

Article

A Feature-Based Model for the Identification of Electrical Devices in Smart Environments

Andrea Tundis ^{1,*}, Ali Faizan ² and Max Mühlhäuser ¹

¹ Department of Computer Science, Technische Universität Darmstadt, Hochschulstrasse 10, 64289 Darmstadt, Germany; max@tk.tu-darmstadt.de

² Software AG, Uhlandstraße 12, 64297 Darmstadt, Germany; ali.faizan@softwareag.com

* Correspondence: tundis@tk.tu-darmstadt.de; Tel.: +49-6151-16-23199

Received: 5 April 2019; Accepted: 5 June 2019; Published: 8 June 2019



Abstract: Smart Homes (SHs) represent the human side of a Smart Grid (SG). Data mining and analysis of energy data of electrical devices in SHs, e.g., for the dynamic load management, is of fundamental importance for the decision-making process of energy management both from the consumer perspective by saving money and also in terms of energy redistribution and reduction of the carbon dioxide emission, by knowing how the energy demand of a building is composed in the SG. Advanced monitoring and control mechanisms are necessary to deal with the identification of appliances. In this paper, a model for their automatic identification is proposed. It is based on a set of 19 features that are extracted by analyzing energy consumption, time usage and location from a set of device profiles. Then, machine learning approaches are employed by experimenting different classifiers based on such model for the identification of appliances and, finally, an analysis on the feature importance is provided.

Keywords: electrical devices; classification; energy management; machine learning; smart environment

1. Introduction

Through the integration and interconnection of software-centered devices, traditional power system, which are typically centered on mechanical and electrical components, are supported by more complex equipment which enable more advanced functionalities in terms of management and control. The result of such evolution which makes those systems more intelligent is named as Smart Grid (SG) [1,2]. The advantage of having software components in the network enables the introduction of more sophisticated mechanisms for monitoring the SG as well as to support, in a more effective way, the decision process for the dynamic energy distribution according to the current use of energy, resource state and weather conditions [3–6]. Additionally, from the consumer perspective, specific optimization techniques can be exploited for managing the scheduling of the device usage (for example based on specific hours of the day or week according to the electricity costs) in order to save money. Moreover, as an actor can be producer and consumer (so called prosumer) of electricity in a SG [7], such flexibility enables a different way to distribute the energy and to deal with unexpected emergency situations, resulting from faults and failures in the network [8–10]. A general overview of a Smart Grid is depicted in Figure 1, which includes City and Buildings, Power Plants, Wind Turbines, Electric Vehicles, Solar Panels and Smart Homes.

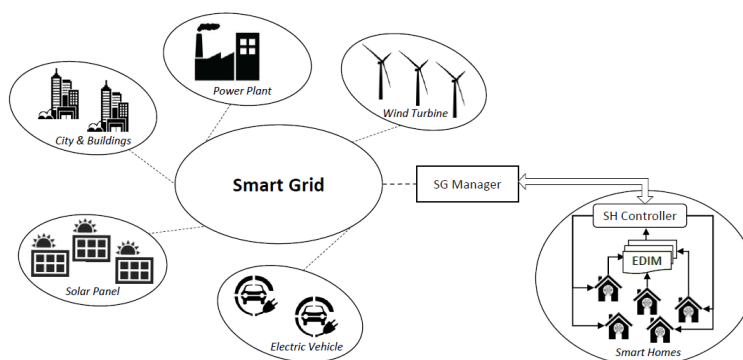


Figure 1. Smart grid overview.

In this scenario, an important role is played by Smart Homes (SHs). They are equipped with electrical devices that can be controlled and monitored remotely not only to achieve economic benefits by saving electricity, but also by contributing in the reduction of carbon dioxide emission in the environment [11]. SHs represent the human side of the SG [12]. They provide a new perspective towards the usage of the energy in the everyday life and, in particular, in the relationship between energy utilities and consumers. Typically, the traditional homes have devices that work locally and manually, usually by switching them on/off by pushing a button, with a limited control in terms of their automatic management. A SH, instead, represents the convergence of energy efficient, controllable electrical appliances and real-time access to energy usage data. This combination of device management and smart grid enables proactively managing energy use in ways that are convenient, cost effective, and good for the environment.

To enable such flexibility, the envisioned communication mechanism foresees two main components: (a) a *Smart Grid Manager (SG Manager)*, which has a global view of the network. It is responsible for making decisions about the energy distribution on the basis of the overall available resources; (b) a *Smart Home Controller (SH Controller)*, which represents an interface between the SG and a house. Every *SH Controller* aims, from one side, to retrieve information regarding the electricity consumption in a house and provides it to the *SG Manager*, and from the other side to manage the electrical devices in the house on the basis of habits and rules specified by the user.

Behind the advantages of a more intelligent energy grid management, one of the main challenges for enabling such a pro-active control relies on the automatic recognition, identification and classification of the electrical appliances. This in turn requires facing several factors [13], such as: (i) *power consumption extraction* that is the process of measuring the energy from different devices in order to identify recurrent consumption patterns; (ii) *multi-mode functionality*, this means that some devices can have multiple operation mode which can be misleading for their identification due to such a complex behavior; (iii) *parallel usage*, this is an important factor that has to be faced, since typically more than one device is in operation at the same time; (iv) *similar characteristics* because many devices can present similarities in the way they use the energy (e.g., consumption, charging time); (v) *external effects* because the data could be spoiled by external and random factors, which are not predictable, such as temperature, communication failures, human influences, etc.

In this context, this work proposes a model for the automatic identification of electrical devices, after they are plugged into an electrical socket. It is the basis for the automatic control of whole functionality of an SH which includes applications regarding, for example, the monitoring and control of appliances, the dynamic load management system based on available resources, power saving by scheduling the devices as well as emergency systems in case of faults or failures of specific resources in the network. The model is based on a set of 19 features which are able to characterize different electrical devices and distinguish them from others. They are derived by analyzing three main aspects: (i) *power consumption*:

related to the electricity being consumed by a device for a certain period of time, (ii) *working schedule*: which includes the hours of the days and the time duration when the device is turned on/off, and (iii) *location*: which represents the place where an electrical device is connected on the basis of the electrical socket within the house. Then, machine learning techniques are used to experiment the model through different classifiers, by using a dataset of 33 types of appliances [14]. The goodness of the proposed model is evaluated in terms of its accuracy, for the identification and classification of electrical devices, with respect to the existing approaches discussed in Section 2.

The rest of the paper is structured as follows. In Section 2, the related works about automatic devices identification are presented, whereas a background on machine learning techniques is reported in Section 3. Then, the adopted research approach is described in Section 4, whereas the proposed model as well as the features are elaborated in Section 5. The experimental results of the classification are highlighted and discussed in Section 6, whereas Section 7 concludes the paper.

2. Related Work

This section discusses the most relevant research efforts and related solutions, which have been proposed for supporting the identification of electrical devices.

In particular, in [15], a middle layer to connect sockets and devices, which is centered on Measurement and Actuation Units (MAUs), is presented. The MAUs monitor and analyze the electrical power consumption of any connected device individually by providing fine grain analysis. The main information for the classification is based on temporal behavior of the appliances, power consumption, shape of the power consumption, and level of noise. Different classifiers have been experimented with, but better performances have been reached by the Random Forest, LogitBoost, Bagging and the Random Committee, which achieved 95.5% accuracy.

In [13], different energy measurements, such as active power, reactive power, phase shift, root mean square voltage and current, by collecting data of each device in an isolated way, are instead considered. This approach aims to provide a plug and play tool to create energy awareness on the basis of real-time energy consumption of electrical devices. Additionally, multi-mode functionality, parallel usage of devices and external effects are also tackled. The difficulties to support the identification of devices which have multi-mode operation compared to those with a single operation mode are discussed. It resulted in an extensive training by deriving a classification model with an accuracy between 94–97%.

