A Benchmark Framework to Evaluate Energy Disaggregation Solutions[*]

Nikolaos Symeonidis, Christoforos Nalmpantis, and Dimitris Vrakas

School of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece
{symeonidn, christofn, dvrakas}@csd.auth.gr
https://www.csd.auth.gr/en/

Abstract. Energy Disaggregation is the task of decomposing a single meter aggregate energy reading into its appliance level subcomponents. The recent growth of interest in this field has lead to development of many different techniques, among which Artificial Neural Networks have shown remarkable results. In this paper we propose a categorization of experiments that should serve as a benchmark, along with a baseline of results, to efficiently evaluate the most important aspects for this task. Furthermore, using this benchmark we investigate the application of Stacking on five popular ANNs. The models are compared on three metrics and show that Stacking can help improve or ensure performance in certain cases, especially on 2-state devices.

Keywords: NILM · Energy Disaggregation · Artificial Neural Networks · Stacked Learning · Benchmark.

1 Introduction

In the modern world, one of the biggest problems humanity is facing is the inadequate management of electrical energy. Overconsumption causes a series of negative phenomena that have both economic and ecological impact, at individual and mass scale. Numerous factors affect this problem, such as the dramatic increase in usage of electrical devices in recent decades, as well as the global population growth. It is apparent that extensive research is required to have better methods of controlling electrical energy consumption. The existence and application of smart meters already contribute in that regard. Energy disaggregation can make further use of those, to enhance load monitoring capabilities.

Energy disaggregation, also known as Non-intrusive load monitoring (NILM), is the task of decomposing an aggregate energy signal into its sub-components, i.e identifying the individual appliances signal from the whole energy consumption of a house. It was first introduced by George Hart [4]. By applying NILM

[*] This work has been funded by the ΕΣΠΑ (2014-2020) Erevno-Dimiourgo-Kainotomo 2018/EPAnEK Program 'Energy Controlling Voice Enabled Intelligent Smart Home Ecosystem', General Secretariat for Research and Technology, Ministry of Education, Research and Religious Affairs.

methods load monitoring becomes easier and less costly, due to the requirement of only a single meter from which the device level signals can be extracted, in contrast to ILM (intrusive load monitoring) methods where multiple meters are needed per house. Therefore ILM has much bigger economical cost and increased difficulty from installing and configuring the meters, its only advantage being its guaranteed higher accuracy.

Many researchers focus on finding solutions where a model is trained per appliance, having as input the whole house energy data and as output the appliance consumption. Each work has its own test cases defined and investigates a set of metrics upon them. Due to this, many different possible scenarios are created, which complicates the comparison between the proposed and existing solutions. There is a lack of structure to follow when conducting the evaluation of a new model. This creates a necessity for a well-defined set of experiments. A set like this should include a variety of scenarios that cover a wide range of aspects and goals for the model under evaluation. Each scenario is tied to a specific purpose, that reveals if the model is suitable for the case.

In this paper we propose a benchmark framework for the evaluation of NILM solutions. Also five ANNs are combined with the method of Stacking and then evaluated with the proposed framework. This paper is structured as follows: in section 2 some of the most popular neural network solutions, that are important for this study, are reviewed. In section 3 the proposed categorization of experiments is described. Section 4 expands upon the method of Stacking. Section 5 presents the most important results produced from Stacked learning. Section 6 includes conclusions from the experiments and discussion for future work. The implementation of Stacking and the five used ANNs, a detailed spreadsheet containing the defined experiments for each category, as well as baseline results can be found in the following repository https://github.com/symeonick15/NILM-Stacking.

2 Related Work

There have been many approaches to solving this problem, among which Machine Learning (ML) has taken the lead of research in recent years. ML methods exhibit great generalization capability on unseen environments, without the need of prior information. Initially, Hart proposed a combinatorial optimization method, which suffered from working only with devices that had a finite number of states. Later Factorial Hidden Markov Models (FHMM) have become quite popular and many developed techniques were based on them [7,15,1,19].

The study around NILM has recently turned towards implementing solutions based on artificial neural networks. Deep and convolutional neural networks have become dominant in many fields like Computer Vision [9] and Natural Language Processing [3], and have been used successfully in many problems that include time series data. Their ability to extract features and handling complex data has lead researchers to start developing NILM solutions based on them, outperforming former approaches [14,5,11,12,18,10,2].

