2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

# Random Forest Based Adaptive Non-Intrusive Load Identification

Jie Mei, *Student Member, IEEE*, Dawei He, *Student Member, IEEE*, Ronald G. Harley, *Fellow, IEEE,* and Thomas G. Habetler, *Fellow, IEEE*

***Abstract*—Non-intrusive load monitoring (NILM) is a load monitoring technique proposed to be used in today's residential energy auditor. It is expected to automatically provide the information of the type, energy consumption, and operation status of the electric loads without getting access to the loads. However, there still not exists any commercialized product so far, mainly because of the extraordinary large load sets comparing with the limited learning data. The fast emerging of new types of loads further aggravates the problem. This paper proposes an adaptive non-intrusive load identification model to address this problem. The proposed model is not dedicated to identify all the loads around the world, but it will grasp knowledge from samples that are not identified in the real application, and gradually form a new learning procedure so as to identify more and more new samples correctly. Random forest algorithm is introduced here to realize the objective and a case study is carried out to verify the effectiveness of the model.**

## I. Introduction

To meet U.S. DOE's goal of achieving market ready net-zero energy residential and commercial buildings, it is proposed to develop a more intelligent energy management system to further reduce building electricity consumption [1]. The Smart Grid [2] and the home automation networks [3] have the potential to become the main energy management tools to realize the goal [4]. However, the deployment of home automation networks might not be feasible under the existing residential condition: the home automation networks require a two-way communication with each household appliance, while most existing appliances don't have necessary communication devices [5]. As an alternative, Non-Intrusive Load Monitoring (NILM) system, which doesn't require any communication with household appliances, is proposed in [6].

A conceptual framework of NILM system is shown in Fig. 1. Briefly speaking, the NILM system monitors voltages and currents at the main breaker or each outlet. Then the state and power consumption of each household load is estimated from the outlet-level information. Further, control outputs based on these estimates can be to appliances to reach goals like energy saving [6].

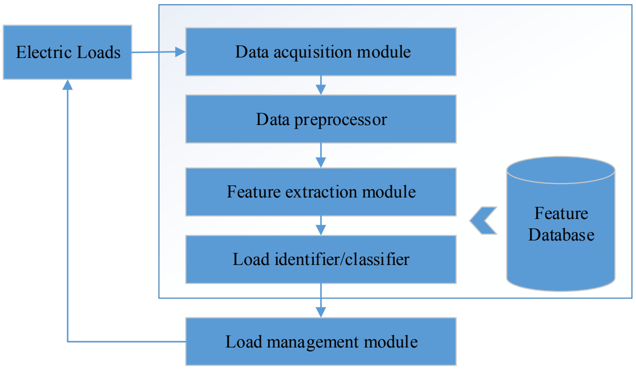

Fig. 1. A conceptual framework of NILM system.[6]

The concept of NILM was firstly proposed in the 1980s. Since then, a great many NILM methods have been investigated [6]. A complete literature review which summarizes almost all the previous work can be found in [7]. The first NILM method, developed by a group in MIT [8], adopts real/reactive power to identify loads. The real/reactive power feature, which the MIT group employs, belongs to the class of 'macroscopic' features, where 'macroscopic' means the features are obtained from low frequency (usually 1Hz or lower) electric measurements. After the MIT method, some other NILM methods using macroscopic features were proposed [9-17]. Then these 'macroscopic' approaches meet with some difficulties: it proves that it's hard to reach high identification accuracy only using macroscopic features [6]. As an example, the MIT method cannot distinguish different loads with similar power ratings [6]. In order to address these problems, most researchers agree that 'microscopic' features extracted from high frequency load signals should be adopted as a complement to the macroscopic features [6]. In [18], the MIT method is extended by incorporating harmonics of load current waveforms as features. Beyond harmonics, other microscopic features like Instantaneous Admittance Waveform (IAW) [19], Instantaneous Power Waveform (IPW) [20], eigenvalues [21], Switching Transient Waveform (STW) [22], Wavelet Transform (WT) features [23-24] and I-V curve features [25-26] have been investigated by various researchers to further improve load identification rate.

However, despite the large set of features, there still not exists a complete feature set available for all the loads around the world [6]. As a result, when the input sample is a load of a model or operating status that is not covered in the training set, incorrect identification frequently occurs. Besides, with the development of home appliance industry, new types, brands and models of appliances are emerging at a fast speed. This further increases the difficulty for load identification.

