# Non-Intrusive Appliance Load Identification with the Ensemble of Classifiers

Piotr Bilski
Institute of Radioelectronics and Multimedia Technologies
Warsaw University of Technology
Warsaw, Poland
pbilski@ire.pw.edu.pl

Wiesław Winiecki
Institute of Radioelectronics and Multimedia Technologies
Warsaw University of Technology
Warsaw, Poland
W.Winiecki@ire.pw.edu.pl

***Abstract*—The paper presents the application of the classifier fusion to identify electrical appliances present in the household. The analysis is based on the processing of features extracted from the current signal, recorded by the dedicated data acquisition hardware, installed in the vicinity of the energy meter. The selected features are based on the medium frequency measurements. The proposed identification module uses three classifiers of the same type: decision trees, rules and random forest. Experimental results prove the effectiveness of combining various classifiers to the same task and show their advantages and drawbacks.**

***Keywords—NIALM; artificial intelligence; classification; medium frequency measurements***

## I. INTRODUCTION

The Non-Intrusive Appliance Load Monitoring (NIALM) [1] is currently the standard tool considered for the analysis of the power consumption in particular devices working in households. The complete systems are proposed to determine the energy profile for each type of appliance, identifying the most expensive ones. As only one measurement module (located in the vicinity of the meter) is used here, the main problem is the identification software, operating on the set of selected voltage and current features. Both issues are evaluated, by proposing combinations of computational algorithms and data vectors for processing. Multiple Artificial Intelligence (AI) approaches were implemented so far, including Artificial Neural Networks [2], Support Vector Machines [3], rules [4] or Hidden Markov Models [5]. Their strength lies in the ability to automatically train and classify events occurring in the power network. The most pressing issue is the maximization of the appliance identification accuracy, which is the basis for the further energy consumption analysis. To achieve this aim, more complex methods may be implemented, such as the deep learning or the ensemble of classifiers. Also, comparison of the particular methods' outcomes is required to select the best ones for the particular problem.

The paper presents the application of the classifiers' fusion in the NIALM approach to identify appliances working in the household. It is the method of implementing multiple decision making algorithms to combine their decisions in attempt to increase the identification accuracy compared to the separate classifiers. Two aspects are important in the presented approach: selection of features extracted from the current and voltage signals (acquired by the specialized hardware) and implementation and configuration of classification algorithms, responsible for identifying all appliances operating in the apartment. The implemented algorithms belong to the rule-based approach: decision trees, rules induction and random forest. The combination of this group of classifiers should be more effective in the task than using any individual method.

The paper's structure is as follows. The description of the implemented NIALM architecture is in Section 2. The vector of features (further called example) and details of its generation are presented in Section 3. The algorithms implemented in the ensemble are introduced in Section 4. Experimental results are presented in Section 5, while Section 6 contains conclusions and future prospects for the proposed solution.

## II. PROPOSED NIALM ARCHITECTURE

The system implemented in the presented research (Fig. 1) is designed to work on-line in the environment with multiple two-state appliances (working in the on-off states, such as the kettle or the lightbulb) and finite state machines (such as washing machines or dryers), working in parallel. It is able to detect changes in the devices configuration, leading to the modified energy consumption. This includes turning the devices on or off, or changing its state (like the washing program). The architecture consists of two parts. The first one is the DAQ hardware, located near the energy meter, typically measuring the overall energy consumption in every apartment. It is assumed that the accurate appliance identification is not possible using only the standard infrastructure currently installed in the household (such as presented in [6]). Therefore the specialized module is used to acquire current and voltage signals with the 2kHz sampling frequency. It is considered by the Authors as the medium range, in the middle between the single [7] and hundreds of Hertz (like in the EMI analysis [8]).

The second part of the architecture is the software module, processing the acquired signals. It is designed to perform the analysis on-line, detecting the change in the configuration of the operating appliances. Its operation is based on executing three subsequent stages (Fig. 2):

1. Detection of changes in the measured current. It is the main source of information about changes in the devices

configuration (further called the “event”). Its detection is performed twice a second, based on the analysis of the one thousand samples vector, for which the maximum value is calculated. This way it is possible to detect the change in every appliance assuming such events do not occur faster than once a half-second. The event is detected if the difference in two subsequent maximum values are above the detection threshold $\theta$. The operation is then performed on the envelope of the current signal (Fig. 3), created from the original current waveform by selecting the maximum values from vectors of samples. Every index value (from the x-axis) represents the half a second period. Selection of the $\theta$ value is the compromise between the ability to detect changes even in the energy efficient devices (such as lightbulbs) and the false alarms (suggesting that the change took place although in fact it did not). Because turning the device on is sometimes related with the transient state (high pitch in the current, quickly returning to the new level), the beginning of the steady state must be indicated (see circles in Fig. 3). Only then the operation of the feature extraction from samples may commence. The black circles are the values differing from the previous ones above the threshold. The red ones indicate time instants when the appliance state change is completed, triggering its identification phase. The event is described either by the single red circle (for the change without the transient state) or the sequence of black circles followed by the red one (for the configuration change with the transient state).