Kelly and Knottenbelt [5] developed three ANN architectures for the task of Energy Disaggregation. The first was a Denoising Auto Encoder that handles the aggregate signal as a “noisy” series, which filters out the noise (i.e. signals from other devices) to extract the target appliance consumption. The 2nd was a Recurrent Neural Network that used LSTM units (long short-term memory) and had the ability to “remember” previously given inputs to use them for prediction. The last architecture would find the Start time, End time and mean consumption of the first activation in a given window. All three were trained on the UK-DALE dataset, having as input the whole-house aggregate signal and as target the appliance consumption. An FHMM approach and Hart’s combinatorial optimization algorithm were also used for the same experiments. In comparison, ANNs outperformed these two methods.

In another study, Mauch and Yang [12] implemented a different architecture with LSTM units for the Energy Disaggregation task. The proposed deep recurrent network had as input the aggregate energy value at a specific timestamp and as output the appliance consumption at the same timestamp. Among the goals of this network was to automatically extract features from low-frequency data and to generalize well on unseen buildings. REDD dataset was used for train and prediction, for two ON/OFF and one multistate devices. Results were promising both on seen and unseen buildings.

Zhang et al. [18] proposed a different deep convolutional architecture that they named sequence to point. The method took its name from the main idea to predict the value of a single time point, based on a sequence of values in the input that has the time point at its midpoint. The input window used had 600 samples (1 hour). The network achieved state of the art results and had great representation power, as it would base its predictions both on the past and the future of a time point.

Krystalakos et al. [10] used Gated Recurrent Units during their experiments to improve current LSTM architectures. To decrease the computational cost and memory demands, LSTM neurons were replaced with GRU, fewer neurons were used and dropout layers were added. Furthermore, a sliding window approach was tested. The network was trained and tested on UK-DALE and compared directly to implementations based on previous architectures, among which a modified seq2point more suitable for online prediction (smaller input windows). The proposed architecture showed promising results, especially on multi-state devices, and it had the same or better results than LSTM while being lighter.

Although there wasn’t a benchmark that was actually followed by these studies to base their experiments, many have used the same or similar structure presented in the work of Kelly and Knottenbelt [5]. Also, most had similarities in their tests, such as having the trained model predicting in an unseen house, aiming to evaluate specific aspects of the models.

In a review of ML approaches for NILM by Nalmpantis and Vrakas [13], it is stated that an objective and direct comparison between different methods is very difficult. The reason is that there are various metrics, many available datasets, different criterias and a variety of methodologies that one can choose when evaluating NILM solutions. A qualitative and a quantitative analysis is presented with methods for evaluation. Among these are Zeifman's requirements [17], as well "generalization" and "privacy" requirements, along with metrics for each. Also a measure of disaggregation complexity is defined, to evaluate the different environments.

3 Purpose

In this paper, we present a categorization of experiments with the purpose to have a unified way of comparing models during research. The scenarios of the proposed categorization include specific train and test cases, each with an explained purpose that explores an aspect of the model, plus a basic set of metrics to evaluate the performance, as both classification and regression. Also, as a reference point it can be further expanded with more cases or have its existing modified, to accommodate additional scenarios, should it be required in a future research, where a more specific goal is examined (e.g. commercial buildings). The proposed benchmark is defined on a set of five appliances, however, the scenarios included can easily be generalized for any device, or even different datasets.

Moreover, the effects of combining several neural networks with the method of Stacking are investigated, by comparing them using the aforementioned taxonomy. Stacking has been used widely in Machine Learning approaches to combine many different models in order to achieve better results than each model individually. The basic idea is to follow this method, in hope of improving the performance of existing neural network architectures, and to see the actual impact it has on them.

4 Taxonomy of Experiments

In this section, the proposed categorization of experiments is described, for usage in the evaluation of new Energy Disaggregation models. Specifically, this method is suitable for evaluating techniques (e.g. ANNs) that take an aggregate signal as input and predict a specific appliance consumption. There are four main categories of experiments described for this method and the purpose that each serves. Although these can be expanded and/or modified to suit the goals of individual studies, we also proceed to define the datasets and appliances used in this study.