J. Mei, D. He, R. G. Harley, and T. G. Habetler are with School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: jmei8@gatech.edu; hedawei@gatech.edu; rharley@ece.gatech.edu; habetler@ece.gatech.edu). R. G. Harley is also Professor Emeritus and an Honorary Research Associate at the University of KwaZulu-Natal, Durban, South Africa.

This work is supported in part by the U.S. Department of Energy National Energy Technology Laboratory under Award Number DE-EE0003911 and Eaton Corporation.

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

This paper proposes an adaptive model as an alternative. The adaptive model can grasp knowledge from samples that are not covered in the training set so that it can gradually identify more and more such input samples correctly. Thus it solves the problem posed above.

The proposed model operates as follows: When an input sample doesn't match any existing loads in the model database, it is assigned an 'unknown' label. To gain knowledge from the unknown samples, online clustering algorithm will be applied to them to find new load classes or new load variants. If a new class or new variant is generated, it will be manually assigned a class label, which can either be one of the existing class labels in the case it's a variant of a known load class, or a new one in the case it's a completely new class of loads. After acquiring the new knowledge, the classification model will be updated with them.

The rest of this paper is organized as follows. Firstly Section II describes the load database and feature pool in this study. Then Section III introduces the adaptive model and Section IV carries out a case study to verify the effectiveness of the model. Finally Section V concludes the paper.

## II. Data Preparation and Feature Selection

### A. *Load Space Definition*

A successful load space should be able to represent as many as types of loads in the market. The load space used here is based on the author's previous paper on load study. The detailed information can be found in [32]. To capture the difficulties of NILM, the load space used here almost covers the all appliances with high difficulty to be classified based on the study of Ref [32]. Thus, the following appliances are selected for the research: TV, Monitor, Set-top Box, DVD Player; Microwave; LED Light, Incandescent Light, Florescent Light; Electric Heater, Fan; Printer, Scanner, Laptop, Desktop, and Projector.

The training set and the test set consist of 100 clips and 80 clips respectively, where each clip consists of 10 cycles of voltage and current waveforms together with a class label. The selection of training set and test set should follow the following principal: the data in the test set should be partly different from training set in brands or models.

### B. *Feature Selection*

A clip in the training set or the test set includes 10 cycles of voltage and current waveforms. Assume the waveforms are denoted by

$$V(t) = \sum_{p=1}^{\infty} V_p \sin(p\omega_0 t + \delta_p) \tag{1}$$

$$I(t) = \sum_{p=1}^{\infty} \mathrm{I}_p \sin(p\omega_0 t + \theta_p) \tag{2}$$

Load features are extracted from the above voltage and current waveform. The feature pool in this study contains most of widely used features in literature. They are listed below.

1) Real power $P$ and reactive power $Q$.

2) Displacement power factor

$$dpf = \cos(\delta_1 - \theta_1) \tag{3}$$

3) The total harmonic distortion (THD) in the current

$$THD = \frac{\sqrt{\sum_{p=2}^{\infty} I_p^2}}{I_1} \tag{4}$$

4) Power factor

$$pf = \frac{\cos(\delta_1 - \theta_1)}{\sqrt{1 + THD^2}} \tag{5}$$

5) Crest factor or peak-to-average ratio (PAR):

$$cf = \frac{|I_{peak}|}{I_{RMS}} \tag{6}$$

6) Eigenvalues: for dynamic loads, their waveforms could vary from cycle to cycle. Eigenvalues are introduced to capture this dynamics. In brief, rearrange the waveform series into a matrix with each row representing one cycle, then apply singular value decomposition (SVD) to this matrix, which will decompose the matrix to the product of 2 unitary matrixes and a diagonal matrix. The eigenvalues would be values in the diagonal matrix. More details can be found in [21].

7) Up to 25th harmonics (amplitude and phase) in current.

## III. Model Construction

### A. *Model Framework*

Fig.2 shows the adaptive load identification framework.

