

Article

Revealing Household Characteristics from Electricity Meter Data with Grade Analysis and Machine Learning Algorithms

Krzysztof Gajowniczek ^{1,*} , Tomasz Ząbkowski ¹ and Mariya Sodenkamp ²

¹ Department of Informatics, Faculty of Applied Informatics and Mathematics, Warsaw University of Life Sciences SGGW, Warsaw 02-776, Poland; tomasz_zabkowski@sggw.pl

² Information Systems and Energy Efficient Systems Group, Information Systems and Applied Computer Sciences, University of Bamberg, Bamberg 96047, Germany; mariya.sodenkamp@uni-bamberg.de

* Correspondence: krzysztof_gajowniczek@sggw.pl; Tel.: +48-506-746-850

Received: 22 August 2018; Accepted: 12 September 2018; Published: 14 September 2018



Abstract: In this article, the Grade Correspondence Analysis (GCA) with posterior clustering and visualization is introduced and applied to extract important features to reveal households' characteristics based on electricity usage data. The main goal of the analysis is to automatically extract, in a non-intrusive way, number of socio-economic household properties including family type, age of inhabitants, employment type, house type, and number of bedrooms. The knowledge of specific properties enables energy utilities to develop targeted energy conservation tariffs and to assure balanced operation management. In particular, classification of the households based on the electricity usage delivers value added information to allow accurate demand planning with the goal to enhance the overall efficiency of the network. The approach was evaluated by analyzing smart meter data collected from 4182 households in Ireland over a period of 1.5 years. The analysis outcome shows that revealing characteristics from smart meter data is feasible, and the proposed machine learning methods were yielding for an accuracy of approx. 90% and Area Under Receiver Operating Curve (AUC) of 0.82.

Keywords: smart metering; Grade Correspondence Analysis; machine learning

1. Introduction

Electricity providers are currently driving deployment of smart electricity meters in a number of households worldwide to collect fine-grained electricity usage data. The changes taking place in the electricity industry require effective methods to provide end users with the feedback on electricity usage which is in turn used by the network operators for formulating pricing strategies, constructing tariffs and undertaking actions to improve the efficiency and reliability of the distribution grid. With high expectations towards smart metering adoption and its influence on households notwithstanding, it is observed that utilization of the information from fine-grained consumption profiles is in its initial stage. This is due to the fact that consumption patterns of individual residential customers vary a lot which is the function of the number of inhabitants, their activity, age and lifestyle [1]. Various techniques for customer classification are discussed in the literature, with the focus on electricity usage behavior of the customers [2–5]. These works contribute to higher energy awareness by providing the input for demand response systems in homes and supporting accurate usage forecasting on the household level [6–8].

Recently, a new relevant research stream may be distinguished with the underlying idea to identify important household characteristics and leverage it for energy efficiency. It is focused on the application

of supervised machine learning techniques for inferring such household properties as number of inhabitants including children, family type, size of the house, and many other characteristics [9,10]. In particular, this work relies upon the works of Beckel et al. and Hopf et al. and further it is supposed to enhance the approach by extending the methodology for features selection. Therefore, this paper applies the GCA segmentation approach to derive important features describing electricity usage patterns of the households. The knowledge of the load profiles captured by smart meters might be helpful to reveal relevant household characteristics. These customer insights can be further utilized to optimize the energy efficiency programs in many ways, including with the introduction of flexible tariff plans and enhanced feedback loop [11,12]. The later one applies to feedback programs that engage households in energy saving behaviors, and helps to recognize what actions inhabitants are undertaking to bring the feedback into energy savings [13].

In particular, the proposed paper enhances methodology for customer classification taking into account historical electricity consumption data captured by a large set of 91 attributes, tailored specially to describe various aspects of behaviors typical for different type of households. Therefore, the scope of the paper is threefold:

- (1) Extraction of the comprehensive set of the behavioral features to capture different aspects of household characteristics;
- (2) Application of grade cluster analysis to identify important attributes to detect distinct consumption patterns of the customers and further, using only a subset of relevant features for classification, to reveal socio-demographic characteristics of the households;
- (3) Classification of households' properties using three machine learning algorithms and three feature selection techniques.

The proposed research fits into the attempt focused on leveraging smart meter data to support energy efficiency on the individual user level. This gives novel research challenges in monitoring usage, data gathering, and inferring from data in a non-intrusive way since customer classification and profiling is methodically sound and offers a variety of potentials for application within the energy industry [14–16]. In the attempt to reduce electricity consumption in buildings, identification of important features responsible for specific patterns of energy consumption at different customer groups is a key to improving efficiency of available energy usage.

In this context, the proposed approach is, to some extent, similar to non-intrusive load monitoring (NILM) or non-intrusive appliance load monitoring (NIALM) [17–19]. However, the difference is that our goal is to extract high-level household characteristics from the electricity consumption instead of disaggregating the consumption of individual appliances. Nevertheless, both approaches—NILM/NIALM and the proposed approach for detecting households' characteristics—are delivering interesting knowledge that has implications for households and utility providers. It may help them to understand the key drivers responsible for the electricity consumption and, finally, the costs associated with this.

In the following sections we characterize the data used in the experiments and introduce the idea of grade analysis. Subsequently, we describe the technical and methodological realization of the classification as well as the evaluation of the results. The final section provides a summary and an outlook on further application scenarios.

2. Smart Meter Data Used

2.1. The CER Data Set

This research is conducted based on the Irish Commission for Energy Regulation (CER) data set. The CER initiated a Smart Metering Project in 2007 with the purpose of undertaking trials to assess the performance of Smart Meters and their impact on consumer behavior. It contains measurements of electricity consumption gathered from 4182 households between July 2009 and December 2010

(75 weeks in total with 30 min data granularity). Each participating household was asked to fill out a questionnaire before and after the study. The questionnaire contained inquiries regarding the consumption behavior of the occupants, the household's socio-economic status, properties of the dwelling and appliance stock [20].

Some characteristics of the underlying data are presented in Figure 1, where the normalized consumption observed at different aggregation levels is visualized. Aggregation reduces the variability in electricity consumption resulting in increasingly smooth load shapes when at least 100 households are considered.

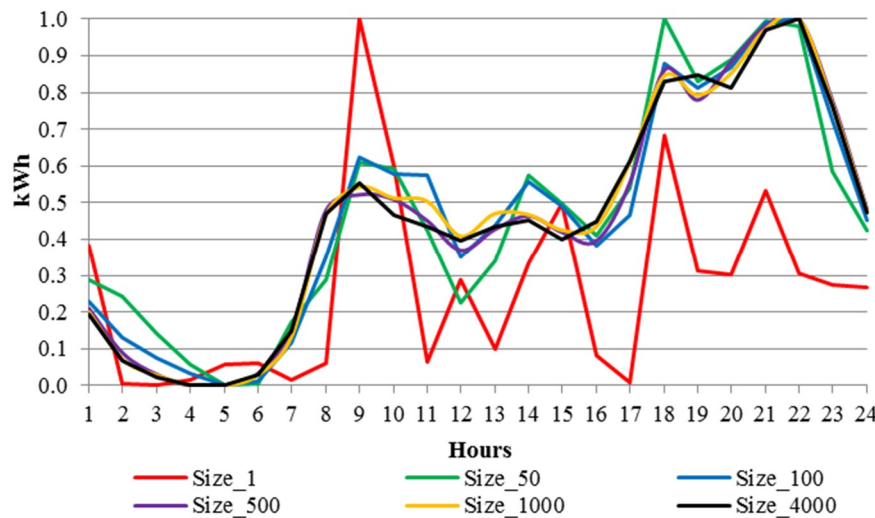


Figure 1. Hourly electricity consumption for various aggregation levels.

The CER data set, to the best of our knowledge, does not account for energy that is consumed by heating and cooling systems. The heating systems of the participating households either use oil or gas as a source of energy or their consumption is measured by a separate electricity meter. The households registered in the project were reported to have no cooling system installed [20].

2.2. Features

The definition of features vector is crucial to the success of any classifier based on a machine learning algorithm. To make the high-volume time series data applicable to the classification problem, they have to be transformed into a number of representative variables. As suggested in [10,20], features can be divided in four groups: consumption features, ratios, temporal features, and statistics. This set of features especially considers the relation between the consumption on weekdays and on the weekend, parameters of seasonal and trend decomposition, estimation of the base load and some statistical features (please refer to Table 1). Altogether the attributes describe consumption characteristics (such as mean consumption at different times of the day and on different days), ratios (e.g., daytime-ratios and ratios between different days), statistical aspects (e.g., the variance, the auto-correlation and other statistical numbers) and finally different temporal aspects (such as consumption levels, peaks, important moments, temporal deviations, values of time series analysis) [10,20].

All attributes were created based on time series, so we did not apply any dimensionality reduction techniques e.g., Principal Component Analysis in order not to reduce interpretability of a particular variable and to prevent information loss. After the feature extraction, the values are normalized. To evaluate algorithms, we have separated the data into training and testing dataset at a 70%:30% ratio.

Table 1. List of 91 features used in the analysis.