The two datasets used in this study are Reference Energy Disaggregation Data Set (REDD) [8] and UK-DALE [6]. Both datasets are freely available, support low-frequency data, refer to domestic buildings and have several houses and appliances for testing. They are also two of the most popular datasets in the field of NILM. The target appliances are: fridge, kettle, microwave, washing machine, and dishwasher. These have been used by many researchers because they cover a wide range of appliance types a house may include. For example, the kettle is a simple ON/OFF device, while the dishwasher is multistate with more complex behavior. Furthermore, they constitute most of a building's consumption and appear in most of the houses making a better target for evaluation.

4.1 Category 1: Single Building NILM

This is actually the most simple form of experiment, where training and prediction happen on the data of the same building, at different time ranges. This evaluates the performance of a model on an environment very similar to the one it was trained on. This is a vital prerequisite for a technique. If it has poor results here, it probably won't do better at other experiments as well. Of course, no house stays the same in reality, but that is also a part of the problem. The specific experiment defined in this study for this category includes training and testing on house 1 of UK-DALE. The test set is defined as the year following April 2016, while the rest of the data are available for training. House 1 has the most available data, thus making it suitable for both training and testing.

4.2 Category 2: Single building learning and generalization on same dataset

Several experiments may be mapped to this category, one per house. Training happens on one house of a selected dataset, while prediction is applied to the rest houses. The purpose of these experiments is to evaluate the ability of the model to generalize on relatively similar unseen houses when trained on data on one house. Although learning is restricted to only one house, this also shows if the trained model is overfitting on its data, which is another aspect to be evaluated. It needs to be noted that houses here are considered similar (e.g. same city/country), but in reality, even neighboring houses may be totally different in their energy patterns (different appliances, consumption patterns, etc.). House 1 of UK-DALE is selected as the training set here again, while the rest of the houses where the target appliance is present compose the test sets. The first house has a good amount of data for learning, while the rest have a few months each, so they are better for prediction.

4.3 Category 3: Multi building learning and generalization on same dataset

In this category, the model is trained with data from more than one, relatively similar houses (same dataset), while prediction is applied on data from a house that was not present during training. The main purpose here is again to evaluate the generalization ability of the method. The second aspect that is being assessed, is the ability of the technique to learn from multiple different sources and combine efficiently the knowledge extracted from them. A model that achieves this will theoretically perform better on new unknown data, due to the variety of data it has learned. This is a similar, but more complex task from the previous category, because a learning algorithm may get "confused", by the variance of the data.

The experiments used for this category are defined for the UK-DALE dataset and follow the same structure as the ones in the study of Kelly and Knottenbelt [5], for tests on unseen buildings. This way a comparison with existing studies becomes easier and more direct.

4.4 Category 4: Generalization to different dataset

By expanding the previous experiment with the prediction on a different dataset, the task becomes even more difficult. The houses on which the model is tested are not similar (e.g. different countries) to the ones it was trained on and may present great differences in characteristics such as included appliances, appliance types, energy grid, consumption behavior, etc. So to perform well the model must possess strong generalization capabilities. The difficulty here can get great and unpredictable. However, it is an interesting query, that evaluates an important ability of a NILM solution. The training set is comprised of UK-DALE data, while testing is applied to REDD data. The first has buildings in the UK, while the second is for buildings in USA. The differences between these two can be apparent, making them a suitable choice for this category.

5 Artificial Neural Networks with Stacking

In this section, we investigate the application of ensemble learning to existing NILM solutions, to further increase their accuracy. The combined model is evaluated using the aforementioned benchmark method, by comparing it with the individual results of the models that were combined.

As recent research suggests that Artificial Neural Networks fare rather well in the task of Energy Disaggregation, 5 such networks have been selected as the base models for the ensemble. The selected networks are Denoising Auto-Encoder (DAE)[5], Recurrent Neural Network (RNN)[5], Short Seq-2-point (SS2P)[18], GRU Network (GRU)[10], GRU with sliding window (WGRU)[10]. All have shown promising results in previous research, where some were better than the others in certain situations. More info about those ANNs can be found in the Related Work section.

The following is a summary of the ANNs architectures. DAE had a convolutional layer, 3 fully connected layers and one more convolutional as the output, with dropout between them. GRU had 2 convolutional, 2 bidirectional GRU and 2 fully connected layers. RNN had 1 convolutional, 2 bidirectional LSTM and 2 fully connected layers. WGRU had 1 convolutional, 2 bidirectional GRU and 2 fully connected layers, with dropout between the last 4. SS2P had 5 convolutional layers and 2 fully connected ones, with dropout between them. More details can be found at the provided code on github.

