge base is at configuration time: when a new system is deployed in a house, the knowledge base is queried to find the models of appliances present in the house. Therefore, the ANN training can be personalized selecting the best power trace examples for that case. In the long run, when the knowledge base contains a significant number of different traces, this solution would require only a minimum configuration effort for the user (who should just list the brand and model of the appliances to be monitored).

In the future, it would be useful to evaluate the use of different neural models characterized by dynamic retraining mechanisms in order to improve the balance between efficiency and complexity [37]. Another direction for future work would be to extend the proposed approach by exploiting information about user habits and daily activity patterns that can be extracted from sensors observation data through complex event processing techniques [38, 39].

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

References

[1] M. Riveiro, R. Johansson, and A. Karlsson, "Modeling and analysis of energy data: state-of-the-art and practical results from an application scenario," Tech. Rep. HS-IKI-TR-11-002, 2011.

[2] S. Darby, *The Effectiveness of Feedback on Energy Consumption: A Review for DEFRA of the Literature on Metering, Billing and Direct Displays*, vol. 486, 2006.

[3] K. Ehrhardt-Martinez, K. A. Donnelly, S. Laitner et al., *Advanced Metering Initiatives and Residential Feedback Programs: A Meta-Review for Household Electricity-Saving Opportunities*, American Council for an Energy-Efficient Economy, Washington, DC, USA, 2010.

[4] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[5] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Nonintrusive load monitoring approaches for disaggregated energy sensing: a survey," *Sensors*, vol. 12, no. 12, pp. 16838–16866, 2012.

[6] A. G. Ruzzelli, C. Nicolas, A. Schoofs, and G. M. P. O'Hare, "Real-time recognition and profiling of appliances through a single electricity sensor," in *Proceedings of the 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '10)*, pp. 1–9, IEEE, June 2010.

[7] A. Reinhardt, D. Burkhardt, M. Zaheer, and R. Steinmetz, "Electric appliance classification based on distributed high resolution current sensing," in *Proceedings of the 37th IEEE Conference on Local Computer Networks Workshops (LCN Workshops '12)*, pp. 999–1005, Clearwater, Fla, USA, October 2012.

[8] "Zigbee alliance protocol," IEEE Standard 802.15.4, 2006.

[9] A. Ridi, C. Gisler, and J. Hennebert, "Automatic identification of electrical appliances using smart plugs," in *Proceedings of the 8th International Workshop on Systems, Signal Processing and Their Applications (WoSSPA '13)*, pp. 301–305, May 2013.

[10] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall PTR, 1994.

[11] H.-H. Chang, C.-L. Lin, and J.-K. Lee, "Load identification in nonintrusive load monitoring using steady-state and turn-on transient energy algorithms," in *Proceedings of the 14th International Conference on Computer Supported Cooperative Work in Design (CSCWD '10)*, pp. 27–32, Shanghai, China, April 2010.

[12] T. Onoda, G. Raetsch, and K.-R. Müller, "Applying support vector machine and boosting to a non-intrusive monitoring system for household electric appliances with inverters," in *Proceedings of the 2nd International ICSC Symposium on Neural Computation (NC '00)*, 2000.

[13] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 1472–1482, 2012.

[14] J. Lifton, M. Feldmeier, Y. Ono, C. Lewis, and J. A. Paradiso, "A platform for ubiquitous sensor deployment in occupational and domestic environments," in *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks (IPSN '07)*, pp. 119–127, April 2007.

[15] B. Plugwise, "Plugwise circle," 2010, https://www.plugwise.com/circle.

[16] A. Reinhardt, D. Burkhardt, P. S. Mogre, M. Zaheer, and R. Steinmetz, "SmartMeter.KOM: a low-cost wireless sensor for distributed power metering," in *Proceedings of the 36th Annual IEEE Conference on Local Computer Networks (LCN '11)*, pp. 1032–1039, IEEE, October 2011.

[17] A. Reinhardt, P. Baumann, D. Burgstahler et al., "On the accuracy of appliance identification based on distributed load metering data," in *Sustainable Internet and ICT for Sustainability (SustainIT)*, pp. 1–9, IEEE, 2012.

[18] F. Chen, J. Dai, B. Wang, S. Sahu, M. Naphade, and C.-T. Lu, "Activity analysis based on low sample rate smart meters," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)*, pp. 240–248, ACM, August 2011.

[19] C. Gisler, A. Ridi, D. Zujferey, O. A. Khaled, and J. Hennebert, "Appliance consumption signature database and recognition test protocols," in *Proceedings of the 8th International Workshop on Systems, Signal Processing and Their Applications (WoSSPA '13)*, pp. 336–341, IEEE, 2013.

[20] T. E. T. Team, "Energy@home: a user-centric energy management system," 2010, http://www.energy-home.it/.

[21] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[22] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, MIT Press, Cambridge, Mass, USA, 1986.

[23] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.

[24] R. Gençay and M. Qi, "Pricing and hedging derivative securities with neural networks: bayesian regularization, early stopping, and bagging," *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 726–734, 2001.

[25] Energy@Home, http://www.energy-home.it.

[26] O. Alliance, "Osgi alliance specifications," 2013, http://www.osgi.org/Specifications.

[27] C. Busemann, V. Gazis, R. Gold et al., "Integrating sensor networks for energy monitoring with service-oriented architectures," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 737680, 12 pages, 2013.

[28] "Open gateway energy management alliance," http://www.ogema.org/ogema/index.html.

[29] R. S. Hall and H. Cervantes, "Challenges in building service-oriented applications for OSGi," *IEEE Communications Magazine*, vol. 42, no. 5, pp. 144–149, 2004.

[30] A. Ranalli and C. Borean, "Energy@ home leveraging zigbee to enable smart grid in residential environment," in *Smart Grid Security*, pp. 132–149, Springer, 2013.

[31] T. E. T. Team, "Energy@home technical specification v.0.95," 2012, http://www.energy-home.it/Documents/Forms/AllItems.aspx.

[32] Z. Alliance, "Zigbee network device gateway specification," 2011, http://zigbee.org/Standards/ZigB.

[33] "Jemma-java energy management application framework," http://ismb.github.io/jemma/.

[34] "Neuroph," http://neuroph.sourceforge.net/index.html.

[35] T. E. T. Team, "Energy@home use case v.1.2," 2010, http://www.energy-home.it/Documents/Forms/AllItems.aspx.

[36] Z. Alliance, "Zigbee cluster library specification," ZigBee Document 075123r01, 2007.

[37] A. D. Doulamis, N. D. Doulamis, and S. D. Kollias, "On-line retrainable neural networks: improving the performance of neural networks in image analysis problems," *IEEE Transactions on Neural Networks*, vol. 11, no. 1, pp. 137–155, 2000.

[38] I. Zappia, F. Paganelli, and D. Parlanti, "A lightweight and extensible complex event processing system for sense and respond applications," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10408–10419, 2012.

[39] I. Zappia, D. Parlanti, and F. Paganelli, "LiSEP: a lightweight and extensible tool for complex event processing," in *Proceedings of the IEEE International Conference on Services Computing (SCC '11)*, pp. 701–708, Washington, DC, USA, July 2011.