Consumption	Ratios	Statistical	Temporal
c_week	r_night_day	s_variance	t_above_base
c_weekday	r_morning_noon	s_cor_wd	t_above_1kw
c_weekend	r_evening_noon	s_num_peaks	t_above_2kw
c_evening	r_mean_max	s_diff	t_above_mean
c_morning	r_min_mean	s_q1	t_daily_max
c_night	r_evening_wd_we	s_q2	t_daily_min
c_noon	r_night_wd_we	s_q3	ts_acf_mean3h
c_min	r_morning_wd_we	s_min_avg	ts_acf_mean3h_weekday
c_max	r_noon_wd_we	s_max_avg	ts_stl_varRem
c_we_max	r_afternoon_wd_we	s_var_we	b_day_diff
c_we_evening	r_min_wd_we	s_var_wd	b_day_weak
c_wd_evening	r_max_wd_we	s_cor_wd_we	wide_peaks
c_we_night	r_var_wd_we	s_cor_we	width_peaks
c_wd_night	r_we_night_day	n_d_diff	sm_variety
c_we_morning	r_wd_night_day	number_zeros	bg_variety
c_wd_morning	r_we_morning_noon		time_above_base2
c_we_noon	r_wd_morning_noon		percent_above_base
c_wd_noon	r_we_evening_noon		value_above_base
c_we_afternoon	r_wd_evening_noon		const_time
c_wd_afternoon	r_mean_max_no_min		value_min_guess
c_afternoon	r_evening_noon_no_min		first_above_base
c_we_min	r_morning_noon_no_min		number_big_peaks
c_wd_max	r_day_night_no_min		number_small_peaks
c_wd_min			dist_big_v
c_sm_max			
c_evening_no_min			
c_morning_no_min			
c_night_no_min			
c_noon_no_min			

3. Grade Data Analysis

In the following lines, Grade Data Analysis is presented. It is an interesting technique that works on variables measured on any scale, including categorical. The method uses dissimilarity measures including concentration curves and the measure of monotonic dependence. The framework is based on grade transformation proposed by [21], and developed by [22]. The general idea is to transform any distribution of two variables into a structure that enables to capture the underlying dependencies of the so-called grade distribution. In practical applications, the grade data approach consists of analyzing the two-way table with rows/columns, which is preceded by proper recoding of variable values and providing the values of monotone dependence measures like Spearman’s ρ^* and Kendall’s τ .

The main component of the grade methods is Grade Correspondence Analysis (GCA), which stems from classical correspondence analysis. Importantly, Grade Data Analysis is going significantly beyond the correspondence approach, thanks to the means of grade transformation. An important feature of GCA is that it does not create a new measure but takes into account the original structure of the underlying phenomenon. GCA performs multiple ordering iterations on both the columns and the rows of the table, in such a way that neighboring rows are more similar than those further apart, and at the same time, neighboring columns are more similar than those that are further apart. Once the optimal structure is found, it is possible to combine neighboring rows and neighboring columns, and therefore, to build the clusters representing similar distributions. The Spearman ρ^* was originally proposed for continuous distributions, however it may be defined also as Pearson’s correlation applied to the distribution after the grade transformation. Importantly, the grade distribution is applicable for

discrete distribution too, and it is possible to calculate Spearman ρ^* for the probability table P with m rows and k columns, where p_{is} is the probability of i -th row in s -th column:

$$\rho^*(P) = 3 \sum_{i=1}^m \sum_{s=0}^k \left(p_{is} \left(2S_{row(i)} - 1 \right) \left(2S_{col(s)} - 1 \right) \right), \tag{1}$$

where

$$S_{row(i)} = \left(\sum_{j=1}^{i-1} p_{j+} \right) + \frac{1}{2} p_{i+}, \quad S_{col(s)} = \left(\sum_{t=1}^{s-1} p_{+t} \right) + \frac{1}{2} p_{+s} \tag{2}$$

and p_{j+} and p_{+t} are marginal sums defined as: $p_{j+} = \sum_{s=1}^k p_{js}$, $p_{+t} = \sum_{i=1}^m p_{it}$.

GCA is supposed to maximize ρ^* by ordering the columns and the rows taking into account their grade regression value, which represents the gravity center for each column or each row. The grade regression for the rows is defined as:

$$regr_{row(i)} = \frac{\sum_{s=1}^k p_{is} S_{col(s)}}{p_{i+}} \tag{3}$$

and, similarly, for the columns:

$$regr_{col(s)} = \frac{\sum_{i=1}^m p_{is} S_{row(i)}}{p_{+s}}. \tag{4}$$

The idea behind the algorithm is to measure the grade regression for columns and to sort the columns by its values, which results in an increase of the regression for columns. At the same time, the regression for rows changes as well. Similarly, if the regression for rows is sorted then regression for columns changes. As evidenced in [23], each sorting iteration with respect to grade regression values, in fact, increases the value of Spearman ρ^* . The number of possible combinations with rows and columns permutations is finite and it is equal to $k!m!$. With the increasing value of Spearman ρ^* , the last sorting iteration produces the largest ρ^* , called local maximum of Spearman ρ^* .

In consecutive steps, GCA randomly permutes rows and columns and reorders them so local maximum can be achieved. In practical application, when the data volume and dimension is huge, the search over the all possible combinations of rows and columns is a computationally demanding and long-lasting process. Therefore, in order to find a global maximum of ρ^* , Monte Carlo simulations are used. To achieve it, the algorithm is iteratively searching for such a representation where ρ^* reaches local maximum, starting from randomly ordered rows and columns. From the whole set of local maxima, the highest value of ρ^* is chosen and it is assumed to be close to the global maximum, which usually happens after 100 iterations of the algorithm. Importantly, the calculation of grade regression requires non-zero sum for each and every row and column in a table, so this requirement is applicable also to the GCA. A more detailed description on grade transformation mechanics can be found in [22,24].

As far as grade cluster analysis (GCCA) is concerned, its framework is based on optimal permutations provided by the GCA. The following assumptions are associated with the cluster analysis: the number of clusters is provided, and the rows and columns of the data table (variables, say X and Y) are optimally aggregated. The respective, aggregated probabilities in the table for cluster analysis, are derived from the sums of component probabilities which are found in initial, optimally ordered table, and number of rows in the aggregated table equals the specified number of clusters. The optimal clustering is supposed to be achieved when $\rho^*(X, Y)$ is maximal in the set of aggregated rows and/or columns, which are adjacent in optimal permutations. The rows and the columns may be combined either separately—by maximizing ρ^* for aggregated X and non-aggregated Y , or for non-aggregated X and aggregated Y , or simultaneously. Details of the maximization procedure can be found in [23].

Finally, the grade analysis is highly supported by visualizations using over-representation maps. The maps are acting as a very convenient tool for plotting both source and transformed data structures

where the idea is to show the various structures in the data with respect to the average values. Every cell in the data table is covered by the respective rectangle in $[0, 1] \times [0, 1]$ space and it is visualized using shades of grey, which corresponds to the level of the randomized grade density. The scale of grade density is divided into several intervals and respective colors represent particular intervals, with black corresponding to the highest values and white corresponding to the lowest. With the grade density used to measure the deviation from independence of variables X and Y , the dark colors indicate overrepresentation while the white ones show underrepresentation.

4. GCA Clustering Experiments

The starting point for the experiments was to prepare the initial matrix with normalized features $(x_i - \min(x)) / (\max(x) - \min(x))$ in the columns and the rows representing each of the households. The structure of the dataset is presented in Table 2.

Table 2. The sample matrix with the features extracted for each of the households.

Household	Feature_1	Feature_2	...	Feature_91
1	0.23	0.57	...	0.85
2	0.64	0.77	...	0.27
...
4182	0.51	0.73	...	0.63

The data structure presented in Table 2 has been analyzed using GradeStat software [25], which is the tool that was developed in the Institute of Computer Science Polish Academy of Science.

The next step was to compute over-representation ratios for each field (cell) of the table with households and the attributes describing them. For a given $m \times k$ data matrix with non-negative values, a visualization using over-representation map is possible, in the same way as a contingency table. However, instead of frequency n_{ij} the value of j -th feature for i -th household is used. Subsequently, it is compared in a contingency table with the corresponding neutral or fair representation of $n_{i\bullet} \times n_{\bullet j} / \sum \sum n_{ij}$ where $n_{i\bullet} = \sum_j n_{ij}$, $n_{\bullet j} = \sum_i n_{ij}$. The ratio of the expression is called the over-representation. An over-representation surface over a unit square is then divided into $m \times k$ cells situated in m rows and k columns, and the area of cells placed in row i and column j is assumed to be equal to fair representation of normalized n_{ij} . Based on the over-representation ratios, the over-representation map for the initial raw data can be constructed. The color intensity of each cell in the map is the result of the comparison between two values: (1) the real value of the measure connected to the underlying cell; (2) the expected value of the measure. In Figure 2 there is an initial over-representation map for the analyzed data presented. The colors of the cells in the map are grouped into three classes representing different properties:

- *gray*—the feature for the element (household) is **neutral** (ranging between the 0.99–1.01) which means that the real value of the feature is equal to its expected value;
- *black* or *dark gray*—the feature for the element (household) is **over-represented** (between 1.01 and 1.5 for weak over-representation and more than 1.5 for strong) which means that the real value of the feature is greater than the expected one;
- *light gray* or *white*—the feature for the element (household) is **under-represented** (between 0.66 and 0.99 for weak under-representation and less than 0.66 for strong under-representation), which means that the real value of feature is less than the expected one.

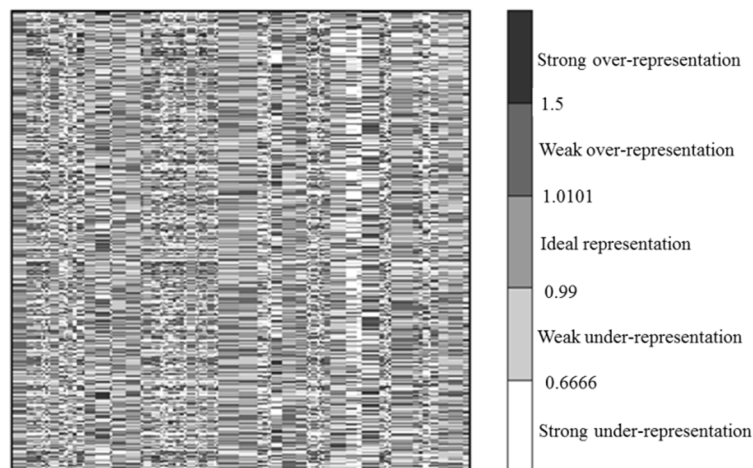


Figure 2. The initial over-representation map.

Besides the differences in color's scales on the map—its rows and columns could be of different sizes. A row's height depends on the evaluation of the element (household) in comparison to the entire population, so the households with higher evaluation are represented by higher rows. A column's width depends on the evaluation of the element (feature) in comparison to the evaluation of all the features from the set, so the features with higher evaluation are represented by wider columns.