5.1 Introduction to Stacking

The basic idea behind stacking is combining several different learners, to achieve better results than each of the base models would achieve individually. It is especially efficient when base learners make different errors. Each algorithm learns a part of the problem, while the final combined model is able to learn a greater space. To achieve this, the training set should further be split into 2 separate parts. The first part, usually much bigger than the other, is used to train each of the base models. After each model is trained, the second part is given as an input to each model to get their predictions. The predictions generated are then combined into one matrix that will make up the training input of a different learner (usually simpler), the meta-learner, while the target output is left as is (from the second part). That way the stacked model learns from the predictions of the base models, so it can combine them. After that, the stacked model is ready for prediction. The prediction follows a similar procedure, where each base model is given a copy of the input and then their predictions are given as an input to the meta-model to generate the final prediction. Stacking is especially efficient when base learners make different errors.

5.2 Implementation

The above procedure was implemented as a 2-step method for the stacking experiments. During the first step, each of the neural networks was trained on a part of the training set. Each trained model was also given the second part of the train set and the test set as input for prediction, to generate the train and test set of the stacked model respectively. The predictions were saved as intermediate files to be reused. In the second step, the prediction for the stack train was loaded and aligned on their timestamps. Then they were scaled and used to fit the selected meta-regressor. In the final phase, the predictions on the test set were loaded in the same way and given as input to the meta-model to generate the final predictions.

The sampling ratio for all data was 6 seconds. Only real data were used, with no synthetic data generation. Code is written in Python. The implementation of the used networks was based on a previous study of Krystalakos et al. [10], which were developed using Keras with Tensorflow backend on GPUs. NILMTK was used for loading and preprocessing of data during the base model training and stacking phase. Scikit-learn was used for the Meta-regressors.

The metrics used for the evaluation were F1, Relative Error in Total Energy (RETE) and Mean Absolute Error (MAE).

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{1}$$

$$RETE = \frac{|E' - E|}{max(E', E)} \tag{2}$$

$$MAE = \frac{1}{T} \sum |y'_t - y_t| \tag{3}$$

Where E' is the total predicted energy, E is the total true energy, y'_t is the consumption predicted at time t and y_t is the true consumption at time t.

6 Results and Discussion

Table 1. Fridge, Category 1, Train and test on UK-DALE house 1.

MODEL	F1	RETE	MAE
DAE	0.637	0.224	35.81
GRU	**0.673**	**0.130**	**34.03**
RNN	0.662	0.270	34.69
SS2P	0.651	0.215	35.23
WGRU	0.641	0.224	34.30
AB-3d	**0.674**	0.163	33.54
AB-30	0.635	**0.017**	**26.56**

Table 2. Fridge, Category 3, Train on houses 1, 2, 4 and test on 5 of UK-DALE.

MODEL	F1	RETE	MAE
DAE	0.514	0.176	**47.17**
GRU	nan	0.296	53.87
RNN	nan	0.318	53.92
SS2P	nan	**0.023**	49.18
WGRU	**0.569**	0.243	51.85
DT5	**0.641**	**0.221**	46.71
AB-30	0.525	0.225	**44.43**

In this section, we present some of the most important results that were observed during the experiments. A complete set of results can be found in the supplied repository and was not presented here due to its size. Some results are highlighted to indicate that they were the best among those that were tested (best among base and among stacked). The following are the meta-regressors referenced from the result matrices. Ada Boost with Decision Tree of depth 3, 25 estimators, learning rate 0.1 and 'square' loss (AB-3d). Ada Boost with Decision Tree, 30 estimators and learning rate 0.5 (AB-30). Ada Boost with Decision Tree, 15 estimators and learning rate 0.5 (AB-15). Multi Layer Perceptron with one hidden layer of 100 neurons (MLP). Decision Tree Regressor of depth 5, 15% min split ratio, 9% min leaf ratio (DT5). Simple Decision Tree Regressor (DT). Gradient Boosting Regressor with 25 estimators and learning rate 0.5 (GB).

As it can be seen among the ANNs, some results in F1 score are 'nan'. In those cases the predictions of the model were never above the activation threshold, leading to division by zero. This happens mostly on generalization tasks, proving the difficulty of disaggregating the signal of an unseen building. On the other hand stacking seems to have the advantage of overcoming this problem.