2. After the detection, when the current signal is already in the steady state, features used for the identification are calculated. They include current and voltage parameters extracted from the samples’ vector (described in Section 3).

3. The features are the input to the fusion of classifiers module, which make their decisions individually. After the voting (where responses of all classifiers are combined), the final decision $d$ is made.

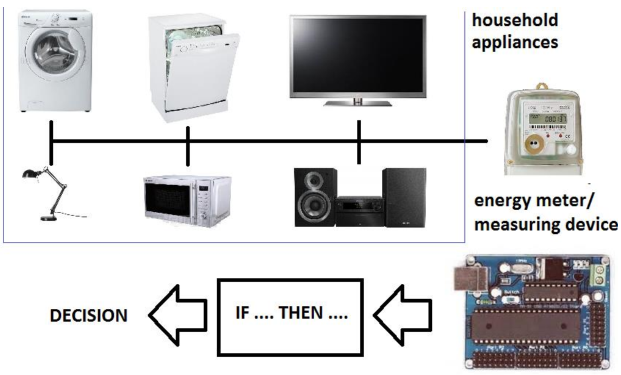


Fig. 1. The proposed NIALM architecture

All implemented classifiers are trained in the off-line mode on the previously prepared data sets $\boldsymbol{L}$ (1). The training examples $\boldsymbol{e}_1$, …, $\boldsymbol{e}_n$ were obtained for each appliance separately during their operation monitoring. The particular device was working in the isolated environment for the predefined amount of time (for instance one hour), in which the current and voltage were measured. All other appliances remained, so the subsequent processing referred only to this device. The samples’ vectors were used to calculate the features $\boldsymbol{s}_i$ representing the single appliance and supplement them with the appliance identifier $c_i$, forming the training examples $\boldsymbol{e}_i$. This way, multiple vectors could be extracted, in various time instants of the device operation period. In the presented research each of $k$ appliances was represented in the set $\boldsymbol{L}$ by 100 labeled examples (i.e. with known identifiers of appliances, which is required for the subsequent supervised learning of the AI approaches [9]).

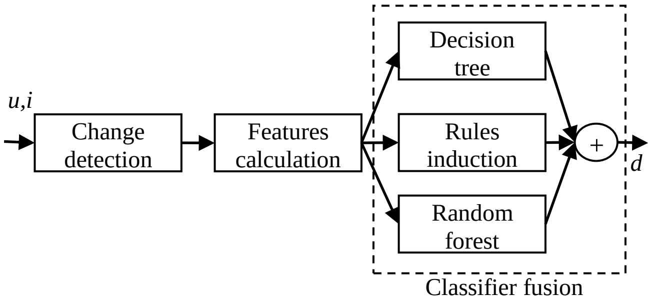


Fig. 2. The software part of the NIALM architecture

$$L = \begin{bmatrix} \boldsymbol{e}_1 \\ \boldsymbol{e}_2 \\ \vdots \\ \boldsymbol{e}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_1 & c_1 \\ \boldsymbol{s}_2 & c_1 \\ & \vdots \\ \boldsymbol{s}_n & c_k \end{bmatrix} = \begin{bmatrix} s_{11} & \cdots & s_{1m} & c_1 \\ s_{21} & & s_{2m} & c_1 \\ \vdots & \ddots & \vdots & \\ s_{n1} & \cdots & s_{nm} & c_k \end{bmatrix} \tag{1}$$

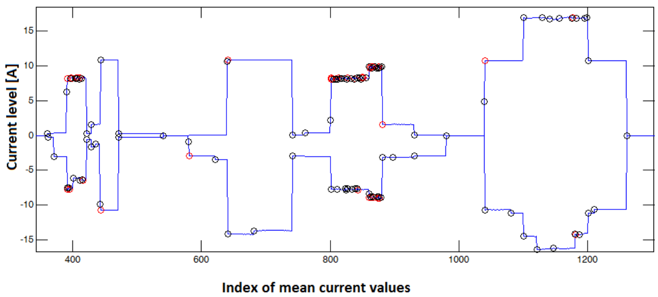


Fig. 3. The current envelope for the event detection

After detecting the event, all features are calculated for the extracted vector of samples in the steady state. The obtained values refer to the combination of currently operating appliances. To identify the device causing the event, the features must be obtained as the difference between the features calculated after the previous event (in the time instant $t_{j-1}$) and after the current event (in the time instant $t_j$) – see (2). Such an operation is required because in the tested environment multiple devices work simultaneously, therefore the identification in the particular moment is not aimed at determining all currently working appliances. Instead, only the change with respect to the previous configuration is detected and the appliance responsible for it identified.