In order to reveal the structural trends in data, the following step was to apply the grade analysis to measure the dissimilarity between analyzed data distributions—households and feature dimensions. The grade analysis was conducted based on Spearman's ρ^* , used as the total diversity index. The value of ρ^* strongly depends on the mutual order of the rows and the columns and therefore, to calculate ρ^* , the concentration indexes of differentiation between the distributions were used. The basic GCA procedure is executed through permuting the rows and columns of a table in order to maximize the value of ρ^* . After each sorting, the ρ^* value increases and the map becomes more similar to the ideal one. As presented on the maps, the darkest fields are placed in the upper-left and the lower-right corners while the rest of the fields are assigned according to the following property: the farther from the diagonal towards the two other map corners (the lower-left and the upper-right ones) the lighter gray (or white) color the fields have.

The result of the GCA procedure is presented in Figure 3. The rows represent households and the columns represent the features describing the households. The resulting order presents the structure of underlying trends in data. The analysis of the map reveals that two groups of the features can be distinguished: the features which non-differentiate the population of households (the middle columns of the map) and those which differentiate the households (the most-left and the most-right columns).

Four vertical clusters were marked in Figure 3 (C1, C2, C3 and C4) and these show typical behavior of the households in terms of the electricity usage characterized by the respective number of features (in brackets).

Finally, the aggregation of some rows representing unique households was performed. The optimal number of four clusters was obtained when the changes of the subsequent ρ^* values appeared to be irrelevant as referenced in [22]. In Figure 4, the chart with the ρ^* values as a function of the number of clusters is presented. The points on the *OX* axis correspond to the cluster numbers. The *OY* axis is denoted by the values of ρ^* .

The proposed GCA method applied for the clustering enables identification of the features describing different aspects of the consumption behaviors. The clusters are further utilized to select representative features within each cluster to be used for revealing selected households' characteristics.

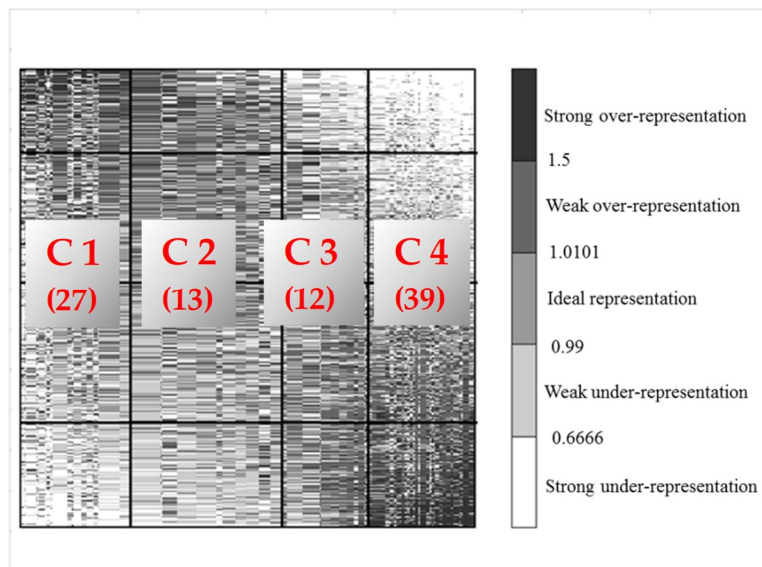


Figure 3. The final over-representation map with four clusters.

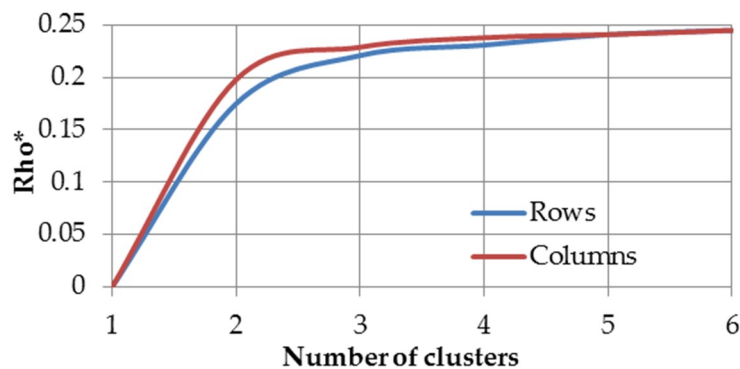


Figure 4. The ρ^* values for different number of clusters.

5. Classification of Selected Household Characteristics

5.1. Problem Statement

In the following lines we present and assess a classification system that applies supervised machine learning algorithms to automatically reveal specific patterns or characteristics of the households, having their aggregated electricity consumption as an input. The patterns/characteristics are related to the socio-economic status of a particular household and its dwelling. In particular, the following properties are explored:

- Family type;
- Number of bedrooms;
- Number of appliances;
- Employment;
- Floor area;
- House type;
- House age;
- Householder age.

Along with the detailed smart metering data, the data set provides information on the characteristics of each household collected through the questionnaires. Such information delivers true

output to classification to validate the proposed models. Table 3 presents eight questionnaire questions that were used as the target features for classification (true outcome).

Table 3. Questionnaire questions and their corresponding category labels.

Category	Person’s Age	Number of Appliances
	What Age Were You on Your Last Birthday?	Approximately How Many Appliances Are in Your Home?
1	18–35	≤8 appliances
2	36–65	between 9 and 11
3	65+	>11 appliances
Category	Number of Bedrooms	Floor Area
	How many bedrooms are in your home?	Approximately what is the area of your home?
1	≤2 bedrooms	Not available
2	3 bedrooms	<100 m ²
3	4 bedrooms	between 100 m ² and 200 m ²
4	≥5 bedrooms	>200 m ²
Category	Employment	Family type
	What is the employment status of the chief income earner in your household?	What best describes the people you live with?
1	An employee, Self-employed (with employees), Self-employed (with no employees)	I live alone
2	Unemployed (actively seeking work), Unemployed (not actively seeking work), Retired, Carer: Looking after relative family	All people in my home are over 15 years of age, both adults and children under 15 years of age live in my home
Category	House Age	House Type
	Approximately how old is your home?	Which best describes your home?
1	≤30 years	Semi-detached house, Terraced house
2	>30 years	Apartment, Detached house, Bungalow

For classification of the households’ properties, three experimental feature setups were considered:

- All the variables (91) were used in the algorithms;
- Eight variables based on GCA and selected as representatives of each cluster having the highest AUC measure (please refer to Appendix A, Table A1);
- Eight variables based on Boruta package which is the feature selection algorithm for finding relevant variables [26].

5.2. Accuracy Measures

For the purpose of model evaluation, four performance measures were used, i.e., classification accuracy, sensitivity, specificity and area under the ROC curve (AUC) [27]. For the binary classification problem, i.e., having positive class and negative class, four possible outcomes exist, as shown in Table 4.

Table 4. Confusion matrix for binary classification.

		Predicted Value	
		Positive (P)	Negative (N)
Real Value	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

Based on Table 4, the accuracy (AC) measure can be computed, which is the proportion of the total number of predictions that were correct:

$$AC = \frac{TP + TN}{TP + FP + TN + FN}. \tag{5}$$

AUC estimation requires two indicators defined as: true positive rate $Tpr = \frac{TP}{TP+FN}$, and false positive rate $Fpr = \frac{FP}{FP+TN} = 1 - Tnr$. These measures can be calculated for different decision threshold values. An increase of the threshold from 0 to 1 will yield a series of points (Fpr, Tpr) constructing the curve with Tpr and Fpr on the horizontal and vertical axes, respectively. In a general form, the value of AUC is given by $AUC = \int_0^1 ROC(u)du$.

From another point of view, AUC can be understood as $P(X_p > X_n)$ where X_p and X_n denote the markers for positive and negative cases, which can be interpreted as the probability that in a randomly drawn pair of positive and negative cases, the classifier probability is higher for the positive one.

5.3. Classification Algorithms

Building predictive models involves complex algorithms, therefore R-CRAN was used as the computing environment. In this research, all the numerical calculations were performed on a personal computer equipped with an Intel Core i5-2430M 2.4 GHz processor (2 CPU × 2 cores), 8 GB RAM and the Ubuntu 16.04 LTS operating system. To achieve predictive models having good generalization abilities, special learning process incorporating AUC measure was performed. Because of this, the following maximized function assures the best parameters of each algorithm:

$$f(AUC_T, AUC_V) = -\frac{1}{2}|AUC_T - AUC_V| + \frac{1}{2}AUC_V, \tag{6}$$

where AUC_T stands for the training accuracy, and AUC_V stands for the validation accuracy.

5.3.1. Artificial Neural Networks

Artificial neural networks (ANN) are mathematical objects in the form of equations or systems of equations, usually nonlinear, for analysis and data processing. The purpose of neural networks is to convert input data into output data with a specific characteristic or to modify such systems of equations to read useful information from their structure and parameters. On a statistical basis, selected types of neural networks can be interpreted in general non-linear regression categories [28].

In studies related to forecasting in power engineering, multilayer, one-way artificial neural networks with no feedback are most commonly used. Multilayer Perceptron networks (MLP) are one of the most popular types of supervised neural networks. For example, the MLP network (3, 4, 1) means a neural network with three inputs, four neurons in the hidden layer and one neuron in the output layer. In general, the three-layer MLP neural network (P, M, K) is described by the expression:

$$f(x_i, w) = h_2(\mathbf{W}_2[h_1(\mathbf{W}_1x_i + \mathbf{b}_1)] + \mathbf{b}_2), \tag{7}$$

where $x_i = (x_1, \dots, x_p)^T$ represents the input data, \mathbf{W}_1 is the matrix of the first layer weights with dimensions $M \times P$, \mathbf{W}_2 is the matrix of the second layer weights with dimensions $K \times M$, $h_i(\mathbf{u})$ and \mathbf{b}_i are nonlinearities (functions of neuron activation e.g., logistic function) and constant values in subsequent layers respectively [28].

