# Comprehensive Performance Comparison of Supervised Machine Learning Algorithms in Non-Intrusive Load Monitoring

Ahmet Furkan Erşen, Ayşe Kübra Erenoğlu, Ozan Erdinç
Electrical Engineering Department
Yildiz Technical University
İstanbul, Turkey
ahmet.furkan.ersen@std.yildiz.edu.tr;
erenayse@yildiz.edu.tr; oerdinc@yildiz.edu.tr

İbrahim Şengör
Electrical and Electronics Engineering Department
İzmir Katip Çelebi University
İzmir, Turkey
ibrahim.sengor@ikcu.edu.tr

João P. S. Catalão
Faculty of Engineering of the University of Porto and INESC TEC
Porto, Portugal
catalao@fe.up.pt

*Abstract*—**Recent developments in the field of smart grid have led to renewed interest in load monitoring strategies for achieving effective energy management schemes. There are vast amount of published studies describing the role of non-intrusive load monitoring (NILM) system based on various learning algorithms. It is widely known that the accuracy of load identification depends strongly on utilized methods and its features. Thus, the main aim of this study is to investigate the comparative accuracy of machine learning algorithms which have the same training data with different feature subsets. Afterwards, a low-cost data acquisition system for NILM using bagged tree ensemble algorithm is developed and demonstrated in detail. The proposed structure is tested on the ThingSpeak IoT platform to reveal the effectiveness of the evaluated concept.**

***Keywords—Bagged tree ensemble algorithm, low-cost data acquisition, non-intrusive load monitoring, supervised learning, smart grid.***

## I. Introduction

### A. Motivation and Background

Growing income, quickening rate in the industrialization, changes in economy and globalization caused to striking increases in electricity demand which are evaluated as beyond the capacity of obsolete and aged design of power system [1]. The rising demand are supplied by generally fossil-based resources which results in increasing environmental concerns due primarily to carbon emission which is the biggest challenge for today's world [2]. The renewable-based energy production has become popular and effective instrument in such occasions for ensuring power supply in a low-carbon community considering the depletion of hazardous sources. However, harnessing the renewable-based distributed generations cause severe operational challenges from the system operator point of view because of their stochastic nature. In order to tackle the barriers and deal the challenges particularly unbalanced supply-demand, the concept of smart grid was represented as a promising platform which is the crucial turning point in power system history [3].

Smart grid paved the way for increasing deployment of energy management schemes in building and appliance levels which are one of the most brilliant solutions in terms of providing efficient usage of electrical power [4]- [5]. Thanks to the proposed energy management approaches, transmission, distribution and consumption areas can be monitored, controlled and manipulated to receive signals from system operators for performance optimization [6].

The fundamental instrument in this research area is real-time load monitoring which is evaluated as an effective framework for obtaining required information and processing them in several demand response programs, load shedding and/or load shifting strategies in the paradigm of smart grid.

In order to monitor the energy consumption profiles of any appliance, the installation of a individual sensor for each of them is necessary in traditional load monitoring systems which are called as Intrusive Load Monitoring (ILM). Unlike the ILM, a Non-Intrusive Load Monitoring (NILM) technique was first introduced by G. W. Hart [7] in 1992 which has several superiorities for performing strategies in an economic fashion. Therefore, NILM system has been comprehensively and systematically examined with growing body of literature for practical implementations instead of ILM as an innovative alternative for power system operations [8,9]. One of the most significant advantageous points for deploying this algorithm is that it needs a reduced number of sensors paving a cost-effective solution in operational and maintenance scale [7].

On the other hand, it is to be noted that it is possible to disaggregate the loading pattern of individual appliances from the aggregated load profile of end-users (generally power profile). This makes possible to conduct an accurate load identification by integrating data extracted from only voltage and current sensors where installed at the entrance of service. Therefore, the NILM technology has become a burgeoning area for active research and attracting great interest from both academic community and industrial stakeholders.

Supervised and unsupervised approaches already exist in the huge literature for implementing NILM which is a high tech innovative solution for commercial and residential end-users. To expand our knowledge, a general framework of this architecture consisting of five steps is illustrated in Fig. 1 [10,11].