$$s_i = s_i(t_j) - s_i(t_{j-1}) \quad (2)$$

## III. Data Vectors

Selection of the features required to identify the particular appliance is the important part of the employed machine learning scheme [10]. It is still unknown, which parameters of the current and voltage signals are the most useful to distinguish between various devices [11]. Therefore it is reasonable to collect their maximum set, as during the training of any implemented classifier only part of them will be considered (although each algorithm might select different subset). The calculations were performed for 1024 samples.

The following characteristics were extracted from the current and voltage patterns in the steady state:

- Amplitudes of the first sixteen harmonic components of the current (where the first component is at 50Hz)
- Phase shifts of the first sixteen harmonic components of the current
- Root Mean Square (RMS) of the current
- Mean value of the current
- Maximum current value
- DC component in the current
- Mean power
- Values of the active power in the first sixteen harmonics
- Values of the reactive power in the first sixteen harmonics

The set of overall 69 real-valued features was obtained for the training examples and is calculated after detecting the change in the current level.

## IV. Ensemble Construction

This section briefly introduces the implemented classification algorithms and their configuration for the ensemble. All the selected methods are rule-based. Knowledge they represent after the training is legible for the human, which facilitates understanding of the ensemble operation and possible modifications. They differ in the generalization ability (i.e. the possibility to correctly identify examples not present during the training). The odd number of algorithms selected for the fusion minimizes the threat of the decision uncertainty when outcomes of the subsequent modules are equally distributed among different appliance identifiers [12].

The implemented algorithms are trained on the same set ***L***, but rules constructed in the process may be different for each algorithm. Although there are multiple approaches to combine the partial identifications into the final decision [13], in the presented research the simplest case was considered, where each classifier has equal vote and the decision is made based on the majority. Below each method is presented, including the adjusted parameters to maximize the identification accuracy.

All three methods were implemented in the Matlab environment by the Authors. This allowed for controlling the code and adjusting the parameters of every approach. Modifications of the original approaches are briefly described in the following subsections.

### *A. Decision tree (DT)*

This is the memory-efficient method [14], presenting the tree-like structure, starting from the highest node (root), connected with nodes of lower levels and ending with the terminal nodes (leaves). All nodes except the latter, store tests that compare the value of the selected feature with the threshold. The results of the test leads to the node on the lower level, until the leaf is reached. Selection of the test to the node during the tree construction is the key element of training. The minimum entropy criterion was used for this purpose. The parameter introduced by the Authors of the DT was the method of selecting feature for the test when multiple features have identical minimum entropy.

### *B. Rules Induction (RI)*

The algorithm implemented by the Authors is the modified version of the classic AQ algorithm [15], adjusted to process the continuous data (for details see [16]). The generated rules are not connected to one another, which allows for firing multiple rules at the same time. Each rule has the IF … THEN form, with the premises (conditions to meet) and the conclusion (identification of the appliance) parts. Their generation consists in the iterative generation of multiple premises with the same conclusion, allowing to cover the same example. Among them, the best one in the sense of generalization is selected and all examples covered by it are eliminated from the set ***L***. The process is concluded when there are no more examples to cover. The parameter of the method is the number of candidates for the rule for each example.

### *C. Random forest (RF)*

This is the newest approach to the rule-based systems, using multiple DT as parts of the classifier [17]. Each tree makes the individual decision based on the input vector of features and the final decision is made based on voting, where the majority of trees pointing at the identical appliance decides about the overall outcome. Therefore the RF is the fusion of trees. Contrary to the DT presented in section 4A, this time each tree is built considering not only the best attributes for the node, but also the worse ones. This way the trained structure is efficient from the statistical point of view and should have high generalization abilities. The adjusted parameters of the RT include the number of constructed trees and the set of attributes' values used for constructing the node.