In [16], an approach centered on plug-based low-end sensors for measuring the electric consumption at low frequency, typically every 10 s, is presented. In particular, a sensor called PLOGG is used to record a vector of electrical parameters related to the appliance being monitored [17]. However, for each appliance class, a stochastic model is built from the observed consumption profiles of several instances of each class that are used to train the models. A k-NN classification algorithm has been employed on the basis of the identified features by reaching a level of accuracy equal to 85%. However, in [18], a plug and play “smart plug” is investigated. It aims to recognize the consumer appliance category, which is specified according to consumption scales and priorities, based on the employment of specific sensors. It allows for measuring and recording instantaneous energy consumption, by estimating specific parameters of consumer appliances such as the total harmonic distortion and the power factor.

In [19], an approach based on Non Intrusive Appliance Load Monitoring (NIALM) at meter level, to detect whether the device is switched on or off, is discussed. When a change occurs in the overall electrical power signal of the house, the change is analyzed and compared to the already-known patterns available in a database. Another centralized approach, for monitoring power signal, exploits the ZigBee device, which is attached to the main electrical unit [20]. It is used to identify in real time the appliance that contributes to each spike of energy. Another research effort, based on a centralized approach, is described

in [21], in which the authors used a custom data collector and, in particular, a power interface oscilloscope and a computer as hardware. It allows for detecting electrical noise to classify electrical devices in homes by exploiting the electrical noise as an additional parameter. However, time series measurements, which represent electrical signatures of different electrical devices, are used in [22] for their identification.

A summary of the above-mentioned related works is reported in Table 1. Two main approaches, to face with automatic identification of electrical devices, emerged from the above related works. One is based on the employment of additional monitoring devices either distributed [15,22] or centralized [19–21] which results expensive in terms of money for their installation and hardly scalable; the second one that does not exploit any additional devices, is centered on energy measurements [13], but it lacks in the categorization and formalization of the adopted features. Some of those works used aggregated traces (AT) of multiple devices and attempt to disaggregate energy usage, whereas other works, as in our case, used directly disaggregated traces (DT). For the sake of the completeness of this paper, Table 1 provides a high-level overview of prior works by highlighting, for example, the used parameters, data collection techniques in terms of type of traces and accuracy they achieved.

Table 1. Comparison of the related works.

Related Work	Main Parameters	Additional Devices	Accuracy (%)	Adopted Approach	Trace Type
[13]	Active and Reactive Power Phase shift, V_{rms} , I_{rms}	None	94–97	Not specified	Not specified
[15]	Power consumption, Working schedule	Measurement and Actuation Units	95.5	Distributed	DT
[16]	Power consumption at low frequency	Plug-based low-end sensor	85	Distributed	DT
[19]	Power consumption	NIALM device	Not reported	Centralized	AT
[18]	Power factor Harmonic distortion	Smart Plug	Not reported	Distributed	DT
[20]	Active power, Reactive power, Phase shift, Signature length, Root mean square voltage, Sampling frequency	Zigbee Monitor	95	Centralized	AT
[21]	Electrical Noise	Oscilloscope, Laptop Custom Data Collector	85–90	Centralized	AT
[22]	Active power, Reactive power, Root mean square voltage, Phase shift	Smart Plug	93.6	Distributed	DT

Other proposals are available in literature, which are not reported and compared in Table 1 because some of them are based on different approaches and/or input data whereas others have different purposes. For example, there have been other works in the context of appliance identification that are centered on both different approaches and input data, such as those based on high frequency conducted electromagnetic interference (EMI) which use Non-Intrusive Load Monitoring (NILM), as described in [23]. It aimed to present some of the key challenges towards exploiting EMI and the dataset of the collected data, which was used in the experiment, is also available online and freely downloadable [24]. However, the work in [25] proposes a technique that aims at identifying anomalous appliances in buildings by using aggregate smart meter data and contextual information in near real time.

In this wide context, our paper is strictly related to those works, reported in Table 1, which dealt with disaggregated traces. In particular, our work stands out from the previous ones because (i) a set of features that characterize electrical devices are proposed and formalized (ii) a model, based on their combination, is used to identify and recognize devices when they are plugged into the circuit without additional monitoring devices and based on disaggregated traces, and (iii) high performances in terms of accuracy are reached.

3. Background on Machine Learning

Machine learning (ML) is a data analysis technique, based on computational algorithms able to learn directly from the data without a predefined model [26]. In particular, ML techniques aim to identify patterns from which the extracted information is used to make better forecasts, prediction and decisions.

Thanks to the huge amount of available data, ML represents a popular approach that is exploited in several fields, for facing classification-related problems, such as in (i) computational finance for the evaluation of credit risk and algorithmic trading; (ii) image processing and artificial vision for facial recognition, motion detection and object identification; (iii) computational biology for the diagnosis of tumors, pharmaceutical research and DNA sequencing; (iv) energy production for price and load forecasts; (v) automotive, aerospace and manufacturing sectors, for predictive maintenance; and (vi) natural language processing, for speech recognition applications.

Among the variety of existing techniques, an overview of the most popular ones is provided and briefly discussed below, such as Decision Trees (DTs), Support Vector Machine (SVMs), Linear Regressions (LRs), Naive Bayes (NB), Random Forest (RF), Random Committee (RC), Boosting, Bagging, and Artificial Neural Networks (ANNs). Some of them are then selected and taken into account in the evaluation part of the proposal.

3.1. Linear Regression and Decision Trees

The goal of linear regression models is to find a linear mapping between observed features and observed real outputs so that, when a new instance is seen, the output can be predicted [27]. Regression is a method of modeling a target value based on independent predictors. This method is mostly used for forecasting and finding out cause and effect relationships between variables. DTs are decision tools based on a tree-model [28,29]. They are navigated from the root to the leaves, each intermediate node represents a decision point and the ramification represents the properties that leads to a particular decision. The predicate that is associated with each internal node, which is used to discriminate among the data, is called “split condition”. When a leaf is reached by navigating the tree, not only is a particular classification associated with the input instance, but, thanks to the path, it is possible to understand the reason for a particular result. A DT should be used when the relations among the various aspects of a specific application context are difficult to explain. In this case, the nonlinear approach of the DT performs better than the Linear Regression.

3.2. Support Vector Machines and Naive Bayes

SVMs are linear models for classification and regression problems which are used to solve linear and nonlinear, problems [30,31]. The idea of SVM is based on the definition of a line or a hyperplane which separates the data into classes. Based on given labeled input, the algorithm outputs a hyperplane-based model that is able to classify new instances. Given a set of training examples (training set), each of which is labeled with the class to which the two classes belong, an SVM training algorithm constructs a model that assigns new examples to one of the two classes, thus obtaining a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped in such a way that the

examples belonging to the two different categories are clearly separated by the widest possible space. The new examples are then mapped in the same space and the prediction of the category to which they belong is made on the basis of the side in which it falls. In addition to linear classification, it is possible to use SVM to effectively perform nonlinear classification using the kernel method, implicitly mapping their inputs into a multi-dimensional feature space. NBs belong to a probabilistic family of classifiers [32]. They are centered on the theorem of Bayes, which is based on the assumptions of independence among features. NBs are highly scalable, requiring a number of parameters linear in the number of variables. Furthermore, the method is very efficient for text categorization, which can compete with more advanced methods including SVMs, with appropriate pre-processing.

3.3. Random Forest and Random Committee

The Random Forest is a very popular algorithm for feature ranking [33]. It belongs to the Bagging methods (Bootstrap Aggregating) and it is based on the use of multiple decision trees DT_k , each of which is trained on a subset S_k of the training set. Each new instance provided in input is classified by all the k -Decision Trees, each of which provides its own classification. A voting mechanism, which can consist of majority vote rule or on the average value gathered from all the k -classifications, is then adopted to establish the final classification based on the most common class in the node. However, the Random Committee builds an ensemble of base classifiers and generates their prediction by averaging of the estimated probability [34]. Each base classifier is based on the same data, but it uses a different random number seed, which makes sense if the base classifier is randomized; otherwise, all classifiers would work in exactly the same way.