### B. Relevant Literature

There are numerous studies in the literature regarding low cost data acquisition systems for NILM researches. Among them, Sawyer et al. [12] provided an analysis on low-cost energy management systems for homes using NILM devices. The NILM is used in [12] for diagnostics purposes within residential areas such as monitoring air conditioner air filters, detection of compressor failure in electric heat pumps, and detection of poorly insulated water heater.

978-1-7281-4701-7/20/$31.00 ©2020 IEEE

Adabi et al. [13] developed a cost-effective instrumentation tool via NILM to support a residential energy management system. The system in [13] was designed to support fast sampling rates up to 65 kHz and amplitude resolution up to 24 bits. Also, internal processor used in [13] could run NILM algorithms in real time. The study in [13] used k-NN algorithm for load classification, however one smart plug per individual appliance was employed. Srinivasan et al. [14] designed a low-cost non-intrusive device identification system to identify devices turned on at any instance in time using devices' current-peak signature. However, the system in [14] did not support energy disaggregation since it was designed with a microcontroller that could not run energy disaggregation algorithms when implemented.

The event detection and classification were used for identifying loads in [15,16]. A low-cost smart sensor for NILM applications was proposed by Nardello et al. [15] using various steady state and transient features to classify and detect loads such as electric oven, electric toaster, fridge, washing machine, and microwave. However, the system in [15] was designed for event detection of loads. A low-cost hardware setup, called Appliance Identification and Management System (AIMS) was developed to identify and control the appliances remotely by Khan et al. [16]. The only research in the literature that used bagged tree ensemble algorithm was presented by Le et al. [17] for household appliance classification.

### C. *Content, Contributions and Organization of the Paper*

This study aims to contribute to this growing area of research by performing different analyses. The popular supervised machine learning algorithms namely logistic regression, artificial neural networks, SVM and decision tree algorithm are compared from different points of view in order to identify theirs superiorities, particularly. Different methods have been considered to determine best and accurate model for effective load identification in power consumption areas for further demand response applications. Prediction accuracy, training time and the features which are used in training the related algorithms are considered as the indicators.

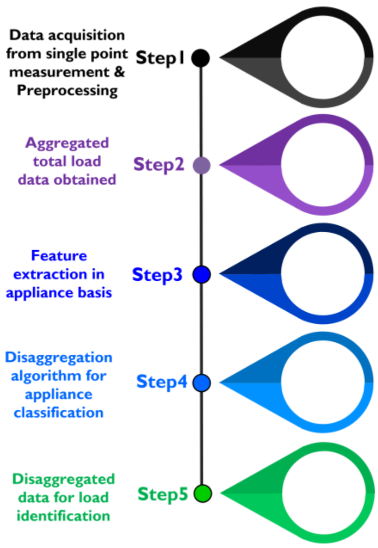


Fig. 1. A general framework of NILM technology [10].

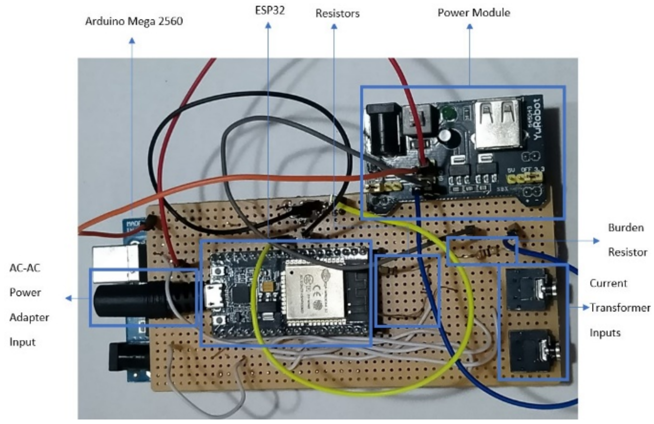


Fig. 2. The data acquisition system.

In addition, bagged tree ensemble algorithm NILM concept is further provided to develop a low-cost data acquisition system where the mentioned algorithm is used for both classification and regression problem due to proper disaggregation of individual load real power for. The energy disaggregation system using MATLAB classification and regression learner toolboxes are used, and the proposed structure is tested on trained models on ThingSpeak IoT platform to reveal the effectiveness of the evaluated concept.