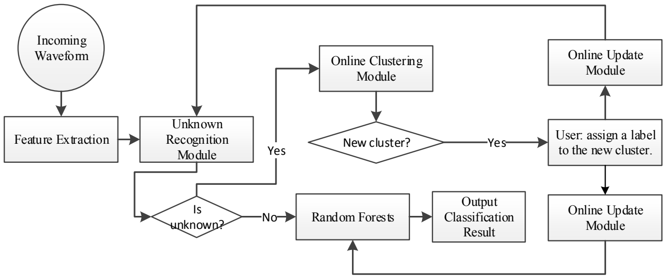

Fig. 2. Framework of proposed adaptive load identification model.

Firstly, features will be extracted from an incoming waveform, yielding a sample point to be processed. Then, the sample point will be processed by the unknown pattern recognition module, which will assign a 'known' or 'unknown' label to the sample point. If the label is 'known', random forest will be applied to the sample point and output a classification result and the model will be suspended waiting for a next incoming waveform. If 'unknown', the sample point will be further delivered to online clustering module, which will store all incoming 'unknown' points and perform online clustering algorithm on them. During the online-clustering process, if a well-shaped cluster is formed such that the points in it could probably derive from the same type of appliance, it will be presented to the user, who would either assign a class label (can be one of existing class labels or a new class) to the cluster or discard it. If the cluster is assigned a label, the random forest and the unknown pattern recognition module will be updated with the labelled cluster. This leads to increase of knowledge of the adaptive model, which would perform better for future incoming waveforms.

### *B. Random Forest*

Random forest is an ensemble learning method for classification (and regression). At training time, a multitude of CARTs are fit into bootstrap sample sets which are generated from the training set. After training, random forest operates by outputting the class that is the mode of the classes output by individual trees [27]. More details can be found at the following pseudo-code [27].

```
Training Stage:
Input: Training Data: $N$ $p$-dimension samples associated with their class labels.
Require Parameter $B$: Number of trees.
Require Parameter $m$: Number of candidate split variables at each split;
Require Parameter $n_{min}$: Minimum node size.
1. For $b$ = 1 to $B$:
        (a) Draw a bootstrap sample $Z^*$ of size $N$ from the training data.
        (b) Grow a random-forest tree $T_b$ to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size $n_{min}$ is reached.
                i. Select $m$ variables at random from the $p$ variables.
                ii. Pick the best variable/split-point among the $m$.
                iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$

To make a prediction at a new point $x$:
Let $\hat{C}_b(x)$ be the class prediction of the $b_{\mathrm{th}}$ random-forest tree. Then $\hat{C}_{\mathrm{rf}}^B(x) = majority\ vote\{\hat{C}_b(x)\}_1^B$
```

Random forest combines the idea of bagging (Bootstrap Aggregating) and CART (Classification and Regression Tree). Bagging is a method that would firstly fit a multitude of models into bootstrap sets resampled from the origin training set and then output the mode of the predictions from the fitted models. Bagging is usually used for reducing variance, namely avoiding over-fitting. CART is a machine learning model that has low bias but suffers from over-fitting. Combining bagging method with CART, which is exactly what random forest does, can effectively avoid over-fitting of CARTs while retaining low bias. Besides, random forest has some other merits, including robust to irrelevant features, insensitivity to outliers in training data and being easy to tune model parameters. All these features make Random Forest one of the most popular models in machine learning [27].

The parameters of random forests can be tuned as follow: The first of the three parameters in random forest is B, the number of trees. The random forest model is the average of B individual trees and it will stabilize when B is big enough. So the principle to tune parameter B is to make B big enough so that the model stabilizes. Generally 500 is enough for B. For parameter m and $n_{min}$, there are recommended values [27]. Set the two parameters as recommended or tune them around the recommended values would generally attain good performance.

### *C. Unknown Pattern Recognition*

The unknown pattern recognition module is based on the threshold Euclidian distance between incoming test points and points in the training set.

Firstly, since scaling of features has a great impact on their contribution to the Euclidian distance, features should be scaled properly before being processed by unknown pattern recognition module. To do this, features are scaled so that their variances are proportional to their importance. The reason is that the more important a feature is, the higher contribution it should have to the distance measure. The importance measure here is the Gini index [27] generated during the training time of Random Forest.