3.4. Boosting and Bagging

Bagging and Boosting are also ensemble methods [35,36]. The idea of Boosting is to combine “weak” classifiers in order to create a classifier with a better accuracy. In algorithms such as Adaboost, the output of the meta-classifier is given by the weighted sum of the predictions of the individual models. Whenever a model is trained, there will be a phase of repeating the instances. The boosting algorithm tends to give greater weight to the misclassified instances, with the aim of obtaining an improved model on the basis of these latter instances. On the contrary, the Bagging approach aims to reduce variance from models that might have a very high level of accuracy, but typically only with the data, on which they have been trained, which is called over-fitting. It tries to reduce this phenomenon by creating its own variance among the data by sampling and replacing data by testing diverse models called hypothesis.

3.5. Artificial Neural Networks

ANNs are particular computational models, which are able to represent knowledge based on massive parallel processing and pattern recognition based on past experience or examples [37]. An ANN is defined through an initial layer on the basis of the available inputs, a final layer which represents the output of the computation and a hidden layer which is defined in terms of potential multi-layers through which the inputs undergo various transformation and calculation steps as long as the final layer is reached and the output is generated. They are computation models inspired by biological networks in which: (i) the information processing occurs at several simple elements that are called neurons; (ii) signals are passed between neurons over connection links; (iii) each connection link has an associated weight, which, in a typical neural net, multiplies the signal transmitted; (iv) each neuron applies an activation function (usually nonlinear) to its net input (sum of weighted input signals) to determine its output signal. By such replicated learning process and associative memory, an ANN model can classify information as pre-specified patterns. A typical ANN consists of a number of simple processing elements called neurons,

nodes or units. Each neuron is connected to other neurons by means of directed communication links. Each connection has an associated weight, which represents the parameters of the model being used by the net to solve a problem. ANNs are usually modeled into one input layer, one or several hidden layers, and one output layer.

4. Research Approach Description

This section aims to clarify the approach adopted in this research task, which is depicted as a process in Figure 2. The process is designed in three main parts, which are organized in lanes: “Data Management”, “Phase” and “Work-product”. More specifically, the “Data Management” lane is related to the dataset elaboration, the “Phase” lane provides the information about what is done and in which order, whereas the “Work-product” lane illustrates which output is generated and how it is eventually used. By describing the process by phase from the top to the bottom, its functioning as well as the sequence order of its actions are highlighted. In particular:

- *Model definition*: this phase starts by taking in as input an *Initial Dataset* which contains a collection of data related to different types of devices. This phase aims at identifying common features among the different type of devices that will be used to characterize and discriminate them. The output of this phase is represented by a model based on different features called *Feature-based model*.
- *Feature-driven value extraction*: this phase uses both the *Initial Dataset* and the *Feature-based model*. In particular, the *Feature-based model* is applied to the *Initial Dataset* and, in particular, on the recorded traces in order to extract additional information, called *Derived Dataset*. Such information, which enriches the traces available in the *Initial Dataset*, is then exploited to distinguish the different appliances.
- *Data splitting*: this phase is centered on the *Derived Dataset* and aims to divide it into two disjoint subsets: *Training Set* and *Test Set*. The *Training Set* is provided in input to a learning algorithm in order to train appropriately the classifier which is built on the basis of the identified features, whereas the *Test Set* is used to validate it.
- *Learning*: in this phase, one or more learning algorithms are chosen on the basis of both the analysis to be conducted (e.g., supervised, semi-supervised or non-supervised) and the kind of available data (e.g., labeled or not labeled). This phase ends by training such learning algorithm by using the *Training Set* in order to obtain a *Trained classifier* from each of them.
- *Model validation*: in this phase, the capacity of the trained classifiers obtained in the previous phase in the identification and classification of electrical devices is assessed. To this aim, only the *Test Set*, which consists of traces of devices that have been not used to train the classifiers, are employed so as to get the *Classification Results*.
- *Feature analysis*: this phase ends the overall process. In particular, starting from the *Classification Results* gathered from the previous phase, the features that play the most important role, in the electrical devices identification, are discussed.

In the next sections, more in-depth details on the conducted research activity are given, by focusing on the work-products. Specifically, Section 5 provides the full description of the *Feature-based model* by describing a particular instance of it called EDIM (Electrical Devices Identification Model) by highlighting which aspects have been considered in its definition and why, whereas the *Trained classifiers*, the *Classification Results* and the *Importance Feature Results Analysis* are discussed in Section 6.

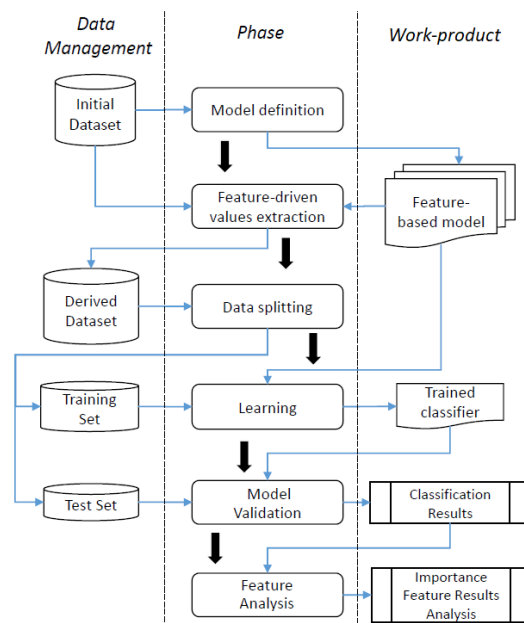


Figure 2. Research approach: data management, phases and work-products.

5. EDIM: An Electrical Devices Identification Model and Related Features

In this section, the proposed *Electrical Devices Identification Model (EDIM)* for the identification of electrical devices is described. Specific aspects, related to the usage of appliances in terms of time, place and electricity consumption have been considered for its definition. Their combination aims to highlight emergent behaviors that are able to characterize and discriminate among different devices.

In particular, as it is depicted in Figure 3, the *EDIM* model is inspired by three main driving questions: (i) HOW MUCH does a device consume? (ii) WHEN does a device consume? (iii) WHERE is a device used? The rationale behind them relies not only on extracting and using information that is directly derivable or measurable from a device, such as its energy consumption, but also to combining it with further information related to the way of using a particular device—for example, by considering differences of the use of a device in specific daily time slots, weekly or seasonal, relationships with other devices such as their use in sequence or in parallel, as well as by distinguishing whether a device is used in a specific area or if it is used, with a certain frequency, in different places.

By dealing with the above-mentioned questions, three main related *feature classes* have been identified, namely *Energy and Power Consumption*, *Temporal Usage* and *Appliance Location*. For each of them, a set of specific features have been proposed, for a total of 19 basic features. In the next subsections, the description of the defined *feature classes* is provided and, for each of them, the features that have been proposed are presented and formalized through a mathematical notation.