The organization of the paper is as follows: in Section II, the data acquisition system is described. Then, in Section III, the trained machine learning algorithms for load classification are explained, briefly. Thereafter, in Section IV, the experimental results are discussed comparatively with evaluating the performances of the algorithms used. The demonstration of energy disaggregation system on ThingSpeak platform is described in Section V. Finally, the concluding remarks are given in Section VI.

## II. Data Acquisition System

In this section, the data acquisition system used for training machine learning algorithms will be described. The data acquisition system can sample instantaneous voltage and current values up to 6.4 kHz for feature extraction without current harmonic feature for training. If current harmonic feature is needed for machine learning algorithms, maximum sampling rate decreases to 3.2 kHz because of accuracy of measurements of other features. The data acquisition system can be operated within MATLAB on computer using serial communication or a local server using Wi-Fi network for logging of the training data. If MATLAB is used, then continuous data logging can be achieved since memory of computer and features of training data can be sampled up to 16 Hz. If a local web server is used, then data logging can be achieved up to 1 hour 15 minutes with feature sampling rate 1 Hz and up to 15 minutes with feature sampling rate 16 Hz regarding the low memory of the microcontroller.

Regarding the hardware side, the most important components of the data acquisition system are current transformers and ac-ac power adapter, and the transformation ratios were determined as 220/9 V for ac-ac power adapter, 30A/1V, and 100A/50mA for current transformers, respectively.

For the purpose of reducing the 9V output of voltage transformer to 0-5V, a voltage divider was used in this circuit. As microcontrollers, Arduino Mega 2560 and ESP32 are chosen as they are among the most well-known types. Arduino is used for sampling of training data features with MATLAB via serial communication in data logging part and for sampling energy disaggregation systems' inputs in demonstration part. ESP32 is used to provide more flexible data logging of training data features via using wireless communication and to send sampled inputs of energy disaggregation system to ThingSpeak platform in demonstration part. The data acquisition circuit is shown in Figure 2.

On the other hand, for the software side, ArduinoFFT library is used. In this software, the instantaneous voltage and current values of the related load are calculated using analog input signals derived from current transformers and ac-ac power adapter. The analog input signals are sampled at 3.2 kHz for 20 ms which means to have 128 discrete values in one period. After sampling and calculating one period of instantaneous voltage and current, rms voltage and current, real and apparent power, and power factor are calculated.

Thereafter, the current harmonics in percentage up to 13th order are extracted from instantaneous current values via ArduinoFFT library functions. Finally, all these features are sent to MATLAB via serial communication for data logging, to computer or ESP32 for wireless data logging to any devices as computer, smart phone etc. using local web server. In demonstration part, all these features are sent to ESP32 via serial communication for sending these features to ThingSpeak platform.

In ESP32 software of data logging part, Wifi, WifiClient, WebServer, ESPmDNS, SPIFFS, and time libraries are used in data logging part. First four libraries are used to create a local web server for accessing file which has saved data that comes along Arduino from any device that is connected same Wi-Fi network with ESP32. SPIFFS library is used to save data gathered from Arduino via serial communication to a file which is created in flash memory of ESP32.

The time library is used to obtain time data from a NTP server to indicate time that data arrive to ESP32 from Arduino. In demonstration part, for Wi-Fi connection of ESP32 Wifi library is used and ThingSpeak library is used to send input of energy disaggregation system implemented on ThingSpeak platform. In data logging, first the data are read over serial communication channel and saved to a string variable. Afterwards, the time stamp is obtained and saved to a string variable.

Finally, time stamp and data derived from Arduino is sequentially saved in a file created for data logging in flash memory of ESP32.

In demonstration part, the relevant data are read firstly and then saved as a string variable. However, these time data are sent from Arduino per second and data are saved as a string variable with a length of 15 as the data can be sent to ThingSpeak one time per 15 second due to limitations of ThingSpeak for free users. After 15 inputs of energy disaggregation system are saved to string variable, this string variable is sent to ThingSpeak platform for prediction of individual real load power.