The goal of supervised learning of the neural network is to search for such network parameters that minimize the error between the desired values L_i and received at the output of the network P_i . The most frequently minimized error function is the sum of the squares of differences between the

actual value of the explained variable and its theoretical value determined by the model, with the values of the synaptic weight vector set:

$$E(w) = \frac{1}{2} \sum_{k=1}^K e^{(k)} = \frac{1}{2} \sum_{k=1}^K \left(\sum_{i=1}^n (P_i^{(k)} - L_i^{(k)})^2 \right), \tag{8}$$

where n is the number of the training sample, $P_i^{(k)}$ and $L_i^{(k)}$ are predicted and reference value and K is the number of training epochs of the neural network [28].

The neural network learning process involves the iterative modification of the values of the synaptic weight vector w (all weights are set in one vector), in iteration $k + 1$:

$$w_{k+1} = w_k + \eta_k p_k, \tag{9}$$

where p_k is the direction of the minimization of the function $E(w)$ and η is the magnitude of the learning error. The most popular optimization methods are undoubtedly gradient methods, which are based on the knowledge of the function gradient:

$$p_k = -[H(w_k)]^{-1} g(w_k), \tag{10}$$

where g and H denote the gradient and the hessian of the last known solution w_k , respectively [28].

In the practical implementations of the algorithm, the exact determination of hessian $H(w_k)$ is abandoned, and its approximation $G(w_k)$ is used instead. One of the most popular methods of learning neural networks is the algorithm of variable metrics. In this method, the hessian (or its reversal) in each step is modified from the previous step by some correction. If by c_k and r_k the increments of the vector w and the gradient g in two successive iterative steps are marked, $c_k = w_k - w_{k-1}$, $r_k = g_k - g_{k-1}$, and by V_k the inverse matrix of the approximate hessian $V_k = [G(w_k)]^{-1}$, $V_{k-1} = [G(w_{k-1})]^{-1}$, according to the most effective formula of Broyden-Fletcher-Goldfarb-Shanno (BFGS), the process of updating the value of the V_k matrix is described by the recursive relationship:

$$V_k = V_{k-1} + \left(1 + \frac{r_k^T V_{k-1} r_k}{c_k^T r_k} \right) \frac{c_k c_k^T}{c_k^T r_k} - \frac{c_k r_k^T V_{k-1} + V_{k-1} r_k c_k^T}{c_k^T r_k}. \tag{11}$$

As a starting value $V_0 = 1$ is usually assumed, and the first iteration is carried out in accordance with the algorithm of the largest slope [28].

Artificial neural networks are often used to estimate or approximate functions that can depend on a large number of inputs. In contrast to the other machine learning algorithms considered in these experiments, the ANN required the input data to be specially prepared. The vector of continuous variables was standardized, whereas the binary variables were converted such that 0 s were transformed into values of -1 [3,5,29].

To train the neural networks, we used the BFGS algorithm implemented in the *nnet* library. The network had an input layer with 91 neurons and a hidden layer with 1, 2, 3, ..., 15 neurons. A logistic function was used to activate all of the neurons in the network. To achieve robust estimation of the neural networks error, 10 different neural networks were learned with different initial weights vector. Final estimation of the error was computed as the average value over 10 neural networks [3,5,29].

In each experiment, 15 neural networks were learned with various parameters (the number of neurons in the hidden layer). To avoid overfitting, after each learning iteration had finished (with a maximum of 50 iterations), the models were checked using the measure defined in (6). Finally, out of the 15 learned networks, that with the highest value was chosen as the best for prediction [3,5,29].

5.3.2. K-Nearest Neighbors Classification

The k -nearest neighbors (KNN) regression [30] is a non-parametric method, which means that no assumptions are made regarding the model that generates the data. Its main advantage is the simplicity of the design and low computational complexity. The prediction of the value of the explained variable L_i on the basis of the vector of explanatory variables x_i is determined as:

$$P_i = \frac{\sum_{k=1}^K L_k I(x_i, x_k)}{K}, \tag{12}$$

where:

$$I(x_i, x_k) = \begin{cases} 1, & \text{if } x_k \text{ is one of the } k \text{ nearest neighbors } x_i \\ 0, & \text{otherwise} \end{cases}, \tag{13}$$

whereas x_k is one of the k -nearest neighbors x_i , in the case where the distance $d(x_i, x_k)$ belongs to k , the smallest distance between the observations from the set \mathbf{X} and x_k . The most commonly used distance is the Euclid distance [3,5,29,30].

To improve the algorithm, we normalized the explanatory variables (standardization for quantitative variables and replacement of 0 by -1 for binary variables). The normalization ensures that all dimensions for which the Euclidean distance is calculated have the same importance. Otherwise, a single dimension could dominate the other dimensions [3,5,29].

The algorithm was trained with knn implemented in the *caret* library. Different values of k were investigated in the experiments: {5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 250, 300}. The optimal value, and thus the final form of the model, was determined as that giving the maximum value according to (6) [3,5,29].

5.3.3. Support Vector Classification

Support Vector learning is based on simple ideas which originated in statistical learning theory [31]. The simplicity comes from the fact that Support Vector Machines (SVMs) apply a simple linear method to the data but in a high-dimensional feature space non-linearly related to the input space. Moreover, even though we can think of SVMs as a linear algorithm in a high-dimensional space, in practice, it does not involve any computations in that high-dimensional space [28].

SVMs use an implicit mapping Φ of the input data into a high-dimensional feature space defined by a kernel function, i.e., a function returning the inner product $\langle \Phi(x_i), \Phi(x'_i) \rangle$ between the images of two data points x_i, x'_i in the feature space. The learning then takes place in the feature space, and the data points only appear inside dot products with other points [32]. More precisely, if a projection $\Phi : \mathbf{X} \rightarrow \mathbf{H}$ is used, the dot product $\langle \Phi(x_i), \Phi(x'_i) \rangle$ can be represented by a kernel function k which is computationally simpler than explicitly projecting x_i and x'_i into the feature space \mathbf{H} [28].

Training an SVM involves solving a quadratic optimization problem. Using a standard quadratic problem solver for training an SVM would involve solving a big QP problem even for a moderately sized data set, including the computation of an $n \times n$ matrix in memory (n number of training points). In general, predictions correspond to the decision function:

$$P_i = \text{sign}(\langle w, \Phi(x_i) \rangle), \tag{14}$$

where solution w has an expansion $w = \alpha_i \sum_i \Phi(x_i)$ in terms of a subset of training patterns that lie on the margin [25].

In the case of the L2-norm soft margin classification, the primal optimization problem takes the form:

$$\text{minimize } (t, w) = \frac{1}{2} \|w\|^2 + \frac{C}{n} \sum_{i=1}^n (\xi_i), \tag{15}$$

$$\begin{aligned} \text{subject to } L_i(\langle \Phi(x_i), w \rangle + b) &\leq 1 - \xi_i^* \\ \xi_i &\geq 0 \quad (i = 1, \dots, n) \end{aligned} \quad (16)$$

where n is the number of training patterns, and $L_i \neq 1$, C is the cost parameter that controls the penalty paid by the SVM for misclassifying a training point and thus the complexity of the prediction function. A high cost value C will force the SVM to create a complex enough prediction function to misclassify as few training points as possible, while a lower cost parameter will lead to a simpler prediction function.

To construct the support vector machine model, C-SVR from the *kernelab* library with sequential minimal optimization (SMO) was used to solve the quadratic programming problem. A linear, polynomial (of degree 1, 2 and 3) and radial (γ from 0.1 to 1 by 0.2) kernel function were used, and ε (which defines the margin width for which the error function is zero) was arbitrarily taken from the following set {0.1, 0.3, 0.5, 0.7, 0.9}. The regularized parameter C that controls overfitting was arbitrarily set to one of the following values {0, 0.2, 0.4, 0.6, 0.8, 1}. Finally, as in all previous cases, the model that maximized the function (6) was chosen [29].

5.4. Classification Results

This section refers to application of classification algorithms mentioned in Section 5.3. For the sake of clarity and synthesis, the results are visualized and provided for the testing dataset only. However, in the appendix section the detailed results for each algorithm and for three feature sets are presented (Appendix B).

Additionally, in Appendix C the final set of independent variables used in classification models and for each dependent variable was provided.

As far as summary results are concerned, Figure 5 shows the accuracy achieved by the algorithms—KNN, NNET and SVM with break down into three feature selection techniques—All variables, 8 GCA, 8 Boruta. From the left to the right are the results for family type, number of bedrooms, employment type, floor area, house type, number of appliances, householder age and house age. The whiskers represent standard deviations.

It can be observed that the methods achieve approx. 90% accuracy for classification of appliances and age of the house, regardless of the classification algorithm. Family type is classified with nearly 75% accuracy. On the other hand, the most difficult characteristic to be discovered by algorithms is number of bedrooms, with the accuracy reaching only 50%.

In terms of different approaches for features selection, it was observed that proposed GCA algorithm (8-GCA), used for clustering variables and selecting only two representatives of the clusters, worked well and can be considered as a technique for feature selection. Broader set of all variables was relevant for classification of floor area only.

The next figure, Figure 6, illustrates the AUC values for the classifiers. The range of AUC values between analyzed households' characteristics vary from 0.52 (for age of the house and using KNN) to 0.82 (for family type, regardless classification algorithm). Overall, all variables are necessary to result in high AUC only for classification of main inhabitant's age and floor area. For other characteristics, using 8 variables, either GCA or Boruta, resulted in equally good classification measured by AUC.

In general, the results indicate that the choice of a classification model should depend on the specific target application. In the experiment it was observed that SVM and NNET stand out as the classifiers that allow to achieve the best performance. However, the results may vary taking into account variable selection mechanism.