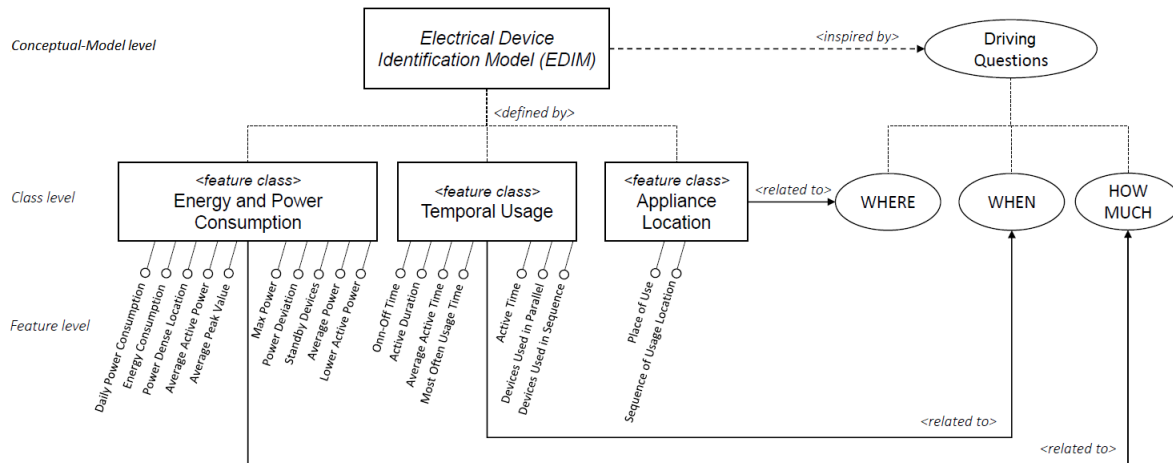


Figure 3. The Electrical Device Identification Model and related questions.

5.1. Energy and Power Consumption Class

This feature class focuses basically on the measurement of power and energy at various levels and at specific points of time. The features belonging to this class aim to extract information related to the electricity consumption of a device in order to characterize it, by answering the question “HOW MUCH does a device consume?” In this class, the following consumption-related features are identified: *Daily Power Consumption*, *Max Power*, *Power Deviation*, *Average Power*, *Average Active Power*, *Lower Activity Power (or MinPower)*, *Energy Consumption*, *Average Peak Value*, *Power Dense Location*, and *Standby Devices*. In the next section, the above-mentioned features, which have been proposed in this class, are elaborated.

Daily Power Consumption. It is used to compute the amount of power p consumed by a device j on a day D , by observing it in a time unit $i[s] \in D$:

$$P_j^{Tot} = \sum_{i \in D} p_j(i), \text{ where } i \text{ corresponds to a second [s].} \quad (1)$$

Max Power. This feature is used to calculate the maximum power value p used by a device j within a specific day D :

$$P_j^{Max} = \max\{p_j(i)\}, \text{ where } i[s] \in D. \quad (2)$$

Power Deviation. This feature deals with the power deviation which is computed as the sum of the difference between the *Max Power* of a device j within a reference day D and its power consumption at every time unit $i[s] \in D$. Only when j is in operation $Status_j(i) > \lambda = 5[W] = On$:

$$P_j^{Dev} = \sum_{i \in D} (P_j^{Max} - p_j(i)) \Leftrightarrow Status_j(i) = On. \quad (3)$$

Average Power. Given a device j , this feature calculates the average power used from it, related to a day D by considering both active and non-active operation time $i \in D$:

$$P_j^{Avg} = \frac{\sum p_j(i)}{\text{count}(i)}, \forall i \in D. \quad (4)$$

Average Active Power. Given a device j , this feature calculates the average power used from it, related to a day D by considering only the active operation time $i \in D$, such that $Status_j(i) > \lambda = 5[W] = On$:

$$P_j^{AvgAct} = \frac{\sum p_j(i)}{count(i)}, \forall i \in D \text{ s.t. } Status_j(i) = On. \quad (5)$$

Lower Activity Power (Min Power). This feature is used to calculate the minimum power value p used by a device j within a specific day D , by considering only the active operation time $i \in D$, such that $Status_j(i) > \lambda = 5[W] = On$:

$$P_j^{Min} = \min\{p_j(i)\}, \text{ where } i[s] \in D \text{ s.t. } Status_j(i) = On. \quad (6)$$

Energy Consumption. Given a time period D (e.g., a Day, a Week, a Month) divided into a set of n sub-periods $\{d_1, d_2, \dots, d_n\} \subset D$. This feature is used to calculate the energy consumption of a device j in a specific sub-period $b \in D$:

$$EC_j^b = \sum_i^D p_j(i) \Leftrightarrow i \in b \subset D. \quad (7)$$

Average Peak Value. Given a reference period of time D_j (e.g., a Day, a Week, a Month) as a disjoint list of $K = \{1, 2, \dots, k\}$ time intervals $I_j = \{sp_j(h_0, h_1), sp_j(h_2, h_3), \dots, sp_j(h_{k-1}, h_k)\}$ in which a device j was actively used, then the Average Peak Value of a device j APV_j calculates the average of all the peak values within the considered period of time D_j , where $peak(sp_j(h', h'')) = \max\{p_j(i)\}$ with $h' < i < h''$ is the max value of energy consumed from the the device j in the time interval $[h', h'']$:

$$APV_j(D_j) = \frac{\sum peak(sp_j(h', h''))}{k}, \forall sp_j(h', h'') \in I_j. \quad (8)$$

Power Dense Location. Given a location l and a set of n devices $J = \{j_1, j_2, \dots, j_n\}$. This feature provides the amount of power consumed in l from all the devices in J in an arbitrary period of time D , if the total power consumed is more than a reference threshold $PD_{threshold}$:

$$PD_D^l = \sum_j^J \sum_{i=0}^T p_j^l(i) > PD_{threshold}. \quad (9)$$

Standby Devices. Given a set of n devices $J = \{j_1, j_2, \dots, j_n\}$, this feature calculates a subset of devices $SB_{Dev} = \{j_1, j_2, \dots, j_k\} \subset J$ which are neither Off nor On, rather those which present a standby mode that is a power consumption $0 < p_j(i) < \lambda = 5[W]$ for at least an uninterrupted period of time δt :

$$SB_{Dev} = \langle j_1, j_2, \dots, j_k \rangle, \text{ if } \exists p_j(i) \text{ s.t. } 0 < p_j(i) < \lambda, \text{ and } continuous(i) > \delta t. \quad (10)$$

5.2. Temporal Usage Class

This feature class focuses on the use of a device, mainly from a temporal point of view, by considering the question “WHEN does a device consume?” The features that fall into this class try to extract information, regardless of the amount of energy consumed, with the aim of identifying temporal usage patterns such as daily, weekly, seasonal related to a single device as well as sequence-parallel relationships between multiple devices (e.g., the dryer after the washing machine, or the decoder along with the television). In this class, the following time-related features are identified: *On-Off Time*, *Active Time*,

Average Active Time, Active Duration, Most often Usage Time, Devices Used in Sequence, Devices Used in Parallel. The above-mentioned features, which have been proposed in this class, are described below in more detail.

On-Off Time. This feature is used to know, in which instant of time i of a day D , a device j is turned On/Off. The function $Status_j(i)$ is used to check the working status of j at the time i [s] $\in D$, in order to identify when a change occurs, based on the previous instant of time $i - 1$, where $\lambda=5[W]$ is the On-threshold:

$$T_j^{On-Off}(i) = \begin{cases} Off, & \text{when } Status_j(i-1) > \lambda \text{ and } Status_j(i) = 0, \\ On, & \text{when } Status_j(i-1) = 0 \text{ and } Status_j(i) > \lambda. \end{cases} \quad (11)$$

Active Time. This feature counts the number of times that a device j is turned on in an arbitrary day D . For example, a dishwasher is typically turned on once or twice a day:

$$T_j^{Act}(D) = count(T_j^{On-Off}(i)) \Leftrightarrow T_j^{On-Off}(i) = On, \forall i \in D. \quad (12)$$

Active Duration. Given a device j , its active duration in an arbitrary day D represents the overall time i in which j is active that is $Status_j(i) = On$:

$$T_j^{ActiveDur}(D) = count_j(i) \Leftrightarrow Status_j(i) > \lambda = 5[W] = On, \text{ with } i \in D. \quad (13)$$