## III. Machine Learning Algorithms

In this section machine learning algorithms trained for load classification in the research part and energy disaggregation system in demonstration part are considered. Logistic regression algorithm, artificial neural network, support vector machine, and decision tree algorithms are trained for load classification. MATLAB functions and the classification learner toolbox of MATLAB are used for support machine learning algorithm and the decision tree algorithm, respectively.

In the logistic regression algorithm, the sigmoid function is used as a hypothesis function with expressing in (1). Also, in order to prevent overfitting, Eq. (2) is used as the regularized cost function.

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \tag{1}$$

$$\begin{aligned} J(\theta) = [-\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right) \\ + \left(1 - y^{(i)}\right)\log\left(1 - h_\theta\left(x^{(i)}\right)\right)] \\ + \frac{\lambda}{2m}\sum_{j=1}^{n} \theta_j^2 \end{aligned} \tag{2}$$

In training process of logistic regression algorithm, the $\theta$ parameters of hypothesis function are firstly found by using training set and cost function in advanced optimization algorithm (fmincg). The loads are predicted with utilizing mentioned parameters as input signals to hypothesis function. The output values of the hypothesis function are compared with the output values of the training set which is an indicator for the prediction accuracy of training set.

An artificial neural network is used with one input layer, one hidden layer, and one output layer. The input layer can consist of 35 units in maximum which varies depending on the number of used features in experimental analysis. There are 25 units in the hidden layer while 7 units exist in output layer due to considering three loads in this algorithm.

The sigmoid function expressed in Eq. (1) is used as the activation function in the artificial neural network. While training first features are normalized by using equation (3) where $X_n$ is normalized features, $X_m$ is mean of features, and $X_{std}$ is standard deviation of features since normalization decreases training time. For example, training takes up to 12 seconds with normalization of features, while without normalization training takes minimum 8 minutes to get same training accuracy.

After,_the initial $\theta$ parameters (weights) were randomly defined. The weights of the model were calculated by using advanced optimization algorithm (fmincg). Since the randomly defined weights of the parameters are in matrix basis, there should be converted in vector format. The cost function of the artificial neural network, training set and initial weight values are the inputs of advanced optimization algorithm (fmincg) in which makes possible to obtain the parameters in vector format. After, the weight parameters are again converted in matrix basis and applied to the input of the activation function together with the training set. The outputs of the activation function are load predictions. The predictions are compared with the actual output values in the training set and the average prediction accuracy is calculated.

$$X_n = \frac{X - X_m}{X_{std}} \quad (3)$$

Support vector machine is a modification of logistic regression in order to fit non-linear decision boundaries by using kernels. Kernels are used to obtain new features by calculating similarities of landmarks and features. In support vector machine algorithm, completely built-in Matlab functions are used.

First, training is done by using fitcecoc function, then load predictions are made by predict function and finally prediction accuracy of training set and test set are calculated. Gaussian kernel is used since there are small numbers of features and intermediate number of training examples. However, test accuracy was very low, so default options are used. Calculation of prediction accuracy of training set and test set is shown in equation (4) where $p_{true}$ is the number of true predictions and $p_{total}$ is total number of predictions.

$$Accuracy = \frac{p_{true}}{p_{total}} \times 100 \quad (4)$$

## IV. EXPERIMENTS AND RESULTS

The obtained results are compared in terms of the prediction accuracy of training set, test set and training time in this section for machine learning tools. The prediction accuracy means a rate in percentage that equals percentage of accurate classifications out of all predictions and training time means the time passing to train machine learning algorithms in seconds. Training and test accuracy show the success of a machine learning algorithm to determine whether it is useful for practical applications or not. For example, if training accuracy high while test accuracy is low, that means an overfitting problem available which should be fixed. Herein, online learning is important as it will increase accuracy of the algorithm. Therefore, training time is important for online learning algorithms.

In the research part, all possible combinations of loads without all the loads are closed are considered in creating training sets. There are three residential appliances which are hair dryer, television, and the mobile phone charger in the research. Thus, 7 classes are considered. Since hair dryer has 2 modes which are low blow-low heat and high blow-high heat, both hair dryer modes are sampled during data logging. Before the experiments, data logging is realized for all these classes.