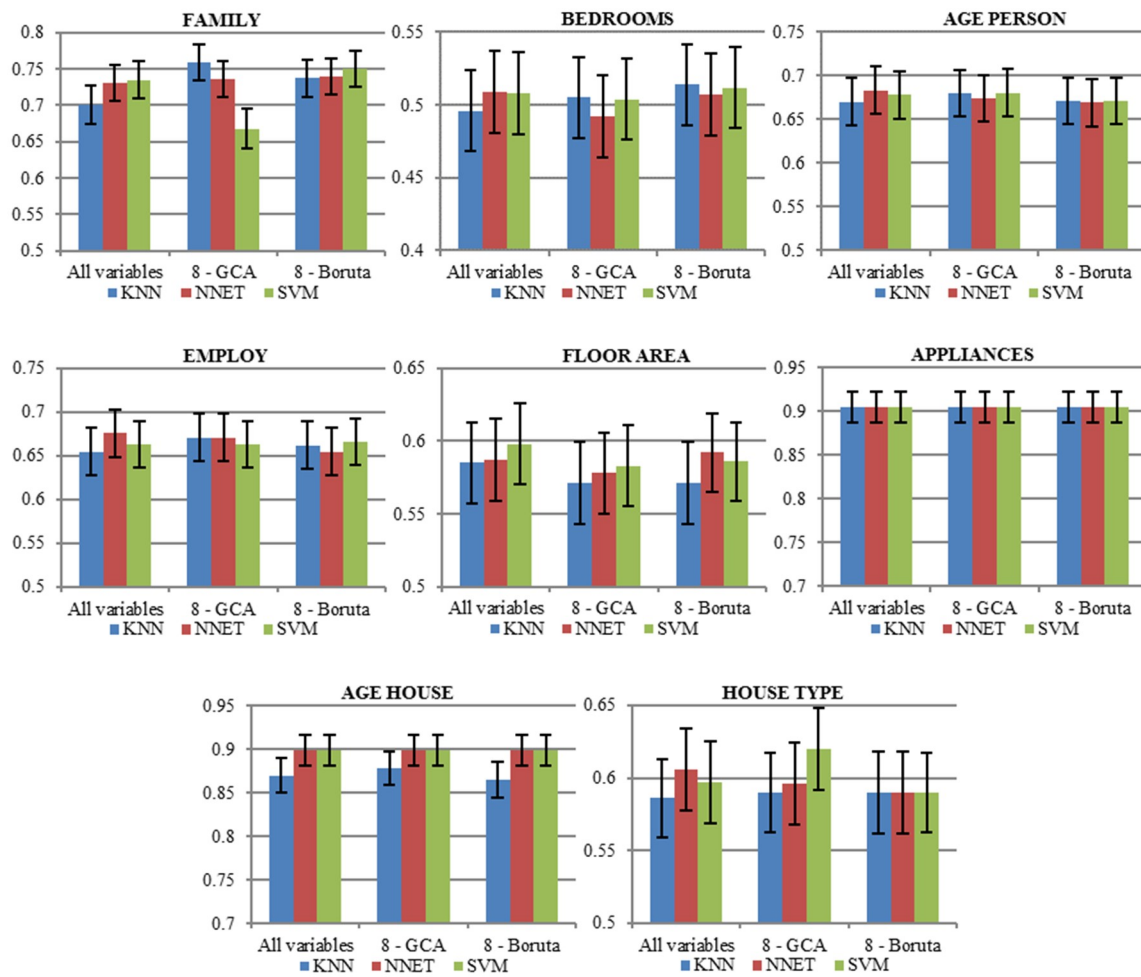


Figure 5. Classification accuracy.

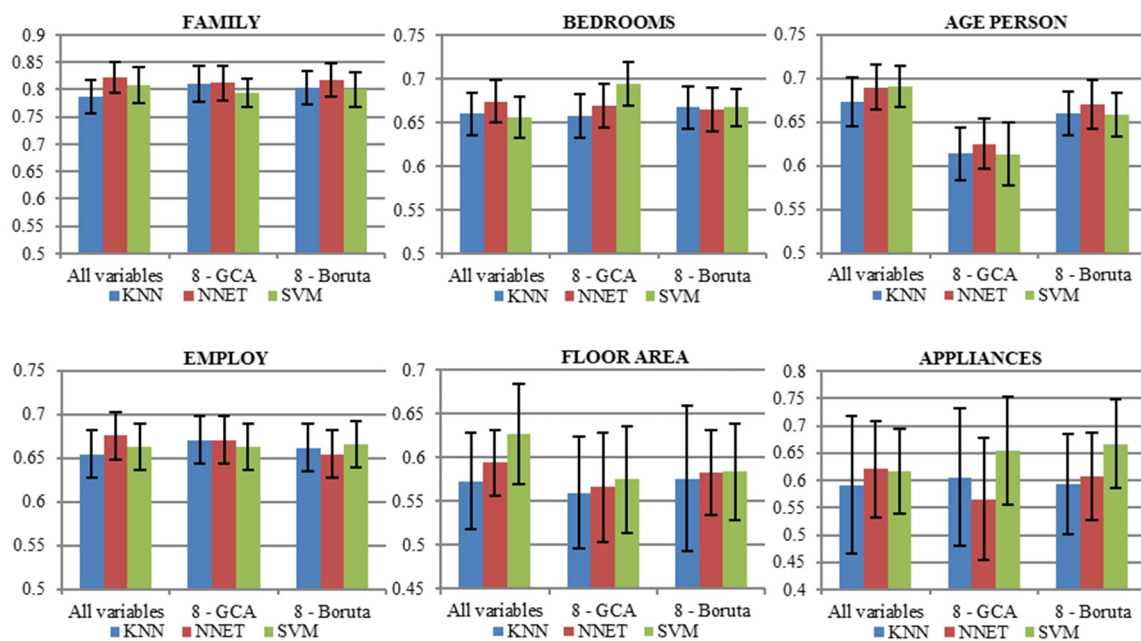


Figure 6. Cont.

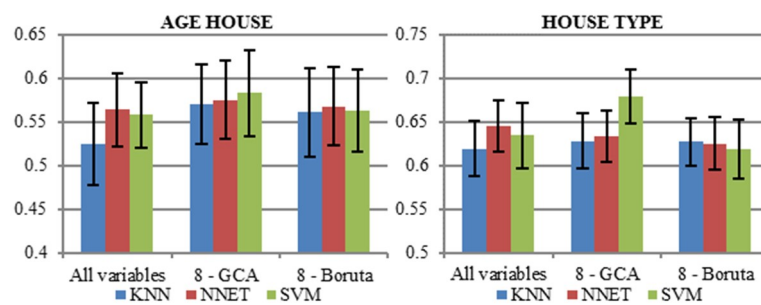


Figure 6. Area Under Receiver Operating Curve (AUC) values.

6. Conclusions

The approach presented in this paper shows that classification of households' socio-demographic and dwelling characteristics based on the electricity consumption is feasible and gives the opportunity to derive additional knowledge about the customers.

In practice, such knowledge can motivate electricity providers to offer new and more customer-oriented energy services. With growing liberalization of the energy market, premium and non-standard services may represent a competitive advantage to both existing customers and new ones.

The experimental results reported in Section 5 show that selected classification algorithms can reveal household characteristics from electricity consumption data with fair accuracy. In general, the choice of a particular classifier should depend on the specific target application. In the experiment, it was observed that SVM and NNET delivered equally good performance, however the results varied depending on the variable selection procedure. For six out of eight household characteristics, using only eight variables, either GCA or Boruta resulted in a satisfactory level of accuracy.

The GCA proposed in this article allowed for quickly grasping general trends in data, and then to cluster the attributes, taking into account historical electricity usage. It is worth underlining that the method was competitive with the Boruta algorithm, having its roots in random forest algorithms. The results obtained by grade analysis might be the basis not only for feature selection but also for the customers' segmentation.

Since the results are promising, we aim, as an extension to this research, to focus on a broader set of variables including external factors like weather information (including humidity, temperature, sunrises and sunsets) as well as holidays and observances (including school holidays). The other direction for future research may involve application of selected segmentation algorithms to extract homogeneous groups of customers and to look for specific socio-demographic characteristics within the clusters.

Author Contributions: K.G. prepared the simulation and analysis and wrote the 2nd, 5th and 6th section of the manuscript; M.S. wrote the 2nd section of the manuscript; T.Z. coordinated the main theme of the research and wrote the 1st, 3rd, 4th and 6th section of the manuscript. All authors have read and approved the final manuscript.

Funding: This study was cofounded by the Polish National Science Centre (NCN), Grant No. 2016/21/N/ST8/02435.

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

Appendix A

Table A1. AUC values for variables grouped into four clusters.

Cluster: 1								
Variable	Family	Bedrooms	Age_Person	Employ	House_Type	Age_House	Appliances	Floor_Area
r_var_wd_we	0.467	0.495	0.495	0.482	0.508	0.488	0.468	0.498
number_zeros	0.513	0.506	0.506	0.482	0.508	0.503	0.49	0.5
r_morning_noon_no_min	0.566	0.539	0.539	0.509	0.485	0.517	0.538	0.555
r_wd_morning_noon	0.568	0.52	0.52	0.608	0.487	0.523	0.542	0.546
r_wd_evening_noon	0.49	0.499	0.499	0.595	0.505	0.546	0.526	0.521
r_evening_noon_no_min	0.525	0.51	0.51	0.638	0.493	0.54	0.513	0.519
r_we_morning_noon	0.61	0.518	0.518	0.644	0.549	0.517	0.495	0.533
width_peaks	0.514	0.538	0.538	0.521	0.577	0.47	0.419	0.514
r_max_wd_we	0.53	0.508	0.508	0.513	0.523	0.501	0.491	0.522
r_morning_noon	0.589	0.52	0.52	0.494	0.527	0.521	0.538	0.552
const_time	0.547	0.542	0.542	0.562	0.511	0.541	0.557	0.537
r_evening_noon	0.496	0.501	0.501	0.53	0.508	0.509	0.53	0.52
r_min_mean	0.513	0.553	0.553	0.633	0.528	0.487	0.581	0.537
r_we_evening_noon	0.515	0.495	0.495	0.484	0.506	0.512	0.518	0.487
r_mean_max_no_min	0.554	0.579	0.579	0.566	0.572	0.546	0.562	0.536
r_wd_night_day	0.584	0.519	0.519	0.55	0.557	0.521	0.631	0.514
value_min_guess	0.481	0.533	0.533	0.502	0.579	0.516	0.577	0.509
first_above_base	0.535	0.515	0.515	0.531	0.524	0.568	0.552	0.522
r_night_day	0.588	0.52	0.52	0.499	0.559	0.519	0.566	0.521
dist_big_v	0.544	0.526	0.526	0.502	0.518	0.512	0.54	0.522
r_noon_wd_we	0.512	0.502	0.502	0.514	0.495	0.496	0.537	0.536
r_we_night_day	0.574	0.518	0.518	0.565	0.55	0.515	0.569	0.542
r_afternoon_wd_we	0.508	0.495	0.495	0.501	0.502	0.505	0.537	0.485
time_above_base2	0.51	0.523	0.523	0.557	0.551	0.538	0.567	0.547
number_big_peaks	0.618	0.541	0.541	0.504	0.51	0.533	0.481	0.512
r_evening_wd_we	0.478	0.508	0.508	0.5	0.511	0.5	0.506	0.537
r_night_wd_we	0.504	0.5	0.5	0.508	0.533	0.503	0.488	0.554

Table A1. Cont.