Results for Category 1 experiments of fridge are shown at table 1. Among base models, GRU was a clear winner. Stacking with AB-3d mostly improved F1, while AB-30 had very good RETE and MAE. AB-3d had short trees, so it could not be as accurate, but managed to hit the activation threshold better. AB-30 has predictions closer to the ground truth values, as can be seen from the figure 1, but seems a bit "unstable" (many spikes where it should have continuous values). Maybe by applying a smoothing technique on top of it, it could be further improved. Generally stacking has good results, especially with AB-30, which enhances regression efficiency.

Table 2 shows the results of category 3 experiments for fridge. Here the best RETE was not improved, however, it's still better than 3 out of 5 of base models. MAE is reduced, while DT5 also has very good F1. As above the short tree (DT5) is better suited for classification, while also is less prone to overfitting. In general, tree-based models seem to work better with the fridge, probably due to the simple, repetitive nature of its time series.

Table 3. Kettle, Category 1, Train and test on UK-DALE house 1.

MODEL	F1	RETE	MAE
DAE	0.496	0.250	15.47
GRU	0.301	0.536	32.00
RNN	0.304	0.461	28.12
SS2P	0.322	**0.110**	19.95
WGRU	**0.582**	0.192	**10.78**
DT	**0.740**	**0.130**	**8.11**
AB-15	**0.804**	**0.161**	**6.37**

Table 4. Kettle, Category 3, Train on houses 1, 2, 3, 4 and test on 5 of UK-DALE.

MODEL	F1	RETE	MAE
DAE	0.504	**0.055**	10.71
GRU	0.104	0.383	24.48
RNN	0.250	0.419	43.58
SS2P	nan	0.593	11.02
WGRU	**0.663**	0.122	**10.16**
DT	0.566	0.098	**9.98**
GB	0.583	**0.033**	10.44

Table 3 has the results of Category 1 experiments for kettle. F1 is greatly increased with the application of stacking here, especially with AB-15 which also greatly improves MAE metric and has better RETE than 3/5 of base models. However even best RETE from stacking was worse than best from base models. Again trees were the best combiners. In category 2 though, stacking did not do so well. Probably because kettle is a relatively simple device and stacking forces the final model to focus even more on house 1, it is easier to overfit.

Results of category 3 experiments for kettle are shown at table 4. Among neural networks DAE and WGRU had the best results. With stacking, GB managed to have the best RETE, about the same MAE and 2nd best F1 after WGRU. It was a fine point between the strong points of the best base models. DT was similar, with little worse on F1 and RETE, but the best MAE overall. A simple tree is again among the best, because of the simplicity of the device's behaviour. Also boosting is susceptible to overfitting, risking generalization capabilities.

Table 5. Dishwasher, Category 1, Train and test on UK-DALE house 1.

MODEL	F1	RETE	MAE
DAE	0.109	**0.001**	43.61
GRU	0.471	0.140	37.06
RNN	0.467	0.072	38.54
SS2P	**0.550**	0.047	**31.01**
WGRU	0.468	0.332	31.22
MLP	**0.571**	0.195	29.46

Table 6. Washing machine, Category 3, Train on houses 1,5 and test on 2 of UK-DALE.

MODEL	F1	RETE	MAE
DAE	0.128	**0.566**	28.18
GRU	0.156	0.601	32.05
RNN	nan	0.679	33.21
SS2P	0.174	0.745	40.27
WGRU	**0.302**	**0.568**	**10.55**
AB-30	**0.324**	0.136	12.07