The average duration of data logging is 5 minutes with up to 16 Hz. After that, the training data is constructed as follows: 100 samples per load combination are chosen randomly from the logged data to reduce the size of training data leading in total 700 samples on the dataset for the research. About 70% of total samples of dataset is used as training dataset which is equal to 490 samples and 30% of total samples is used as test dataset which is equal to 210 samples. The training and test sets are chosen randomly.

The features calculated on the data acquisition system are divided into four sub-groups as Dataset 1, 2, 3, and 4 to investigate effects of features on training and test accuracy of trained machine learning algorithms in this study.

In the first sub-group, only rms values of voltage and current are considered as features while power factor, active and apparent power values considered in addition to the features in the first sub-group in following sub-group. Similarly, in addition to second sub-group, THD values of current and voltage is used in third one. Lastly, all calculated features in the data acquisition system are used by also adding the percentage harmonic values up to the 13th harmonic order of the current and voltage in fourth sub-group. Also, the maximum iteration number of advanced optimization algorithm (fmincg) is changed to see effect of maximum iteration value during artificial neural network algorithm training. Maximum iteration is the maximum number of iterations that advanced optimization algorithm tries to minimize cost of hypothesis function.

In logistic regression algorithm training experiments, the value of the λ is taken as 0.1 while training logistic regression, since it is the regularization coefficient which gives best accuracy results. The results are shown in Table I. Unlike the logistic regression algorithm training experiments, the effects of the maximum iteration number of the advanced optimization algorithm on training time of artificial neural network algorithm and prediction accuracy of trained artificial neural network algorithm were also_investigated during the artificial neural network algorithm training experiments. In feature experiments, the maximum number of iterations was taken as 50 since it provides the minimum artificial neural network training time. All features were used in the algorithm training in maximum iteration experiments. In all experiments, the value of the λ was taken as 0.1 while training artificial neural network algorithm since it is the regularization coefficient which gives best accuracy results. The test results obtained by using the features in various scenarios which were shown in Table I and the results of the tests performed by changing the maximum number of iterations were shown in Table II.

To evaluate support machine algorithm, training time and prediction accuracy of training and test set are examined. Because of built-in MATLAB functions are used results were better than algorithms we wrote. The results are shown in Table I. To evaluate decision tree algorithm first decision tree (fine tree) algorithm is trained using MATLAB classification learner toolbox. In Table I, training and test results are shown.

To choose the best machine learning algorithm that will be used in demonstration part, the feature experiments as explained above is also realized with training decision trees, which are fine, medium, and coarse trees, and ensemble algorithms which are boosted and bagged tree ensemble algorithms using classification and regression learner toolboxes. In these experiments the notebook charger is used as the fourth load. The notebook has two modes as fast and normal charging. To achieve regression algorithm training with data acquisition system that has only one current transformer which is used to sample aggregated current, the following method is used. Random individual load real power for four loads between their maximum and minimum real power values are generated using the individual real power data sampled in the beginning of every data logging for all load combinations.

TABLE I. FEATURE EXPERIMENT RESULTS FOR TRAINING SUPERVISED MACHINING LEARNING ALGORITHMS

| Algorithms | Logistic Regression | | | Artificial Neural Network | | | Support Vector Machine | | | Decision Tree | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Features | Training Time (seconds) | Training Accuracy (%) | Test Accuracy (%) | Training Time (seconds) | Training Accuracy (%) | Test Accuracy (%) | Training Time (seconds) | Training Accuracy (%) | Test Accuracy (%) | Training Time (seconds) | Training Accuracy (%) | Test Accuracy (%) |
| Dataset1 | 4 | 60.82 | 54.76 | 4 | 51.02 | 44.29 | 2 | 61.84 | 48.09 | 0.6 | 95.3 | 83.33 |
| Dataset2 | 4 | 78.36 | 82.86 | 4 | 73.88 | 71.43 | 2 | 99.39 | 98.09 | 0.75 | 99.4 | 95.24 |
| Dataset3 | 4 | 93.06 | 95.24 | 4 | 74.95 | 78.09 | 2 | 99.39 | 98.57 | 0.72 | 99 | 93.33 |
| Dataset4 | 4 | 95.92 | 95.72 | 4 | 96.53 | 93.33 | 2 | 99.59 | 99.05 | 0.54 | 99.58 | 95.24 |