Average Active Time. This feature calculates the active average duration of a device j within an arbitrary day D :

$$T_j^{AvgAct}(D) = \frac{T_j^{ActiveDur}(D)}{T_j^{Act}(D)}. \quad (14)$$

Most Often Usage Time. Given a reference period of time D_j (e.g., a Day, a Week, a Month) as a disjoint list of $K = \{1, 2, \dots, k\}$ time intervals $I_j = \{sp_j(h_0, h_1), sp_j(h_2, h_3), \dots, sp_j(h_{k-1}, h_k)\}$ in which a device j was actively used, then the most often usage time of a device j indicates the longest interval of time $\Delta h_j = \langle h', h'' \rangle_j$ such that $sp_j(h', h'') \in I_j$ and $h', h'' \in K$, in which the device j was used:

$$\Delta h_j = \max \langle h', h'' \rangle_j = \max \{(h'' - h')_j\} = \max \{sp_j(h', h'')\}. \quad (15)$$

Devices Used in Sequence. Given two instants of time i_1 and i_2 with $i_2 \geq i_1$. A device j_2 works in sequence after j_1 $seq((j_2, j_1))$, when j_1 stops working at time i_1 , which is $Status_{j_1}(i_1 - 1) = On$ and $Status_{j_1}(i_1) = Off$ and the device j_2 starts working in a subsequent instant of time i_2 , which is $Status_{j_2}(i_2) = Off$ and $Status_{j_2}(i_2 + 1) = On$. Thus, this feature $S\bar{E}Q_J(D)$ returns all the possible couples of devices $j_1, j_2 \in J$, which work in sequence in a reference period D :

$$S\bar{E}Q_J(D) = \{seq(j_1, j_2)\}_D \forall j_1, j_2 \in J \text{ set of devices.} \quad (16)$$

Devices Used in Parallel. Given two instants of time i_1 and i_2 with $i_1 > i_2$, $i_1 > (i_2 + 1) - \Delta t$ and $\Delta t \geq 1$. The device j_2 works in parallel with j_1 $par((j_2, j_1))$, when j_1 stops working at time i_1 that is $Status_{j_1}(i_1 - 1) = On$ and $Status_{j_1}(i_1) = Off$ and the device j_2 starts working in an instant of time i_2 that is $Status_{j_2}(i_2) = Off$ and $Status_{j_2}(i_2 + 1) = On$. This feature $P\bar{A}R_J(D)$ returns all the possible couples of devices $j_1, j_2 \in J$, which work in parallel at least for a threshold Δt in a reference period D :

$$P\bar{A}R_J(D) = \{par(j_1, j_2)\}_D \forall j_1, j_2 \in J \text{ set of devices.} \quad (17)$$

5.3. Appliance Location Class

The place of use of a device is another important indicator, since some electrical devices are often used in the same place (e.g., the hairdryer in the bathroom, the kettle in the kitchen). Some of them are movable and others are not. As a consequence, some devices can be used in more than one location inside a house and they can be active in not more than one location at a time. This feature class is driven by the question “WHERE is a device used?” Considering this aspect, in this class, the following location-related features are identified: *Place of Use*, *Sequence of Usage Location*.

Sequence of Usage Location. Given a set of locations $L = \{l_1, \dots, l_z, \dots, l_k\}$ that represent specific places (e.g., a kitchen, a bathroom, a bedroom, a living room and so on) in a given house h . The feature computes the list of locations in a house h , where a device j was chronologically used $p_j^{l_z}(i_w) > \lambda = 5[W] = On$ in an arbitrary day D :

$$SoUL_j^h(D) = \langle l_1(i_1), \dots, l_z(i_w), l_{z+1}(i_{w+1}), \dots, l_k(i_w) \rangle_j^h, \text{ with } i_1 \dots i_w \in D, \quad (18)$$

$$\forall \langle i_w, i_{w+1} \rangle \text{ with } i_w < i_{w+1} \text{ and } p_j^{l_z}(i_w) > \lambda \text{ and } p_j^{l_{z+1}}(i_{w+1}) > \lambda.$$

Place of Use Given a set of locations $L = \{l_1, l_2, \dots, l_k\}$ that represent specific places (e.g., a kitchen, a bathroom, a bedroom, a living room) in a given house h . This feature allows for knowing in which location $z \in L$ the device j was used $p_j^z(i) > \lambda = 5[W] = On$. That is, it was On for at least one time unit i in an arbitrary day D :

$$z_j^h(i) \Leftarrow p_j^z(i) > \lambda, \text{ with } z \in L \text{ and } i \in D. \quad (19)$$

6. Experiments and Results Discussion

In this section, first an overview of the used dataset is given, then the results gathered by experimenting the proposed model are described and, finally, a discussion on the importance of the features is provided.

6.1. Dataset Overview

The dataset used to evaluate the proposal consists of a collection of traces related to the daily use of different electrical devices. This dataset, which is made available under the Open Database License (ODbL) [38], is public available and freely downloadable [14]. Each entry of the dataset contains basic data such as the identifier of a device, the time unit with a granularity of a second, which is used to collect the data of each device, the amount of energy consumed in a time unit and so on.

The trace-base data have been grouped into three categories in accordance with the process they have been collected: “Full-day traces”: which contains complete traces that have been recorded for a time period over 24 h; “Incomplete traces”: which contains traces with missing measurements in different instants of time over the day; “Synthetic traces”: which contains trace fragments of devices that have been manually completed with values corresponding to zero consumption readings, when the real values were not available. In our case, the folder containing the “Full-day traces” has been used in the experimentation, which corresponds to 33 types of devices, as listed in Table 3.

6.2. Features Evaluation

A machine learning based approach has been used for the evaluation of the proposed identification model. Starting from such row data available in the “Full-day traces” folder, additional information has been extracted by using the features that have been proposed in Section 5. Both the basic and extracted information is used to train and test different classifiers. As the dataset is labeled, the case in consideration

falls under a supervised learning problem. As a consequence, only supervised machine learning techniques have been considered and compared. In particular, Random Forest, Bagging, LogitBoost, Decision Tree, Naive Bayes and SVM algorithms have been selected and experimented with for two main reasons: (i) on one hand, to the best of our knowledge, they have shown the best performance in literature, (ii) on the other hand, it is possible to analyze and understand the logic behind their classification process, which is, instead, not always possible with other techniques. For example, neural networks make difficult to understand what happens during the classification, since they act as a black box, to understand which features play the most important role for the device identification. As a consequence, they have not been considered. For the sake of completeness and clarity of this work, Figure 4 summarizes the machine learning algorithms that have been used in the experiments along with the values of the parameters that have been used after having tuned them.

Bagging	
Parameter	Value
n_estimators	10
max_samples	1
max_features	1
bootstrap	WAHR
bootstrap_features	FALSCH
oob_score	FALSCH
warm_start	FALSCH
n_jobs	none
random_state	none
verbose	0

Decision Tree	
Parameter	Value
criterion	gini
splitter	best
max_depth	none
min_samples_split	2
min_samples_leaf	1
min_weight_fraction_leaf	0
max_features/random_state	none
max_leaf_nodes	none
min_impurity_decrease	0
min_impurity_split	1.00E-07
class_weight	none
persort	FALSCH

Random Forest	
Parameter	Value
n_estimators	100
criterion	gini
max_depth	None
min_samples_split	2
min_samples_leaf	1
min_weight_fraction_leaf	0
max_features	Auto
max_leaf_nodes	None
min_impurity_decrease	0
min_impurity_split	1.00E-07
bootstrap	WAHR
oob_score	FALSCH
n_jobs	None
random_state	None
verbose	0
warm_start	FALSCH
class_weight	None

Logit Boost	
Parameter	Value
n_estimators	100
weight_trim_quantile	0.05
max_response	4
learning_rate	1
bootstrap	FALSCH
random_state	none

Naïve Bayes	
Parameter	Value
Priors	None
v_smoothing	1.00E-09

Support Vector Machine	
Parameter	Value
gamma	scale
kernel	rbf
shrinking	true

Figure 4. Machine learning algorithms and related parameters.