Cluster: 2								
Variable	Family	Bedrooms	Age_Person	Employ	House_Type	Age_House	Appliances	Floor_Area
number_small_peaks	0.569	0.546	0.546	0.522	0.516	0.527	0.526	0.508
s_num_peaks	0.569	0.546	0.546	0.522	0.516	0.527	0.526	0.508
r_min_wd_we	0.508	0.51	0.51	0.511	0.552	0.512	0.503	0.537
r_morning_wd_we	0.567	0.483	0.483	0.579	0.544	0.509	0.467	0.499
t_daily_max	0.502	0.506	0.506	0.494	0.507	0.523	0.516	0.55
s_cor_we	0.525	0.519	0.519	0.533	0.506	0.523	0.54	0.515
s_cor_wd_we	0.548	0.541	0.541	0.562	0.511	0.503	0.544	0.542
percent_above_base	0.63	0.585	0.585	0.499	0.542	0.511	0.571	0.547
s_cor_wd	0.549	0.545	0.545	0.55	0.522	0.521	0.584	0.517
t_above_mean	0.569	0.556	0.556	0.558	0.523	0.541	0.542	0.516
ts_acf_mean3h	0.547	0.561	0.561	0.518	0.561	0.52	0.565	0.527
t_daily_min	0.544	0.549	0.549	0.53	0.535	0.507	0.555	0.548
ts_acf_mean3h_weekday	0.589	0.565	0.565	0.497	0.52	0.51	0.574	0.526
Cluster: 3								
Variable	Family	Bedrooms	Age_Person	Employ	House_Type	Age_House	Appliances	Floor_Area
r_mean_max	0.57	0.59	0.59	0.527	0.587	0.528	0.598	0.539
t_above_base	0.718	0.584	0.584	0.519	0.502	0.516	0.552	0.528
r_day_night_no_min	0.582	0.529	0.529	0.553	0.512	0.539	0.568	0.506
wide_peaks	0.457	0.541	0.541	0.497	0.577	0.508	0.588	0.515
c_max	0.758	0.635	0.635	0.636	0.545	0.546	0.634	0.557
c_wd_max	0.757	0.63	0.63	0.642	0.538	0.557	0.629	0.544
c_we_max	0.746	0.634	0.634	0.623	0.546	0.538	0.616	0.541
s_max_avg	0.783	0.647	0.647	0.65	0.551	0.549	0.652	0.543
value_above_base	0.779	0.65	0.65	0.609	0.55	0.559	0.632	0.54
c_sm_max	0.766	0.646	0.646	0.639	0.564	0.551	0.653	0.551
c_min	0.658	0.641	0.641	0.574	0.578	0.52	0.635	0.533
sm_variety	0.731	0.631	0.631	0.584	0.575	0.517	0.612	0.56
Cluster: 4								
Variable	Family	Bedrooms	Age_Person	Employ	House_Type	Age_House	Appliances	Floor_Area
c_wd_min	0.661	0.647	0.647	0.572	0.585	0.519	0.661	0.514
c_we_evening	0.737	0.649	0.649	0.63	0.573	0.541	0.619	0.554

Table A1. Cont.

c_evening	0.764	0.661	0.661	0.645	0.583	0.541	0.645	0.557
c_wd_evening	0.765	0.657	0.657	0.645	0.581	0.541	0.641	0.553
c_evening_no_min	0.761	0.65	0.65	0.645	0.571	0.542	0.624	0.548
b_day_diff	0.744	0.64	0.64	0.61	0.566	0.545	0.647	0.552
c_wd_night	0.661	0.633	0.633	0.576	0.611	0.505	0.654	0.54
c_afternoon	0.654	0.633	0.633	0.578	0.615	0.5	0.652	0.522
c_night	0.654	0.633	0.633	0.578	0.615	0.5	0.652	0.522
b_day_weak	0.714	0.631	0.631	0.605	0.565	0.543	0.63	0.557
c_wd_morning	0.684	0.628	0.628	0.599	0.587	0.498	0.622	0.55
c_morning	0.673	0.629	0.629	0.585	0.601	0.498	0.624	0.55
c_weekend	0.742	0.657	0.657	0.603	0.597	0.489	0.641	0.528
c_we_morning	0.627	0.617	0.617	0.541	0.615	0.504	0.612	0.535
c_we_min	0.648	0.64	0.64	0.572	0.606	0.49	0.629	0.549
c_we_night	0.639	0.628	0.628	0.579	0.606	0.494	0.638	0.494
c_we_afternoon	0.749	0.635	0.635	0.603	0.559	0.535	0.611	0.529
s_min_avg	0.665	0.655	0.655	0.575	0.617	0.492	0.659	0.526
c_week	0.761	0.669	0.669	0.603	0.599	0.496	0.67	0.555
c_night_no_min	0.638	0.606	0.606	0.566	0.594	0.512	0.623	0.521
s_diff	0.765	0.668	0.668	0.6	0.594	0.499	0.668	0.554
c_weekday	0.765	0.668	0.668	0.6	0.594	0.499	0.668	0.554
bg_variety	0.806	0.657	0.657	0.603	0.561	0.525	0.631	0.546
n_d_diff	0.636	0.603	0.603	0.567	0.59	0.507	0.624	0.512
c_morning_no_min	0.676	0.612	0.612	0.581	0.578	0.501	0.59	0.548
s_q1	0.715	0.665	0.665	0.573	0.612	0.498	0.657	0.515
c_we_noon	0.71	0.625	0.625	0.557	0.571	0.517	0.61	0.541
c_wd_afternoon	0.766	0.646	0.646	0.574	0.562	0.541	0.648	0.553
s_q3	0.743	0.662	0.662	0.588	0.6	0.494	0.66	0.554
c_noon	0.73	0.644	0.644	0.536	0.581	0.505	0.63	0.538
s_q2	0.759	0.662	0.662	0.565	0.589	0.486	0.651	0.532
c_wd_noon	0.717	0.636	0.636	0.518	0.573	0.5	0.622	0.532
c_noon_no_min	0.715	0.624	0.624	0.513	0.557	0.498	0.604	0.528
ts_stl_varRem	0.748	0.632	0.632	0.646	0.556	0.494	0.616	0.541
s_var_we	0.735	0.635	0.635	0.623	0.553	0.491	0.61	0.535
t_above_1kw	0.74	0.655	0.655	0.605	0.595	0.499	0.668	0.549
s_variance	0.75	0.641	0.641	0.635	0.563	0.499	0.645	0.545
s_var_wd	0.752	0.637	0.637	0.634	0.559	0.504	0.644	0.545
t_above_2kw	0.745	0.651	0.651	0.632	0.573	0.496	0.657	0.53

Appendix B

Table A2. Classification results for each dependent variable.

		Training/Validation Sample		
		AC	AUC	
Model for family	All variables	ANN (iteration = 17, neurons = 9)	0.766 (±0.015)/0.731 (±0.025)	0.854 (±0.016)/0.822 (±0.029)
		KNN (k = 260)	0.722 (±0.017)/0.701 (±0.026)	0.806 (±0.019)/0.787 (±0.031)
		SVM (kernel = polynomial, degree = 1, C = 0.3, gamma = 0.1)	0.778 (±0.016)/0.735 (±0.026)	0.825 (±0.017)/0.808 (±0.033)
	8 best variables based on AUC and GCA	ANN (iteration = 28, neurons = 7)	0.755 (±0.016)/0.736 (±0.025)	0.834 (±0.019)/0.812 (±0.031)
		KNN (k = 280)	0.776 (±0.016)/0.759 (±0.025)	0.831 (±0.019)/0.811 (±0.033)
		SVM (kernel = sigmoid, degree = 1, C = 0.9, gamma = 0.1)	0.673 (±0.018)/0.668 (±0.027)	0.798 (±0.019)/0.794 (±0.025)
	8 best variables based on Boruta	ANN (iteration = 28, neurons = 14)	0.769 (±0.016)/0.740 (±0.025)	0.847 (±0.016)/0.817 (±0.031)
		KNN (k = 160)	0.754 (±0.016)/0.737 (±0.025)	0.833 (±0.017)/0.803 (±0.031)
		SVM (kernel = sigmoid, degree = 1, C = 0.3, gamma = 0.1)	0.761 (±0.016)/0.750 (±0.025)	0.826 (±0.018)/0.800 (±0.032)
		Training/Validation Sample		
		AC	AUC	
Model for bedrooms	All variables	ANN (iteration = 2217, neurons = 4)	0.493 (±0.018)/0.509 (±0.028)	0.700 (±0.013)/0.674 (±0.024)
		KNN (k = 250)	0.494 (±0.019)/0.496 (±0.028)	0.668 (±0.012)/0.660 (±0.024)
		SVM (kernel = sigmoid, degree = 1, C = 0.1, gamma = 0.1)	0.494 (±0.018)/0.508 (±0.028)	0.674 (±0.014)/0.656 (±0.023)
	8 best variables based on AUC and GCA	ANN (iteration = 19, neurons = 6)	0.482 (±0.019)/0.492 (±0.028)	0.683 (±0.013)/0.669 (±0.025)
		KNN (k = 300)	0.491 (±0.018)/0.505 (±0.028)	0.685 (±0.013)/0.657 (±0.025)
		SVM (kernel = polynomial, degree = 1, C = 0.9, gamma = 0.9)	0.490 (±0.018)/0.504 (±0.028)	0.667 (±0.012)/0.664 (±0.025)
	8 best variables based on Boruta	ANN (iteration = 26, neurons = 9)	0.494 (±0.018)/0.507 (±0.028)	0.683 (±0.013)/0.665 (±0.025)
		KNN (k = 300)	0.492 (±0.019)/0.514 (±0.028)	0.687 (±0.013)/0.667 (±0.024)
		SVM (kernel = polynomial, degree = 3, C = 0.7, gamma = 0.7)	0.486 (±0.018)/0.512 (±0.028)	0.679 (±0.011)/0.667 (±0.021)
		Training/Validation Sample		
		AC	AUC	
Model for age_person	All variables	ANN (iteration = 16, neurons = 3)	0.678 (±0.017)/0.683 (±0.027)	0.708 (±0.016)/0.690 (±0.026)
		KNN (k = 90)	0.670 (±0.017)/0.670 (±0.027)	0.713 (±0.017)/0.673 (±0.028)
		SVM (kernel = polynomial, degree = 1, C = 0.1, gamma = 0.9)	0.674 (±0.017)/0.678 (±0.027)	0.726 (±0.013)/0.691 (±0.023)
	8 best variables based on AUC and GCA	ANN (iteration = 27, neurons = 4)	0.666 (±0.017)/0.674 (±0.027)	0.666 (±0.019)/0.625 (±0.029)
		KNN (k = 260)	0.665 (±0.018)/0.680 (±0.026)	0.663 (±0.021)/0.614 (±0.030)
		SVM (kernel = polynomial, degree = 2, C = 0.93, gamma = 0.1)	0.666 (±0.018)/0.680 (±0.027)	0.639 (±0.023)/0.613 (±0.036)
	8 best variables based on Boruta	ANN (iteration = 23, neurons = 9)	0.666 (±0.017)/0.669 (±0.027)	0.699 (±0.017)/0.670 (±0.028)
		KNN (k = 300)	0.665 (±0.017)/0.671 (±0.027)	0.698 (±0.019)/0.660 (±0.025)
		SVM (kernel = polynomial, degree = 3, C = 0.5, gamma = 0.1)	0.662 (±0.017)/0.671 (±0.027)	0.659 (±0.016)/0.658 (±0.025)