TABLE II. MAXIMUM ITERATION EXPERIMENT RESULTS FOR ARTIFICIAL NEURAL NETWORK TRAINING

| **Maximum Iteration Number** | **Training Time (seconds)** | **Training Accuracy (%)** |
|---|---|---|
| 50 | 3 | 97.75 |
| 75 | 4 | 99.79 |
| 125 | 5 | 99.79 |
| 250 | 8 | 100 |

The training accuracy of 99.4% and training time less than 20 seconds are achieved via training bagged tree ensemble algorithm using classification learner toolbox of Matlab. The root mean square error (RMSE) value of regression learner toolbox indicates the success of regression algorithm.

The RMSE value should be as low as possible for using qualified machine learning algorithms for energy disaggregation systems to make successful individual real power value predictions close to the measured individual real power values. After training machine learning algorithms using regression learner toolbox, the minimum RMSE values belong to bagged tree ensemble algorithm trained algorithms for all four loads are compared. Maximum RMSE value among the trained bagged tree algorithms for all four loads belongs to trained bagged tree algorithm for hair dryer which equals to 6.

## V. DEMONSTRATION OF ENERGY DISAGGREGATION SYSTEM ON THINGSPEAK PLATFORM

In this study, the demonstration is also used to proof that trained machine learning algorithm models are working stable as expected from the training results in previous section.

The energy disaggregation system demonstrated on ThingSpeak platform first downloads trained classification and regression algorithm models from a data sharing platform. These models are extorted via classification and regression toolboxes of MATLAB as explained in the previous section and uploaded to the same data sharing platform for using these models in energy disaggregation system demonstration. There are one trained classification model and four trained regression model for four loads. After downloading models to ThingSpeak, the channels are read including input data of energy disaggregation system sent from ESP32 on the data acquisition system to ThingSpeak platform. The input data are sent in string format and need to be parsed. After data parsing, input data are ready to use for energy disaggregation. First, trained classification model predicts loads which are ON at that moment and indicated as 1 with output variable of the classification model. For the loads that are OFF, the output variable of classification model is equal to 0.

After obtaining operated loads at a specific moment, the trained regression models are used to predict individual load real power for energy disaggregation system. Aggregated real power of loads is calculated via summing predicted individual load real powers. At last, energy disaggregation system assigns measured rms voltage and current, disaggregated individual real powers of all four loads, measured and calculated aggregated real power data to a new channel in order to report results of the system to the user as shown in Fig. 3.

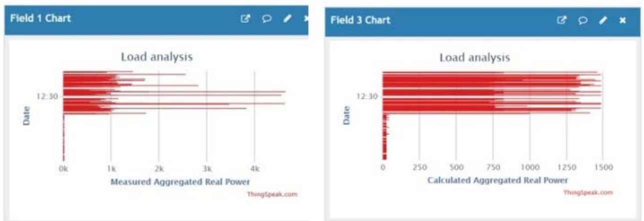


Fig. 3. Some of the reported results of energy disaggregation system.

## VI. CONCLUSION

In this study, a low-cost data acquisition system was developed to train machine learning algorithms and demonstrate an energy disaggregation system. In this study, the performances of different supervised machine learning algorithms were compared in terms of accurate load identification as a fundamental part of energy management schemes. The energy disaggregation system model based on NILM technology using bagged tree algorithm was performed in both Arduino and MATLAB environment. The features of three residential appliances as well as the maximum iteration number of advanced optimization algorithms were changed in the training dataset for creating different scenarios. The detailed performance assessment of logistic regression, artificial neural networks, support vector machine, and decision tree approaches was conducted by taking training time, the prediction accuracy of training and test set into consideration. According to the obtained results, it should be highlighted that the SVM algorithm presents a promising solution in load identification, with 99% accurate load classification using test dataset. Since this approach requires a rather long training time, the bagged tree ensemble algorithm is chosen instead of SVM algorithm during the development of the energy disaggregation system. The tests on ThingSpeak platform revealed the applicability of the proposed low cost approach.