## V. Experimental Results

The experiments were performed in the isolated laboratory containing all appliances working in various combinations. The data acquisition and processing hardware was implemented on the personal computer equipped with the DAQ card responsible for measurements. The analyzed devices included (in brackets the number of the identifier is given): power-saving bulb (1), dryer (2), vacuum cleaner (3), mixer (4), juicer (5) and kettle (6). They were turned on or off in various moments of time, leading to different configurations of devices working at the particular moment. The example of the half-hour experiment is presented in Fig. 4, where the upper diagram contains the measured current, while the lower one shows which appliances were actually turned on. Each color represents the individual device, having one of two values: “0” (when it is turned off) and its identifier (when it is turned on).

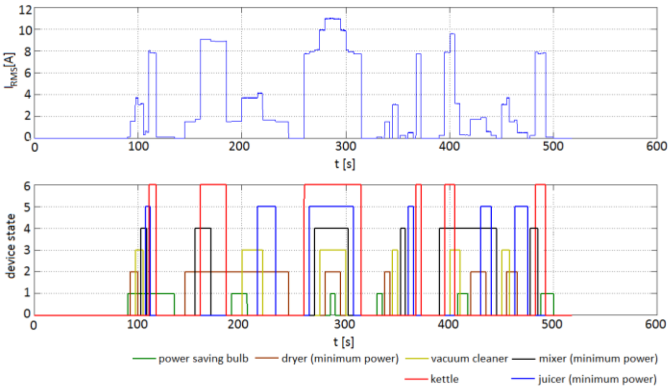


Fig. 4. The example of the operating sequence for six appliances

In the sequence from Fig. 4, 128 events were detected. Because the event detection threshold $\theta$ was set to 100mA, among the actual changes in the appliances configuration, the false alarms were identified as well. For each event the identification procedure was commenced by every classifier in the ensemble. The fragment of the sequence processing is shown in Tab. 1. The first column contains the index of the event, while the second one (*c*) contains the actual identifier of the appliance (according to previous assignments of the identifier to the particular device). Three subsequent columns ($d_{DT}$, $d_{RI}$, $d_{RF}$) contain individual decisions made by the classifiers, while the last column (*d*) contains the decision made by the ensemble. The “0” value for the eight event is for the false alarm (no actual change in the device configuration occurred). The “-1” value for the RI method means no rule was fired - the classifier is unable to make any decision.

TABLE I. RESULTS OF THE SEQUENCE PROCESSING

| No | *c* | $d_{DT}$ | $d_{RI}$ | $d_{RF}$ | *d* |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | **1** |
| 4 | 3 | 1 | 3 | 3 | **3** |
| 8 | 0 | 0 | 0 | 0 | **0** |
| 12 | 5 | 1 | -1 | 5 | **?** |
| 16 | 6 | 6 | 6 | 6 | **6** |
| 31 | 4 | 4 | -1 | 4 | **4** |
| 51 | 3 | 2 | 2 | 2 | **2** |
| 58 | 0 | 0 | 0 | 1 | **0** |

Results show the input of every classifier to the ensemble. Their combination allows for minimizing the incorrect decisions, made by each module individually. In some cases even the combination of classifiers is unable to make decisions (event No. 12), as each produces different output. To solve the problem, the weighting to the voting should be introduced, supporting the classifier with the smallest number of incorrect decisions on the testing data. The comparison of the individual performance of selected algorithms is in Tab. 2. According to them, the outcome of the ensemble would be the (correct) identifier 5, as pointed by the RF, which has the maximal accuracy on the testing set. The column *d* shows the accuracy of the ensemble, which is greater than any individual classifier.

TABLE II. IDENTIFICATION ACCURACY [%] OF INDIVIDUAL CLASSIFIERS

| Algorithm | *DT* | *RI* | *RF* | *d* |
|---|---|---|---|---|
| Overall accuracy | 82.81 | 64.06 | 85.93 | **92.96** |
| False alarm accuracy | 97.82 | 66.30 | 94.56 | **96.73** |
| Appliance identification accuracy | 55.55 | 58.33 | 83.33 | **83.33** |

The following conclusions can be drawn from the experiments:

- The overall ensemble accuracy is higher than its partial accuracies (as different classifiers are the best in detecting false alarms and identifying appliances).
- The most difficult event to detect is the change of the power saving bulb state. It is especially hard to identify when devices with high power consumption operate in the background. Sometimes its switching on or off is missed, which should be corrected by the monitoring system during the next event detection.
- Because of the large number of false alarms, their correct detection must be performed by the classifiers. In most cases, it is possible to achieve by the ensemble.
- The DT is the most efficient in detecting the false alarms, although is not so efficient in other cases. Only two (out of 92) of the former were incorrectly identified.
- The RF has overally the best performance, but also makes more mistakes during the false alarm identification.