Table A2. Cont.

			Training/Validation Sample	
			AC	AUC
Model for employ	All variables	ANN (iteration = 13, neurons = 8)	0.696 (±0.017)/0.676 (±0.027)	0.754 (±0.017)/0.732 (±0.027)
		KNN (k = 140)	0.674 (±0.017)/0.655 (±0.027)	0.734 (±0.017)/0.711 (±0.025)
		SVM (kernel = linear, degree = 1, C = 1, gamma = 1)	0.703 (±0.017)/0.663 (±0.027)	0.758 (±0.018)/0.728 (±0.034)
	8 best variables based on AUC and GCA	ANN (iteration = 6, neurons = 11)	0.678 (±0.018)/0.671 (±0.027)	0.713 (±0.018)/0.712 (±0.030)
		KNN (k = 260)	0.682 (±0.018)/0.671 (±0.027)	0.734 (±0.018)/0.713 (±0.027)
		SVM (kernel = polynomial, degree = 3, C = 0.1, gamma = 0.5)	0.672 (±0.017)/0.663 (±0.027)	0.726 (±0.020)/0.713 (±0.031)
	8 best variables based on Boruta	ANN (iteration = 5, neurons = 13)	0.652 (±0.018)/0.655 (±0.027)	0.704 (±0.018)/0.702 (±0.030)
		KNN (k = 300)	0.677 (±0.017)/0.662 (±0.027)	0.723 (±0.019)/0.703 (±0.031)
		SVM (kernel = sigmoid, degree = 1, C = 0.9, gamma = 0.9)	0.678 (±0.017)/0.666 (±0.027)	0.718 (±0.021)/0.704 (±0.030)
			Training/Validation Sample	
			AC	AUC
Model for floor_area	All variables	ANN (iteration = 17, neurons = 9)	0.622 (±0.018)/0.587 (±0.028)	0.604 (±0.033)/0.594 (±0.038)
		KNN (k = 260)	0.598 (±0.018)/0.585 (±0.028)	0.681 (±0.031)/0.573 (±0.055)
		SVM (kernel = sigmoid, degree = 1, C = 0.7, gamma = 0.1)	0.609 (±0.018)/0.598 (±0.028)	0.587 (±0.053)/0.627 (±0.057)
	8 best variables based on AUC and GCA	ANN (iteration = 28, neurons = 7)	0.613 (±0.018)/0.578 (±0.028)	0.580 (±0.033)/0.566 (±0.063)
		KNN (k = 280)	0.585 (±0.018)/0.571 (±0.028)	0.692 (±0.026)/0.560 (±0.064)
		SVM (kernel = polynomial, degree = 1, C = 0.9, gamma = 0.1)	0.599 (±0.018)/0.583 (±0.028)	0.574 (±0.044)/0.575 (±0.061)
	8 best variables based on Boruta	ANN (iteration = 28, neurons = 14)	0.603 (±0.017)/0.592 (±0.027)	0.583 (±0.034)/0.583 (±0.049)
		KNN (k = 160)	0.591 (±0.018)/0.571 (±0.028)	0.625 (±0.032)/0.576 (±0.083)
		SVM (kernel = polynomial, degree = 1, C = 0.9, gamma = 0.5)	0.593 (±0.018)/0.586 (±0.027)	0.579 (±0.032)/0.584 (±0.055)
			Training/Validation Sample	
			AC	AUC
Model for appliances	All variables	ANN (iteration = 19, neurons = 1)	0.908 (±0.011)/0.905 (±0.017)	0.686 (±0.048)/0.566 (±0.088)
		KNN (k = 40)	0.908 (±0.011)/0.905 (±0.017)	0.784 (±0.023)/0.591 (±0.126)
		SVM (kernel = polynomial, degree = 1, C = 0.3, gamma = 0.9)	0.908 (±0.011)/0.905 (±0.017)	0.596 (±0.060)/0.616 (±0.078)
	8 best variables based on AUC and GCA	ANN (iteration = 12, neurons = 2)	0.908 (±0.011)/0.905 (±0.017)	0.605 (±0.055)/0.566 (±0.111)
		KNN (k = 70)	0.908 (±0.011)/0.905 (±0.017)	0.766 (±0.022)/0.606 (±0.125)
		SVM (kernel = polynomial, degree = 1, C = 0.5, gamma = 0.3)	0.908 (±0.011)/0.905 (±0.017)	0.659 (±0.049)/0.654 (±0.099)
	8 best variables based on Boruta	ANN (iteration = 11, neurons = 7)	0.908 (±0.011)/0.905 (±0.017)	0.650 (±0.056)/0.607 (±0.080)
		KNN (k = 120)	0.908 (±0.011)/0.905 (±0.017)	0.740 (±0.024)/0.594 (±0.092)
		SVM (kernel = radial, degree = 1, C = 1, gamma = 0.9)	0.908 (±0.011)/0.905 (±0.017)	0.666 (±0.068)/0.667 (±0.041)

Table A2. Cont.

		Training/Validation Sample		
		AC	AUC	
Model for age_house	All variables	ANN (iteration = 17, neurons = 9)	0.900 (±0.011)/0.899 (±0.018)	0.563 (±0.029)/0.564 (±0.042)
		KNN (k = 260)	0.876 (±0.012)/0.870 (±0.020)	0.616 (±0.035)/0.525 (±0.047)
		SVM (kernel = sigmoid, degree = 1, C = 0.3, gamma = 0.5)	0.900 (±0.011)/0.899 (±0.018)	0.548 (±0.033)/0.558 (±0.038)
	8 best variables based on AUC and GCA	ANN (iteration = 28, neurons = 15)	0.900 (±0.011)/0.899 (±0.018)	0.593 (±0.032)/0.575 (±0.045)
		KNN (k = 280)	0.871 (±0.013)/0.878 (±0.019)	0.625 (±0.030)/0.570 (±0.046)
		SVM (kernel = polynomial, degree = 3, C = 0.3, gamma = 0.5)	0.900 (±0.011)/0.899 (±0.018)	0.586 (±0.029)/0.583 (±0.049)
	8 best variables based on Boruta	ANN (iteration = 28, neurons = 2)	0.900 (±0.011)/0.899 (±0.018)	0.581 (±0.033)/0.568 (±0.045)
		KNN (k = 160)	0.873 (±0.013)/0.865 (±0.021)	0.606 (±0.029)/0.561 (±0.051)
		SVM (kernel = polynomial, degree = 1, C = 0.1, gamma = 0.1)	0.900 (±0.011)/0.899 (±0.018)	0.558 (±0.028)/0.563 (±0.047)
		Training/Validation Sample		
		AC	AUC	
Model for house_type	All variables	ANN (iteration = 10, neurons = 13)	0.611 (±0.017)/0.606 (±0.028)	0.650 (±0.020)/0.616 (±0.029)
		KNN (k = 300)	0.598 (±0.018)/0.559 (±0.027)	0.626 (±0.020)/0.587 (±0.032)
		SVM (kernel = sigmoid, degree = 1, C = 0.1, gamma = 0.5)	0.590 (±0.018)/0.597 (±0.028)	0.606 (±0.020)/0.596 (±0.038)
	8 best variables based on AUC and GCA	ANN (iteration = 13, neurons = 7)	0.600 (±0.018)/0.596 (±0.028)	0.632 (±0.019)/0.604 (±0.029)
		KNN (k = 210)	0.602 (±0.018)/0.590 (±0.027)	0.628 (±0.023)/0.597 (±0.031)
		SVM (kernel = polynomial, degree = 3, C = 0.1, gamma = 0.5)	0.615 (±0.018)/0.620 (±0.028)	0.679 (±0.025)/0.648 (±0.031)
	8 best variables based on Boruta	ANN (iteration = 25, neurons = 2)	0.603 (±0.018)/0.590 (±0.028)	0.628 (±0.021)/0.595 (±0.030)
		KNN (k = 240)	0.602 (±0.018)/0.590 (±0.028)	0.627 (±0.019)/0.600 (±0.027)
		SVM (kernel = sigmoid, degree = 1, C = 0.5, gamma = 0.7)	0.599 (±0.018)/0.590 (±0.027)	0.619 (±0.021)/0.585 (±0.034)

Appendix C

Table A3. Final set of independent variables for classification models for each dependent variable.