Secondly, clustering analysis will be applied to the training points. In detail, a Gaussian Mixture model will be fit into training points of each class label by EM (Expectation Maximization) algorithm. Gaussian Mixture is a clustering model that assumes the data are generated from superposition of Gaussian distributions. The reason of using Gaussian Mixture model instead of other clustering algorithms like k-means, is that Gaussian Mixture model not only outputs the cluster centers, but also gives the shape of clusters by representing the cluster using a Gaussian distribution with a mean value and a covariance matrix. One thing should be noted here that the EM algorithm requires the number of Gaussian distributions be fixed. Since the number of Gaussian distributions of the training data is unknown, a multitude of Gaussian Mixture Models with different number of Gaussian distributions are fit to the data and the model with the least BIC (Bayesian Information Criteria) will be selected [28].

Thirdly, for an incoming test point, the Unknown Pattern Recognition Module operates by calculating the least distance among the distances between the test point and the Gaussian means in the Gaussian Mixture Models and compare it to an 'unknown' threshold value. When it's bigger than the 'unknown' threshold, it will be regarded as 'unknown'; otherwise 'known'. To determine an appropriate 'unknown' threshold value, the following method is used: manually define several pairs of waveforms so that each pair are critically 'unknown', calculate the distances between waveforms within each pair and take the average over the distances as the 'unknown' threshold.

### *D. Online Clustering*

Online clustering means clustering algorithm that updates itself every time a new clustering is formed.

Once a new 'unknown' point is received, online clustering will check if any clusters are 'qualified' enough so that they could potentially represent a new class of load or a new variant of an existing load. Here 'cluster k is qualified' is defined by (7) (8).

$$\sum_{C(i)=k} 1 \ge A \qquad (7)$$

$$\sum_{C(i)=k} (x_i - \overline{x_k}) \le B \qquad (8)$$

where $C(i) = k$ denotes that point $x_i$ is assigned to cluster $k$ and $\overline{x_k}$ denotes the center of cluster $k$. What these formulas actually mean is that the size of the cluster (number of points assigned to this cluster) must be big enough and meanwhile the scatter (how close the points in the cluster is to the cluster center) is small enough. Going one step further, a cluster is qualified only when it has collected enough number of points and it is well-shaped. The flowchart of the online clustering module is shown in Fig. 3.

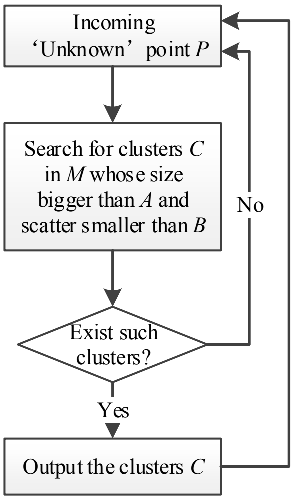
Incoming 'Unknown' point P
Search for clusters C in M whose size bigger than A and scatter smaller than B
Exist such clusters?
No
Yes
Output the clusters C

Fig. 3. Flowchart of online-clustering module.

Qualified clusters will be presented to the user (meanwhile being deleted in the k-means modules), and it's up to user to either assign a class label to the points in the cluster or discard the cluster. Clusters with label assigned will be delivered to the next stage: Online Update.

When a new cluster with label assigned is generated, it means that new knowledge about the load is acquired. So the model needs updating with the new knowledge, which can be divided into two independent part: update of the random forest module and update of unknown recognition module.

1) *Update of random forest:* Here a sequential training algorithm for random forest [31] will be used. It can update random forest with new incoming points. It is presented in the pseudo-code below.
2) *Update of unknown recognition module:* Firstly, since the importance measures (namely Gini Index) of features have been changed after updating the random forest, the scaling of points in the model, including the means and covariance in Gaussian Mixture Model and the cluster centers in online-clustering module should also be updated correspondingly. Secondly, for the Gaussian mixture module, since a new cluster has been discovered, a Gaussian distribution is fit to the points in the new cluster and it is added to the Gaussian Mixture module.

```
Require: Sequential training example $< x, y >$
Require: The size of the forest: $T$
Require: The minimum number of samples: $\alpha$
Require: The minimum gain: $\beta$
// For all trees
for $t$ from 1 to $T$ do
    $k \leftarrow$ Poisson($\lambda$)
    // Update k times
    for $u$ from 1 to $k$ do
        $j$ = findLeaf($x$).
        updateNode($j$; $< x, y >$).
        if $|\mathcal{R}_j| > \alpha$ and $\exists s \in \mathcal{S}: \Delta L(\mathcal{R}_j, s) > \beta$ then
            Find the best test: $s_j = \arg\max_{s \in \mathcal{S}} \Delta L(\mathcal{R}_j, s)$
            createLeftChild($\mathbf{p}_{jls}$)
            createRightChild($\mathbf{p}_{jrs}$)
            UpdateGiniIndex.
        end if
    end for
end for
Output the forest F.
```