## VI. CONCLUSIONS

The proposed scheme of the ensemble proved its efficiency when working with a set of appliances operating in the household. Although each classifier fails in identifying multiple events, their combination allows for detecting most of state changes. The most problematic for the module are false alarms and detections of the power-saving devices. Identifying all these events may need the sophisticated voting mechanisms or implementing more classifiers to the scheme. The problem may be using all presented methods in the embedded microprocessor system, requiring low level code optimization. Also, the module should be tested on the greater number of devices working at the same time, also outside the laboratory environment, where the influence of other apartments may be present. These aspects will be investigated in the nearest future.

## ACKNOWLEDGMENT

This work has been accomplished within the project PBS2/A4/0/2013 “The Non-invasive System for Monitoring and Analysis of Electricity Consumption in the Area of the End-user” financially supported by the National Centre for Research and Development.

## References


[1] M. Zeifman, and K. Roth, “Nonintrusive Appliance Load Monitor-ing: Review and Outlook,” *IEEE Trans. Consumer Electronics, vol*. 57, No. 1, pp. 76-84, February 2011.

[2] H.-T. Yang, H.-H. Chang, C.-L. Lin, "Design a Neural Network for Features Selection in Non-intrusive Monitoring of Industrial Electrical Loads," Proc. 11th International Conference on Computer Supported Cooperative Work in Design, 26-28 April 2007, Melbourne, Australia, pp. 1022-1027.

[3] T. Onoda, H. Murata, G. Ratsch, and K.-R. Muller, "Experimental Analysis of Support Vector Machines with Different Kernels based on Non-Intrusive Monitoring Data," Proc. of the International Joint Conference on Neural Networks, 12-17 May 2002, Honolulu, pp. 2186-2191.

[4] J. Powers, B. Margossian, and B. Smith, "Using a Rule-Based Algorithm to Disaggregate End-Use Load Profiles from Premise-Level Data," IEEE Computer Applications in Power, 1991, pp. 42-47.

[5] J. Z. Kolter and T. Jaakkola “Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation,” Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands.

[6] G.W. Hart, „Nonintrusive Appliance Load Monitoring,” Proceedings of the IEEE, Vol. 80, No. 12, Dec. 1992, pp. 1870-1891.

[7] J. Liao, G. Elafoudi, L. Stankovic and V. Stankovic, "Power Disaggregation for Low-sampling Rate Data," NILM Workshop, Austin, Texas, 3. June 2014.

[8] S. Gupta, M. S. Reynolds and S. N. Patel, “ElectriSense: Single-Point Sensing Using EMI for Electrical Event Detection and Classification in the Home,” Proc. Ubicomp '10 Proceedings of the 12th ACM international conference on Ubiquitous computing, 2010, pp. 139-148.

[9] J. Kelly, W. Knottenbelt, “Neural NILM: Deep Neural Networks Applied to Energy Disaggregation,” 2015, online: arxiv.org/pdf/1507.06594.

[10] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Non-Intrusive Load Monitoring Approaches for Disaggregated Energy Sensing: A Survey," Sensors (Basel). 2012 Dec., 12(12), pp. 16838–16866.

[11] L. Jiang, S. Luo and J. Li, "Automatic power load event detection and appliance classification based on power harmonic features in nonintrusive appliance load monitoring," *Proc. 8th IEEE Conference on Industrial Electronics and Applications* 19-21 June 2013, pp. 1083-1088 .

[12] P. Bilski, „Analysis of the classifier fusion efficiency in the diagnostics of the accelerometer,” Measurement, Vol. 67, May 2015, pp. 116–125.

[13] L. Kuncheva, "Combining Pattern Classifiers. Methods and Algorithms," Wiley and Sons, New Jersey, 2004.

[14] N.-T. Nguyen and H.-H. Lee, "Decision Tree with Optimal Feature Selection for Bearing Fault Detection," Journal of Power Electronics, vol. 8, nº. 1, pp. 101-107, January 2008.

[15] R. S. Michalski, I. Mozetic, J. Hong, and N. Lavrac, “The AQ15 inductive learning system: An overview and experiments, Report 1260,” Department of Computer Science, University of Illinois at Urbana-Champaign, 1986.

[16] P. Bilski, W.Winiecki, „The Rule-Based Method for the Non-Intrusive Electrical Appliances Identification,” Proc. IDAACS 2015 Conference, 24-26 Sept. 2015, Warsaw, Poland, pp. 220-225.

[17] A.-L. Boulesteix, S. Janitza, J. Kruppa and I.R. König, "Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 2, nº. 6, pp. 493–507, November/December 2012.