Family			Bedrooms		
Variable	AUC	Cluster	Variable	AUC	Cluster
number_big_peaks	0.618	1	r_mean_max_no_min	0.578	1
r_we_morning_noon	0.609	1	r_min_mean	0.553	1
percent_above_base	0.630	2	percent_above_base	0.584	2
ts_acf_mean3h_weekday	0.58	2	ts_acf_mean3h_weekday	0.565	2
s_max_avg	0.782	3	value_above_base	0.649	3
c_min	0.658	3	c_min	0.640	3
bg_variety	0.806	4	c_week	0.668	4
c_wd_min	0.660	4	c_wd_min	0.646	4
Age_Person			Employ		
Variable	AUC	Cluster	Variable	AUC	Cluster
r_mean_max_no_min	0.578	1	r_evening_noon_no_min	0.643	1
r_min_mean	0.553	1	r_wd_morning_noon	0.595	1
percent_above_base	0.584	2	r_morning_wd_we	0.578	2
ts_acf_mean3h_weekday	0.565	2	s_cor_wd_we	0.562	2
value_above_base	0.649	3	s_max_avg	0.649	3
c_min	0.640	3	sm_variety	0.583	3
c_week	0.668	4	ts_stl_varRem	0.646	4
c_wd_min	0.646	4	c_wd_morning	0.598	4
House_Type			Age_House		
Variable	AUC	Cluster	Variable	AUC	Cluster
value_min_guess	0.578	1	first_above_base	0.567	1
width_peaks	0.577	1	r_wd_evening_noon	0.546	1
ts_acf_mean3h	0.561	2	t_above_mean	0.540	2
r_min_wd_we	0.552	2	number_small_peaks	0.527	2
r_mean_max	0.586	3	value_above_base	0.559	3
c_min	0.578	3	r_day_night_no_min	0.539	3
s_min_avg	0.616	4	b_day_diff	0.544	4
s_q3	0.600	4	c_we_evening	0.540	4
Appliances			Floor_Area		
Variable	AUC	Cluster	Variable	AUC	Cluster
r_wd_night_day	0.631	1	r_morning_noon_no_min	0.55	1
r_min_mean	0.580	1	time_above_base2	0.54	1
s_cor_wd	0.584	2	t_daily_max	0.549	2
percent_above_base	0.571	2	t_daily_min	0.548	2
c_sm_max	0.653	3	sm_variety	0.560	3
c_min	0.634	3	c_max	0.556	3
c_week	0.669	4	c_evening	0.557	4
c_wd_min	0.661	4	c_wd_morning	0.549	4

References

1. Chicco, G. Overview and performance assessment of the clustering methods for electrical load pattern grouping. *Energy* **2012**, *421*, 68–80. [[CrossRef](#)]
2. Chicco, G.; Napoli, R.; Piglion, F.; Postolache, P.; Scutariu, M.; Toader, C. Load pattern-based classification of electricity customers. *IEEE Trans. Power Syst.* **2004**, *192*, 1232–1239. [[CrossRef](#)]
3. Gajowniczek, K.; Ząbkowski, T. Short term electricity forecasting based on user behavior using individual smart meter data. *Intell. Fuzzy Syst.* **2015**, *30*, 223–234. [[CrossRef](#)]

4. Haben, S.; Singleton, C.; Grindrod, P. Analysis and clustering of residential customers energy behavioral demand using smart meter data. *IEEE Trans. Smart Grid* **2016**, *7*, 136–144. [[CrossRef](#)]
5. Gajowniczek, K.; Ząbkowski, T. Electricity forecasting on the individual household level enhanced based on activity patterns. *PLoS ONE* **2017**, *12*, e0174098. [[CrossRef](#)] [[PubMed](#)]
6. Sial, A.; Singh, A.; Mahanti, A.; Gong, M. Heuristics-Based Detection of Abnormal Energy Consumption. In *International Conference on Smart Grid Inspired Future Technologies*; Chong, P., Seet, B.C., Chai, M., Eds.; Springer: Cham, Switzerland, 2018; pp. 21–31.
7. Batra, N.; Singh, A.; Whitehouse, K. Creating a Detailed Energy Breakdown from just the Monthly Electricity Bill. In Proceedings of the 3rd International NILM Workshop, San Francisco, CA, USA, 14–15 May 2016.
8. Rashid, H.; Arjunan, P.; Singh, P.; Singh, A. Collect, compare, and score: A generic data-driven anomaly detection method for buildings. In Proceedings of the Seventh International Conference on Future Energy Systems Poster Sessions, Waterloo, ON, Canada, 21–24 June 2016.
9. Beckel, C.; Sadamori, L.; Santini, S. Automatic socio-economic classification of households using electricity consumption data. In Proceedings of the Fourth International Conference on Future Energy Systems, Waterloo, ON, Canada, 15 January 2013.
10. Hopf, K.; Sodenkamp, M.; Kozlovkiy, I.; Staake, T. Feature extraction and filtering for household classification based on smart electricity meter data. *Comput. Sci. Res. Dev.* **2016**, *31*, 141–148. [[CrossRef](#)]
11. Poortinga, W.; Steg, L.; Vlek, C.; Wiersma, G. Household preferences for energy-saving measures: A conjoint analysis. *J. Econ. Psychol.* **2003**, *24*, 49–64. [[CrossRef](#)]
12. Vassileva, I.; Campillo, J. Increasing energy efficiency in low-income households through targeting awareness and behavioral change. *Renew. Energy* **2014**, *67*, 59–63. [[CrossRef](#)]
13. Ehrhardt-Martinez, K. Changing habits, lifestyles and choices: The behaviours that drive feedback-induced energy savings. In Proceedings of the 2011 ECEEE Summer Study on Energy Efficiency in Buildings, Toulon, France, 1–6 June 2011.
14. Chicco, G.; Napoli, R.; Postolache, P.; Scutariu, M.; Toader, C. Customer characterization options for improving the tariff offer. *IEEE Trans. Power Syst.* **2003**, *18*, 381–387. [[CrossRef](#)]
15. Carroll, J.; Lyons, S.; Denny, E. Reducing household electricity demand through smart metering: The role of improved information about energy saving. *Energy Econ.* **2014**, *45*, 234–243. [[CrossRef](#)]
16. Anda, M.; Temmen, J. Smart metering for residential energy efficiency: The use of community based social marketing for behavioural change and smart grid introduction. *Renew. Energy* **2014**, *67*, 119–127. [[CrossRef](#)]
17. Hart, G.W. *Nonintrusive Appliance Load Monitoring*; IEEE: New York, NY, USA, 1992.
18. Zeifman, M.; Roth, K. Nonintrusive appliance load monitoring: Review and outlook. *IEEE Trans. Consum. Electron.* **2011**, *57*, 76–84. [[CrossRef](#)]
19. Zoha, A.; Gluhak, A.; Imran, M.A.; Rajasegarar, S. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors* **2012**, *12*, 16838–16866. [[CrossRef](#)] [[PubMed](#)]
20. Beckel, C.; Sadamori, L.; Staake, T.; Santini, S. Revealing household characteristics from smart meter data. *Energy* **2014**, *78*, 397–410. [[CrossRef](#)]
21. Szczesny, W. On the performance of a discriminant function. *J. Classif.* **1991**, *8*, 201–215. [[CrossRef](#)]
22. Kowalczyk, T.; Pleszczyńska, E.; Ruland, F. *Grade Models and Methods for Data Analysis: With Applications for the Analysis of Data Populations*; Springer: Berlin/Heidelberg, Germany, 2004; Volume 151.
23. Ciok, A.; Kowalczyk, T.; Pleszczyńska, E. How a new statistical infrastructure induced a new computing trend in data analysis. In *International Conference on Rough Sets and Current Trends in Computing*; Springer: Berlin/Heidelberg, Germany, 1998.
24. Szczesny, W. Grade correspondence analysis applied to contingency tables and questionnaire data. *Intell. Data Anal.* **2002**, *6*, 17–51.
25. Program for Grade Data Analysis. Available online: gradestat.ipipan.waw.pl (accessed on 9 June 2018).
26. Kurska, M.B.; Rudnicki, W.R. Feature selection with the Boruta package. *J. Stat. Softw.* **2010**, *36*, 1–13. [[CrossRef](#)]
27. Fawcett, T. An introduction to ROC analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874. [[CrossRef](#)]
28. Gajowniczek, K.; Ząbkowski, T. Simulation study on clustering approaches for short-term electricity forecasting. *Complex* **2018**, *2018*, 3683969. [[CrossRef](#)]
29. Gajowniczek, K.; Ząbkowski, T. Two-Stage Electricity Demand Modeling Using Machine Learning Algorithms. *Energies* **2017**, *10*, 1547. [[CrossRef](#)]

30. Nguyen, B.; Morell, C.; De Baets, B. Large-scale distance metric learning for k-nearest neighbours regression. *Neurocomputing* **2016**, *214*, 805–814. [[CrossRef](#)]
31. Davò, F.; Vespucci, M.T.; Gelmini, A.; Grisi, P.; Ronzio, D. Forecasting Italian electricity market prices using a Neural Network and a Support Vector Regression. In Proceedings of the 2016 AEIT International Annual Conference (AEIT), Capri, Italy, 5–7 October 2016.
32. Muandet, K.; Fukumizu, K.; Sriperumbudur, B.; Schölkopf, B. Kernel mean embedding of distributions: A review and beyond. *Found. Trends Mach. Learn.* **2017**, *10*, 1–141. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).