## IV. Simulation

A total of three tests will be carried out in the case study. They are summarized below:

TABLE I. SIMULATION LISTS

| Group | Training | Test | Model |
|---|---|---|---|
| B | Training Set | Test Set | Support Vector Machine |
| E | Training Set | Test Set | Random Forest with Unknown Pattern Recognition |
| F | Training Set | Test Set | Adaptive Random Forest |

Test group A is benchmark model. The purpose is to show random forest model can achieve better identification rate. Test group B would show that the easily-misclassified loads, defined in the previous test, would be labelled as 'unknown' of the unknown pattern recognition module. Finally test group C would validate the adaptive module: knowledge could be collected from unknown points and be used to increase identification rate for future inputs.

### A. Identification Rate Comparison

The results are shown in the table II. Parameters of the SVM are selected through cross validation. Parameters of Random Forests are selected according to the method described in chapter 3 section B in this report.

In Table II, identification rates of groups A and B is much lower than that of group C. This proves that adaptive random forest can achieve a better identification rate.

TABLE II. PARAMETER SETTINGS AND RESULTS

| Group | Identification Rate | Parameter Settings |
|---|---|---|
| A | 58.97% | C=0.18, γ=1/21, ϵ=0.01 |
| B | 78.47% | ntrees=500, mtry=4 |
| C | 94.17% | ntrees=500, mtry=4 |

TABLE III. EASILY MISCLASSIFIED LOADS

| Group Name | Identification Rate |
|---|---|
| LCD TV-Toshiba 32-Default-steady | 0% |
| STB-DishTV-ViP622-steady | 0% |
| DVD-Toshiba-4990-steady | 0% |
| Florencent Light-13W-Default-on_off | 3.75% |
| Printer-HP-deskjet-color_printing | 3.75% |
| Scanner-Microtek-ScanMaker4800-scan | 0% |
| Desktop-Dell-Tech-steady | 0% |
| Laptop-HP-8740W-steady | 0% |
| Heater-Holmes-Default-low | 0% |

### B. *Test of Online Clustering Algorithm*

For groups A and B, loads that are easily misclassified are listed below in table III. A group of loads is regarded as 'easily misclassified', when the identification rate in the group is lower than 10%.

Test group C functions as an evaluation of the complete adaptive model proposed in this report. Random Forests along with unknown pattern recognition module, online clustering module and online update module is applied to the test set. Points in test set are processed by the adaptive module in random order. Clusters are generated by online-clustering module. Table IV shows the generated clusters. They are sorted by discover time in ascending order, where the discover time of a cluster is defined as the number of test points already processed right before the cluster is generated. Types of loads in the clusters scatters of clusters are also shown. After a cluster is generated, it's up to user to decide whether to update the random forest with loads in the cluster or not, and to assign a label to the cluster before updating. Here the following rule is used: if the Gini index of the points in the cluster is less than 0.1, then apply updating algorithm and the class label is the mode of the types of the points in the cluster. Gini index of a set of points is defined in (9):

$$L(R) = \sum_{i=1}^{K} p_i(1-p_i) \qquad (9)$$

where R denotes the point set and K denotes number of classes and $p_i$ denotes the label density of class i in the points set. The reason of using Gini index is that Gini index measures the homogeneity of a point set, and the smaller it is, the more homogeneous the point set is.