The training and test set have been constructed by using a standard approach based on 5-fold cross validation, so as to reduce both the risk of losing important patterns/trends in data set and the error induced by bias. In more detail, 80% of all 33 categories of available devices’ traces have been exploited to build the training set extracted from the traces, with the remaining 20% of the data to build the test set, which are employed respectively to train and test the above-mentioned classifiers. Moreover, the testing of the model was done for each single device in order to get the prediction of each device separately (i.e., at device level), and then the accuracy of all the devices were averaged to calculate the overall accuracy of each trained model. Moreover, as different traces of different models of certain device categories were

available (for example LCD-TV, Router, Washing Machine), the construction of the training set and the test set took into account that the traces belonging to a certain device model were used only in the training set or in the test set. This allowed for showing that the proposed features are able to recognize even new devices with similar behavior, in terms of energy usage, belonging to one of the 33 device categories under consideration.

A first result is shown in Table 2 in terms of accuracy of the different classifiers. A general observation is that all the classifiers present an accuracy higher than 90%, which provides an indication of the goodness of the selected features. Indeed, they show a certain degree of independence from a specific classifier and, as a consequence, they are well suited to describe appliance types and distinguish them from others. Moreover, among them the best performance is reached by the Random Forest algorithm with 96.51% accuracy, which is why the rest of the analysis is specifically based on its use for the subsequent evaluation. The details are reported in Table 3 by showing the result values for *true* and *false positives* as well as the *precision*, which measures the proportion of actual negatives that are correctly identified as such, and the *recall*, which measures the proportion of actual positives that are correctly identified as such, for each kind of device.

Table 2. Accuracy of the different classifiers.

#	Algorithm	Accuracy [%]
1	Random Forest	96.51
2	LogitBoost	94.99
3	Bagging	93.02
4	Decision Tree	91.10
5	Naive Bayes	90.26
6	Support Vector Machine	90.11

As we can see, our implementation reaches at least 80% accuracy and almost all the devices are always classified correctly, which can be seen from the true positive ratio, which is equal to 1.0. However, for other devices, the false positive ratio is, however, very low. Additionally, both the results obtained by calculating the precision and the recall values comply with the observed values related to the true positive rate. Only for one device, and in particular for the Water Kettle, a lower level of classification is shown. This is associated with the limited number of instances available during the training phase of the classifier that made the training phase of the classifier very difficult for this typology of device.

In general, some devices present electrical characteristics that are easier to recognize and which require few instances for training the classifier; others instead require a greater number of instances. This can be traced back to the fact that the behavior of some devices is not only strongly dependent on their mode and state of operation, such as for an alarm clock or a vacuum cleaner. Others, instead, have dependencies on their state and external factors, for example in the case of the Water Kettle, the amount of water to be heated could influence the duration of its heating process. Consequently, not only is it necessary to have a sufficient amount of traces, but they should also be collected considering these additional factors.

However, more than 60% of the devices are correctly identified and classified, as we can see from the true positive rate, precision and recall, without errors. In summary, we globally obtained very high performances in terms of accuracy compared with other related works, and in particular with respect to the related work [15], by only using 19 features instead of 517 features on the same dataset, by requiring both less computational resources and computing time.

Table 3. Accuracy classification for each appliance provided by the Random Forest.

Electrical Device	True Positive	False Positive	Precision	Recall
Alarm Clock	1.0	0	1.0	1.0
Amplifier	1.0	0	1.0	1.0
Bean to cup	1.0	0	1.0	1.0
Coffee machine	1.0	0	1.0	1.0
Dishwasher	1.0	0	1.0	1.0
Desktop PC	1.0	0	1.0	1.0
Dryer	1.0	0	1.0	1.0
DVD	0.99	0.001	0.941	0.99
Ethernet	0.95	0	1.0	0.95
Freezer	1.0	0	1.0	1.0
Iron	0.80	0.002	0.65	0.80
Lamp	0.88	0.002	0.85	0.88
Laptop	0.96	0	1.0	0.96
Mediacentre	0.99	0	1.0	0.99
Microwave	1.0	0.001	0.95	1.0
Monitor-CRT	0.92	0.002	0.93	0.92
Monitor-TFT	1.0	0	1.0	1.0
PlayStation	0.87	0	1.0	0.87
Printer	1.0	0.001	0.98	1.0
Projector	0.97	0	1.0	0.97
Refrigrator	1.0	0	1.0	1.0
Router	1.0	0	1.0	1.0
Stove	1.0	0	1.0	1.0
Toaster	1.0	0.001	0.95	1.0
TV-CRT	1.0	0	1.0	1.0
TV-LCD	1.0	0	1.0	1.0
TV-REC	0.96	0	1.0	0.96
USB Harddrive	1.0	0	1.0	1.0
Vacuum cleaner	1.0	0	1.0	1.0
Water Fountain	1.0	0	1.0	1.0
Water Kettle	0.57	0.003	0.58	0.57
Wash Machine	1.0	0.002	0.983	1.0
Xmas Lights	0.99	0	1	0.99
Weighted average	0.9651	0.0004	0.964	0.9651

6.3. Discussion of the Features Importance

In addition, not only is it important to have a model that performs well, but it is also very important to understand why it works good (or bad) and under which conditions. This helps to understand the logic of the model and the reasoning behind a decision. Knowing the importance of a feature in the classification process may motivate the exploitation of more complex one or removing them based on their significance, even by sacrificing some accuracy for the sake of the interpretability. In our case, an analysis on the feature importance has been conducted, in order to know which of them plays the most important role in the electrical device identification process.

In the assessment of the feature importance, a common evaluation criterion is called impurity, which is used to express the level of homogeneity (or heterogeneity) among a group of items [39]. In our case, the classification model is based on the Random Forest, which in turn consists of a set of sub-trees. In this case, the impurity value for each feature is calculated on the basis of each sub-tree. As a consequence, the impurity is assessed over all the nodes for all the trees. Every node in the decision tree splits the dataset into two subsets, so that all the results showing similarities fall in the same subset. As a consequence, from one side, the more important a feature is, the more it decreases the impurity in the tree, but, on the

other side a mistake based on it will produce a greater impact on the overall classification. Typically, the features which generate greatest decrease in terms of impurity are located closer to the root-level of a tree, whereas those that produce less decrease of impurity are closer to the leaf-level.

In this case, Table 4 shows how much each feature contributes in the reduction of the level of impurity and, as a consequence, their importance in percentage in the identification of the electrical devices. As it is visible, the features are sorted by level of importance. The most important feature is shown in the first row of the table. Moving down, the less important features are listed. In particular, the *Average Peak Power* is the most important one because it provides the highest level of impurity reduction, unlike with the *Energy Consumption*, which is the feature that contributes the least to discriminating between appliances and, consequently, to reduce the level of impurity among them. It is important to note that no value of importance is reported regarding to the *On-Off Time*. This is because it does not contribute directly as a discriminating characteristic, but it is used indirectly from other more complex features in the classification process, as a supporting feature.

Table 4. Descending order of features, for level of importance, expressed as a percentage.