## ACKNOWLEDGMENT

J.P.S. Catalão acknowledges the support by FEDER funds through COMPETE 2020 and by Portuguese funds through FCT, under POCI-01-0145-FEDER-029803 (02/SAICT/2017).

## References


[1] J. Hussein, T. Jiashen, I. Dahaman, and A. Hamza, “Impacts of demand-side management on electrical power systems: A review,” Energies, vol. 11, pp. 1-19, May 2018.

[2] OECD, “Energy,” OECD Green Growth Studies, OECD Publishing, Paris, 2012

[3] R. A. Verzijlbergh, L. J. De Vries, G. P. J. Dijkema, and P.M. Herder, “Institutional challenges caused by the integration of renewable energy sources in the European electricity sector,” Renewable and Sustainable Energy Reviews, vol. 75, pp. 660-667, Aug. 2017.

[4] G. Flett, and N. Kelly, “A disaggregated, probabilistic, high resolution method for assessment of domestic occupancy and electrical demand,” Energy and Buildings, vol. 140, pp. 171-187, April 2017.

[5] M. Li, S. Han, and J. Shi, “An enhanced ISODATA algorithm for recognizing multiple electric appliances from the aggregated power consumption dataset,” Energy and Buildings, vol. 140, pp. 305-316, April 2017.

[6] Y. Zhang, G. Yang, and S. Ma, “Non-intrusive load monitoring based on convolutional neural network with differential input,” Procedia CIRP, vol. 83, pp. 670-674. 2019.

[7] G. W. Hart, "Nonintrusive appliance load monitoring", Proceedings of the IEEE, vol. 80, pp. 1870–1891, 1992.

[8] S. Lin, L. Zhao, F. Li, Q. Liu, D. Li, and Y. Fu “A nonintrusive load identification method for residential applications based on quadratic programming,” Electric Power System Research, vol. 133, pp. 241-248, April 2016.

[9] J. M. Gillis, S. M. Alshareef, and W. G. Morsi, “Nonintrusive load monitoring using wavelet design and machine learning,” IEEE Trans. Smart Grid, vol. 7, pp. 320 – 328, Jan. 2016.

[10] N. F. Esa, M. P. Abdullah, and M. Y. Hassan, "A review disaggregation method in non-intrusive appliance load monitoring," Renewable and Sustainable Energy Reviews, vol. 66, pp. 163–173, Dec. 2016.

[11] L. Hiu, “Non-intrusive load monitoring: Theory, technologies and applications”, Springer, 2020, pp. 13-15.

[12] R. L. Sawyer, J. M. Anderson, E. L. Foulks, J. O. Troxler and R. W. Cox, "Creating low-cost energy-management systems for homes using non-intrusive energy monitoring devices," 2009 IEEE Energy Conversion Congress and Exposition, San Jose, CA, 2009, pp. 3239-3246.

[13] A. Adabi, P. Manovi, and P. Mantey, "Cost-effective instrumentation via NILM to support a residential energy management system," 2016 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, 2016, pp. 107-110.

[14] G. Srinivasan, C. Anandan, S. A. K. Jain, S. S. Ahmed, and V. Vijayaraghavan, "Low-cost non-intrusive device identification system," 2016 IEEE Dallas Circuits and Systems Conference (DCAS), Arlington, TX, 2016, pp. 1-4.

[15] M. Nardello, M. Rossi, and D. Brunelli, "A low-cost smart sensor for non intrusive load monitoring applications," 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), Edinburgh, 2017, pp. 1362-1368.

[16] S. Khan, A. F. Latif, and S. Sohaib, "Low-cost real-time non-intrusive appliance identification and controlling through machine learning algorithm," 2018 International Symposium on Consumer Technologies (ISCT), St. Petersburg, 2018, pp. 32-36.

[17] T. Le, H. Kang, and H. Kim, "Household appliance classification using lower odd-numbered harmonics and the bagging decision tree," IEEE Access, vol. 8, pp. 55937-55952, March 2020.