TABLE IV. CLUSTERS DISCOVERED BY ONLINE CLUSTERING MODULE

| No. | Discover Time | Loads inside Cluster | Updated? |
|---|---|---|---|
| 1 | 708 | 'STB-DishTV-ViP622-steady' [26] | Yes |
| 2 | 708 | 'Heater-Holmes-Default-low' [27] | Yes |
| 3 | 708 | 'Desktop-Dell-Tech-steady' [26] | Yes |
| 4 | 1114 | 'Florencent Light-13W-Default-on_off' [29]<br>'Florencent Light-19W-Default-on_off' [ 1] | Yes |
| 5 | 1114 | 'Printer-Dell-3130cn-doublesided' [ 1]<br>'Scanner-Microtek-ScanMaker4800-scan' [38] | Yes |
| 6 | 1645 | 'Desktop-iMac-iMac 7.1-steady_02' [ 2]<br>'Laptop-HP-8740W-steady' [33] | Yes |
| 7 | 1645 | 'DVD-Toshiba-4990-steady' [37]<br>'Printer-Dell-3130cn-doublesided' [ 1] | Yes |

According to Table IV, 7 clusters are discovered and all of they are homogeneous enough to be learned by Random Forest. A comparison between Table IV and Table III shows that each cluster in table IV represents an easily misclassified load type in table III and 7 out of 9 easily misclassified load types are captured by the online clustering module. This validates the effectiveness of the online clustering algorithm.

The classification result of the adaptive model is summarized here: 405 out of 2240 points are 'unknown'; for the 1835 known points, identification rate of the adaptive model is 94.17% (1728 out of 1835 are correctly classified), while the identification rate of the non-adaptive model is 78.47% (1440 out of 1835 are correctly classified). This suggests that, after recognizing 405 points as unknown, most of which are those easily misclassified loads as shown in Test B, and capturing knowledge from them, the adaptive model avoids 288 misclassifications for the rest test points and increases the identification rate by around 16%. This proves the effectiveness of the online update module and hence, the complete proposed adaptive model.

## V. CONCLUSIONS

This paper proposes a new perspective for the nonintrusive load identification problem. Instead of introducing more features, the report presents an adaptive solution consisting of random forest, unknown recognition module, online clustering module and online update module. The adaptive solution proves to be able to recognize those easily misclassified loads as unknown and correctly identify them after gaining knowledge from those unknown.

The core idea of the proposed model is 'gaining knowledge and self-correcting while operating'. This adaptive model might present a practical and effective solution.

## DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

## REFERENCES

[1] U. DoE, "Recovery Act: Advanced Energy Efficient Building Technologies," Washington, DC, 2009.

[2] S. Park, H. Kim, H. Moon, J. Heo, S.Yoon, "Concurrent Simulation Platform for Energy-Aware Smart Metering Systems," IEEE Trans. Consumer Electron., vol. 56, no. 3, pp. 1918-1926, Aug. 2010.

[3] K. Balasubramanian and A.Cellatoglu, “Improvements in Home Automation Strategies for Designing Apparatus for Efficient Smart Home,” IEEE Trans. Consumer Electron., vol. 54, no. 4, pp. 1681-1687, Nov. 2008.

[4] J. Heo, C. S. Hong, S. B. Kang, S. S. Jeon, “Design and Implementation of Control Mechanism for Standby Power Reduction,” IEEE Trans. Consumer Electron., vol. 54, no. 1, pp. 179-185, Feb. 2008.

[5] S. Lipoff, “Home Automation – the Unrealized Promise,” IEEE Consumer Electronics Society Newsletter, pp. 13-14, Fall 2010.

[6] M. Zeifman and K. Roth, “Nonintrusive Load Appliance Load Monitoring: Review and Outlook,” IEEE Trans. Consumer Electronics, vol.58, no.1, Feb. 2011.

[7] Y. Du, L. Du, B. Lu, R. G. Harley, and T. G. Habetler, "A review of identification and monitoring methods for electric loads in commercial and residential buildings," In Proc. 2010 IEEE Energy Conversion Conf. and Expo., Atlanta, GA, 2010, pp. 4527-4533.

[8] G. W. Hart, “Nonintrusive Appliance Load Monitoring,” Proceedings of the IEEE, Vol. 80, pp. 1870-1891, 1992.

[9] L. K. Norfold, S. B. Leeb, “Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms,” Energy and Buildings, Vol. 24, pp. 51-64, 1996.

[10] A. I. Albicki, and A. Cole, “Data Extraction for Effective Non-Intrusive Identification of Residential Power Loads,” IEEE Instrumentation and Measurement Technology Conference, pp. 812-815, 1998.