#	Feature Name	Importance (I) [%]
1	Average Peak Power	14.8142
2	Average Active Power	14.6424
3	Max Power	11.3404
4	Average Active Time	10.1315
5	Lowest Active Power	10.1219
6	Place of Use	8.366
7	Active Duration	7.6177
8	Devices Used in Parallel	5.2789
9	Average Power	4.9267
10	Most of Usage Time	4.5993
11	Standby Devices	2.0524
12	Active Time	1.9884
13	Power Deviation	1.7767
14	Devices Used In Sequence	1.2207
15	Power Dense Location	0.6378
16	Sequence of Usage Location	0.2529
17	Daily Power Consumption	0.2319
18	Energy Consumption	0.0002
19	On-Off Time	-
Tot.	-	100

A further observation can be made by calculating the average percentage value of importance of a feature $\bar{I}_F = 5.26\%$ as:

$$\bar{I}_F = \frac{100}{\text{count}(F)} = \frac{100}{19} = 5.26\%, \text{ with } F \text{ the set of all features.} \quad (20)$$

On the basis of such reference parameter \bar{I}_F , only the first eight features of Table 4 present a higher value than \bar{I}_F . It means that *Average Peak Power*, *Average Active Power*, *Max Power*, *Average Active Time*, *Lowest Active Power*, *Place of Use*, *Active Duration*, *Devices Used in Parallel* can be considered the most important features in the classification process. Indeed, by summing their percentage values of importance, the result is equal to 82.31%, which reflects the level of accuracy of at least 80%, in the classification of almost all the devices with a true positive ratio equal to 1.0, as described above. However, the diagram depicted in Figure 5 reports for such features both the absolute importance values with respect to all 19 features as well as the relative values when only the most important ones are considered.

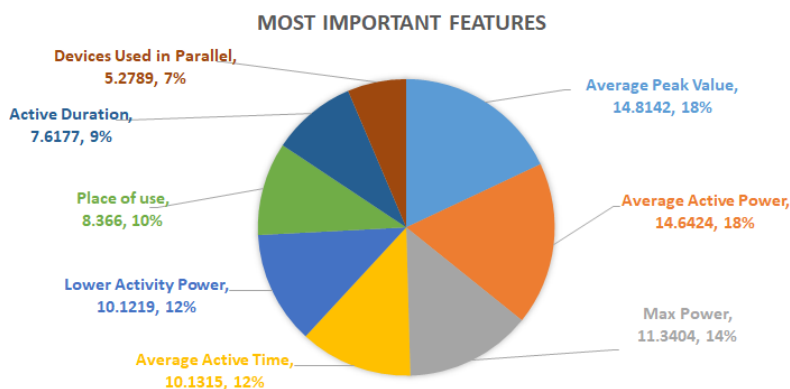


Figure 5. The most important features along with their absolute and relative importance values.

7. Conclusions

The paper focused on the automatic identification of electrical devices based on features. An *Electrical Device Identification Model* centered on three feature classes related to energy consumption, time usage and location, have been proposed. For each class, specific features have been defined and formalized for a total of 19 distinct features. The information extracted by applying such features has been used to train six different classifiers (i.e., Random Forest, Bagging, LogitBoost, Decision Tree, Naive Bayes and SVM), which have shown a high level of accuracy as a symptom of goodness of the proposed features. Of course, a number of variations of techniques, which are designed for different types of tasks, are also available. From one side, they typically allow for obtaining even better results; this means that, to some extent, they influence the assessment. Since our aim was to evaluate the proposed features by avoiding any potential kind of bias, in this research activity, we considered only standard techniques in order to obtain a more neutral assessment of the features and to compare the results with the prior works reported in Section 2. Further experiments and details related to the Random Forest classifier that provided the highest accuracy equal to 96.51% have been conducted and discussed, as it outperformed with respect to the related work [15] by using only 19 features on the same dataset. In particular, the ratios of true and false positives, as well as the precision and the recall with reference to the specific appliances, have been evaluated. An additional analysis has been done, in order to understand the logic of the classifier and the reasoning behind its decisions. Specifically, not only did it emerged how much each feature contributes in the classification process, but also the most important ones have been identified.

Ongoing works aim to extend this identification model in order to (i) enhance the interaction between Smart Homes and Smart Grids to improve the decision-making process for the energy automatic management and distribution in the network, and (ii) improve the local management of electrical devices in smart homes automatically on the basis of users' habits, and centered on the definition of specific user' profiles. Furthermore, it is worth noting that the behavior of some devices might change over time due to aging, temperature and environmental effects. For example, with aging, the battery of a smart phone might show performance degradation in terms of needed energy while charging or it might take more time to charge or might discharge at a faster rate. Similarly, it could happen with other devices like an air conditioner whose usage changes from season to season, also depending on the external temperature. Such behaviors cannot be covered in a limited time trace dataset, which is why there could be some points where the classification fails. As the devices' behavior or their use can change over the time, a possible future work regards the extension of our model with a patching-based approach for classifiers [40], which focuses on the adaption of existing classification models to new data. As classification often faces scenarios where an already existing model needs to be adapted to a changing environment. Such research

work identifies the regions of instance space in which adaptation is needed and, after that, local classifiers for these regions are trained. Such regions after training are incorporated into the model and can handle the predictions even after some changes due to aging or environment. Thus, whenever a decay is detected in the performance of the model, the adaptation is triggered with the goal of finding patches to the classifier that can act efficiently without training the model again from scratch.

Author Contributions: The authors contributed equally in all parts of the article in terms of literature review, adopted methodology, feature identification, model definition, experimentation and results analysis.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SG	Smart Grid
SGs	Smart Grids
SH	Smart Home
SHs	Smart Homes
EDIM	Electrical Devices Identification Model
MAUs	Measurement and Actuation Units
ODbL	Open Database License
ML	Machine Learning
DT	Decision Tree
LR	Linear Regression
SVM	Support Vector Machine
NB	Naive Bayes
RF	Random Forest
RC	Random Committee
ANN	Artificial Neural Network
NIALM	Non Intrusive Appliance Load Monitoring

References

1. Karnouskos, S. Cyber-Physical Systems in the SmartGrid. In Proceedings of the 2011 9th IEEE International Conference on Industrial Informatics, Lisbon, Portugal, 26–29 July 2011; pp. 20–23.
2. IEEE. *IEEE Vision for Smart Grid Control: 2030 and Beyond Reference Model*; IEEE: Piscataway, NJ, USA, 2013; pp. 1–10. [CrossRef]
3. Eurostat. 2018. Available online: <https://www.statista.com/statistics/418078/electricity-prices-for-households-in-germany> (accessed on 28 February 2019).
4. Egert, R.; Tundis, A.; Roth, S.; Mühlhäuser, M. *A Service Quality Indicator for Apriori Assessment and Comparison of Cellular Energy Grids*; Kaparaju, P., Howlett, R.J., Littlewood, J., Ekanyake, C., Vlacic, L., Eds.; Sustainability in Energy and Buildings 2018; Springer International Publishing: Cham, Switzerland, 2019; pp. 322–332.
5. Rogovchenko-Buffoni, L.; Tundis, A.; Hossain, M.Z.; Nyberg, M.; Fritzson, P. An integrated toolchain for model based functional safety analysis. *J. Comput. Sci.* **2014**, *5*, 408–414. [CrossRef]
6. Tundis, A.; Egert, R.; Mühlhäuser, M. Applying a Properties Modeling approach for monitoring Smart Grids. In Proceedings of the 2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC), Calabria, Italy, 16–18 May 2017; pp. 714–719. [CrossRef]
7. Mahmood, A.; Butt, A.R.; Mussadiq, U.; Nawaz, R.; Zafar, R.; Razzaq, S. Energy sharing and management for prosumers in smart grid with integration of storage system. In Proceedings of the 2017 5th International Istanbul Smart Grid and Cities Congress and Fair (ICSG), Istanbul, Turkey, 21 April 2017; pp. 153–156. [CrossRef]

8. Zafar, R.; Mahmood, A.; Razzaq, S.; Ali, W.; Naeem, U.; Shehzad, K. Prosumer based energy management and sharing in smart grid. *Renew. Sustain. Energy Rev.* **2018**, *82*, 1675–1684. [[CrossRef](#)]
9. Egert, R.; Cordero, C.G.; Tundis, A.; Mühlhäuser, M. HOLEG: A simulator for evaluating resilient energy networks based on the Holon analogy. In Proceedings of the 2017 IEEE/ACM 21st International Symposium on Distributed Simulation and Real Time Applications (DS-RT), Rome, Italy, 18–20 October 2017; pp. 1–8. [[CrossRef](#)]
10. Egert, R.; Tundis, A.; Volk, F.; Mühlhäuser, M. An integrated tool for supporting the design and virtual evaluation of smart grids. In Proceedings of the 2017 IEEE International Conference on Smart Grid Communications (SmartGridComm), Beijing, China, 21–24 October 2017; pp. 259–264. [[CrossRef](#)]
11. Alam, M.R.; Reaz, M.B.I.; Ali, M.A.M. A Review of Smart Homes—Past, Present, and Future. *IEEE Trans. Syst. Man Cybern. C (Appl. Rev.)* **2012**, *42*, 1190–1203. [[CrossRef](#)]
12. Smart Home: The Human Side of the Smart Grid. 2010. Available online: <http://www.smartgrids-cre.fr/media/documents/1003-CapG-SmartHome.pdf> (accessed on 28 February 2019).
13. Abeykoon, V.; Kankanamdurage, N.; Senevirathna, A.; Ranaweera, P.; Udawalpola, R. Real Time Identification of Electrical Devices through Power Consumption Pattern Detection. *Pervasive Comput.* **2016**, *10*, 40–48.
14. Tracebase. 2017. Available online: <http://www.tracebase.org> (accessed on 28 February 2019).
15. Reinhardt, A.; Baumann, P.; Burgstahler, D.; Hollick, M.; Chonov, H.; Werner, M.; Steinmetz, R. On the accuracy of appliance identification based on distributed load metering data. In Proceedings of the Sustainable Internet and ICT for Sustainability, Pisa, Italy, 4–5 October 2012; pp. 1–9.
16. Zufferey, D.; Gisler, C.; Khaled, O.; Hennebert, J. Machine learning approaches for electric appliance classification. In Proceedings of the 2012 11th International Conference on Information Science, Signal Processing and their Applications, Montreal, QC, Canada, 2–5 July 2012; pp. 740–745.
17. PLOGG—Energy Optimizers Limited—Plogg Wireless Energy Management. 2019. Available online: <https://www.automatedhome.co.uk/new-products/plogg-appliance-energy-metering-and-control.html> (accessed on 28 February 2019).
18. Morsali, H.; Shekarabi, S.M.; Ardekani, K.; Khayami, H.; Fereidunian, A.; Ghassemian, M.; Lesani, H. Smart plugs for building energy management systems. In Proceedings of the Iranian Conference on Smart Grids, Tehran, Iran, 24–25 May 2012; pp. 1–5.
19. Hart, G.W. Residential Energy Monitoring and Computerized Surveillance via Utility Power Flows. *Technol. Soc. Mag.* **1989**, *8*, 12–16. [[CrossRef](#)]
20. Ruzzelli, A.G.; Nicolas, C.; Schoofs, A.; O’Hare, G.M.P. 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), Boston, MA, USA, 21–25 June 2010; pp. 1–9.
21. Patel, S.N.; Robertson, T.; Kientz, J.A.; Reynolds, M.S.; Abowd, G.D. At the Flick of a Switch: Detecting and Classifying Unique Electrical Events on the Residential Power Line (Nominated for the Best Paper Award). In Proceedings of the UbiComp, Innsbruck, Austria, 16–19 September 2007.
22. Ridi, A.; Gisler, C.; Hennebert, J. Automatic identification of electrical appliances using smart plugs. In Proceedings of the 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA), Algiers, Algeria, 12–15 May 2013; pp. 301–305.
23. Gulati, M.; Ram, S.S.; Singh, A. An in Depth Study into Using EMI Signatures for Appliance Identification. In Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings, Memphis, TN, USA, 4–6 November 2014; ACM: New York, NY, USA, 2014; pp. 70–79. [[CrossRef](#)]
24. HFED—High Frequency Energy Dataset. 2014. Available online: <http://hfed.github.io/> (accessed on 6 May 2019).
25. Rashid, H.; Batra, N.; Singh, P. Rimor: Towards Identifying Anomalous Appliances in Buildings. In Proceedings of the 5th Conference on Systems for Built Environments, Shenzhen, China, 7–8 November 2018; ACM: New York, NY, USA, 2018; pp. 33–42. [[CrossRef](#)]
26. Géron, A. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques for Building Intelligent Systems*, 1st ed.; O’Reilly: Springfield, MO, USA, 2017.

27. Kavitha, S.; Varuna, S.; Ramya, R. A comparative analysis on linear regression and support vector regression. In Proceedings of the 2016 Online International Conference on Green Engineering and Technologies (IC-GET), Kuala Lumpur, Malaysia, 25–27 July 2016; pp. 1–5. [CrossRef]
28. Hu, Q.; Che, X.; Zhang, L.; Zhang, D.; Guo, M.; Yu, D. Rank Entropy-Based Decision Trees for Monotonic Classification. *IEEE Trans. Knowl. Data Eng.* **2012**, *24*, 2052–2064. [CrossRef]
29. Perner, P. How to Compare and Interpret Two Learnt Decision Trees from the Same Domain? In Proceedings of the 2013 27th International Conference on Advanced Information Networking and Applications Workshops, Barcelona, Spain, 25–28 March 2013; pp. 318–322. [CrossRef]
30. Han, B.; Davis, L.S. Density-Based Multifeature Background Subtraction with Support Vector Machine. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1017–1023. [CrossRef] [PubMed]
31. Cheng, G.; Tong, X. Fuzzy Clustering Multiple Kernel Support Vector Machine. In Proceedings of the 2018 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR), Chengdu, China, 15–18 July 2018; pp. 7–12. [CrossRef]
32. Jahromi, A.H.; Taheri, M. A non-parametric mixture of Gaussian naive Bayes classifiers based on local independent features. In Proceedings of the 2017 Artificial Intelligence and Signal Processing Conference (AISP), Shiraz, Iran, 25–27 October 2017; pp. 209–212. [CrossRef]
33. Valecha, H.; Varma, A.; Khare, I.; Sachdeva, A.; Goyal, M. Prediction of Consumer Behaviour using Random Forest Algorithm. In Proceedings of the 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Okinawa, Japan, 2–4 November 2018; pp. 1–6. [CrossRef]
34. Niranjana, A.; Prakash, A.; Veena, N.; Geetha, M.; Deepa Shenoy, P.; Venugopal, K.R. EBJRV: An Ensemble of Bagging, J48 and Random Committee by Voting for Efficient Classification of Intrusions. In Proceedings of the 2017 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), Uttarkhand, India, 18–19 December 2017; pp. 51–54. [CrossRef]
35. Khoshgoftaar, T.M.; Van Hulse, J.; Napolitano, A. Comparing Boosting and Bagging Techniques with Noisy and Imbalanced Data. *IEEE Trans. Syst. Man Cybern. A Syst. Hum.* **2011**, *41*, 552–568. [CrossRef]
36. Wang, B.; Pineau, J. Online Bagging and Boosting for Imbalanced Data Streams. *IEEE Trans. Knowl. Data Eng.* **2016**, *28*, 3353–3366. [CrossRef]
37. Kapoor, S.; Singh, A.; Goswami, U.; Kumar, S.; Chitranshi, G. Design and implementation of a robust system for recognizing alphabets using artificial neural network. In Proceedings of the 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 7–9 September 2016; pp. 606–609. [CrossRef]
38. Open Database License. 2017. Available online: <http://opendatacommons.org/licenses/odbl/1.0/> (accessed on 28 February 2019).
39. Xu, H.; Yang, M.; Liang, L. An improved random decision trees algorithm with application to land cover classification. In Proceedings of the 2010 18th International Conference on Geoinformatics, Beijing, China, 18–20 June 2010; pp. 1–4. [CrossRef]
40. Kauschke, S.; Fürnkranz, J. Batchwise Patching of Classifiers. In Proceedings of the 32nd AAI Conference on Artificial Intelligence (AAAI-18), New Orleans, LA, USA, 2–7 February 2018; pp. 3374–3381.