[11] A. . Albicki, and A. Cole, “Algorithm for Non-Intrusive Identification of Residential Appliances,” Circuits and Systems, Proceedings of the IEEE International Symposium on, Vol. III, pp. 338-341, 1998.

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

[13] L. Farinaccio, R. Zmeureanu, “Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses,” Energy and Buildings, vol. 30, pp. 245-259, 1999.

[14] M. L. Marceau, R. Zmeureanu., “Nonintrusive Load Disaggregation Computer Program to Estimate the Energy Consumption of Major End Uses in Residential Buildings,” Energy Conversion & Management, Vol. 41, pp. 1389-1403, 2000.

[15] M. Baranski, J. Voss, “Non-Intrusive Appliance Load Monitoring Based on an Optical Sensor,” IEEE Power Tech Conference, Bologna, 2003.

[16] M. Baranski, J. Voss, “Genetic Algorithm for Pattern Detection in NIALM Systems,” IEEE International Conference on Systems, Man and Cybernetics, pp. 3462-3468, 2004.

[17] M. Baranski, J. Voss, “Detecting Patterns of Appliances from Total Load Data Using a Dynamic Programming Approach,”Fourth IEEE International Conference on Data Mining (ICDM’04), 2004.

[18] C. Laughman, et al., “Power Signature Analysis,” IEEE Power & Energy Magazine, pp. 56-63, March/April 2003.

[19] K. Musierowicz, et al., "A fuzzy logic- based algorithm for discrimination of damaged line during intermittent earth faults," In Power Tech, 2005 IEEE Russia, 2005, pp. 1-5.

[20] E. A. Cano Plata and H. E. Tacca, "Power quality assessment and load identification," In Harmonics and Quality of Power, 2000. Proceedings. Ninth International Conference on, 2000, pp. 840-845 vol.3.

[21] H. LAM, et al., "Building a vector-based load taxonomy using electrical load signatures," presented at the Proceedings ICEE 2005, 2005.

[22] J. Liang, S. K. K. Ng, G Kendall, and J. W. M. Cheng, “Load Signature Study—Part I: Basic Concept, Structure, and Methodology,” Power Delivery, IEEE Transactions on, vol. 25, no.2, pp. 551-560, April 2010.

[23] W. L. Chan, et al., "Harmonics load signature recognition by wavelets transforms," In Electric Utility Deregulation and Restructuring and Power Technologies, 2000. Proceedings. DRPT 2000. International Conference on, 2000, pp. 666-671.

[24] W. L. Chan, et al., "Wavelet feature vectors for neural network based harmonics load recognition," In Advances in Power System Control, Operation and Management, 2000. APSCOM-00. 2000 International Conference on, 2000, pp. 511-516 vol.2.

[25] W. K. Lee, G. S. K. Fung, H. Y. Lam, F. H. Y. Chan, M. Lucente, “Exploration on Load Signatures,” International Conference on Electrical Engineering (ICEE), ref. 725, 2004.

[26] H.Y. Lam, G. S. K. Fung, W. K. Lee, “A Novel Method to Construct Taxonomy of Electrical Appliances Based on Load Signatures,” IEEE Trans. Consumer Electron., vol. 53, no. 2, pp. 653-660, May 2007.

[27] T. Hastie, R. Tibshirani and J. Friedman, The Elements of Statistical Learning: data mining, inference, and prediction, 2nd ed., New York: Springer, Feb. 2009, p. 587-604.

[28] R. J. Steele and A. E. Raftery, “Performance of Bayesian Model Selection Criteria for Gaussian Mixture Models,” Dept. of Statistics, University of Washington, Tech. Rep. No. 559, September 2009.

[29] online k-means

[30] D. Pelleg and A. Moore, “X-means: Extending K-means with efficient estimation of the number of clusters,” In Proceedings of the 17th International Conf. on Machine Learning, pp. 727-734, 2000.

[31] A. Saffari, C. Leistner, J. Santner, M. Godec and H. Bischof, “On-line random forests,” In Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, pp. 1393-1400, 2009.

[32] D. He, L. Du, Y. Yang, R. G. Harley, and T. G. Habetler, “Front-End Electronic Circuit Topology Analysis for Model-Driven Classification and Monitoring of Appliance Loads in Smart Buildings,” Smart Grid, IEEE Transactions on , vol.3, no.4, pp.2286-2293, Dec. 2012.