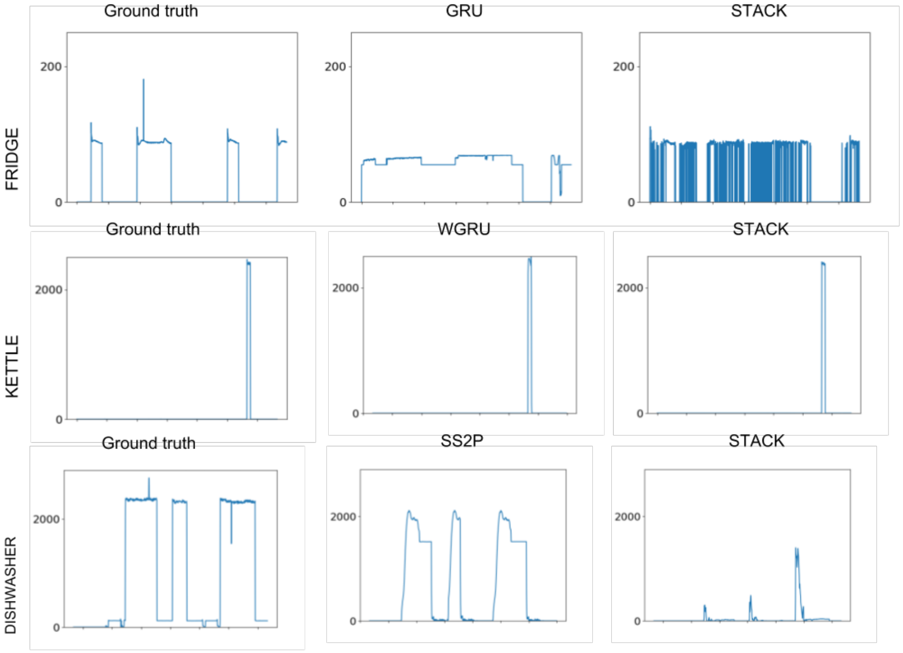

Fig. 1. Sample signal plots including: original, predicted from ANN, predicted from Stacking.

Dishwasher category 1 results can be found at table 5. Stacking here seems to fail at the score of RETE. On the other hand MLP improves the best F1 and the best MAE. AB-3d had an even better MAE, but lost even more on the metrics of F1 and RETE. This time it was not trees, but MLP with 100 neurons that had the best results, suggesting that a more complex model is required for devices such as dishwasher, which have multiple states and complicated behaviour that is time-dependent. Category 2 experiments of the dishwasher had mixed results: F1 would be improved, while the other metrics varied, depending on the house and meta-regressor. In Categories 3 and 4 stacking did not succeed, indicating the requirement of a more appropriate technique.

Results of washing machine category 3 are shown at table 6. Among the experiments of washing machine, some interesting results were found here. Most meta-models had a great improvement of RETE. AB-30 managed to even increase best F1 score, while keeping a near best MAE.

7 Conclusions and Future Work

We have proposed a benchmark to evaluate Energy Disaggregation models and used it to evaluate the application of stacking on existing techniques. Through the experiments, many aspects of the proposed method were explored and stacking showed promising results.

Across the results it appeared that some models were more suited than others in different scenarios. The advantage of stacking was that it could either improve those, or at least find a fine point between them, making it a robust solution. The tested stacked models had good results mostly for disaggregating simple devices (fridge, kettle), especially on same house train-test scenarios. Mainly, Tree based models were the best combiners, while AdaBoosting could further enhance them with the risk of overfitting. This risk was made apparent in generalization experiments (Categories 2-4).

An example scenario that uses stacking could include a weak fast solution that produces online results, which is later combined with the output of other models to produce more accurate final predictions. For generalization tasks improvements seemed less and on complex devices stacking sometimes did not succeed, possibly due to their complicated behaviour that is time dependant, along with the number of states and functions they have. This suggests that other meta-regressors may be more suited, possibly more complicated, time series oriented techniques like Neural Nets, that also have better generalization abilities. Another form of ensemble learning or stacking would also be interesting to test, like meta decision trees [16]. Maybe even another method on top of that could be used to smooth/filter the predicted signal. Regarding the proposed benchmark and the categories of experiments defined, there could also be some other similar scenarios not included here. For example one could have a training set combined from the 2 used datasets (UK-DALE, REDD) and test on both of them or even a third.

References

1. Aiad, M., Lee, P.H.: Non-intrusive load disaggregation with adaptive estimations of devices main power effects and two-way interactions. Energy and Buildings **130**, 131 – 139 (2016). https://doi.org/https://doi.org/10.1016/j.enbuild.2016.08.050, http://www.sciencedirect.com/science/article/pii/S0378778816307472
2. Chen, K., Wang, Q., He, Z., Chen, K., Hu, J., He, J.: Convolutional sequence to sequence non-intrusive load monitoring. The Journal of Engineering **2018**(17), 1860–1864 (2018). https://doi.org/10.1049/joe.2018.8352
3. Graves, A.: Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850 (2013)
4. Hart, G.W.: Nonintrusive appliance load monitoring. Proceedings of the IEEE **80**(12), 1870–1891 (1992)
5. Kelly, J., Knottenbelt, W.: The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. Scientific Data, Volume 2, id. 150007 (2015). **2**, 150007 (Mar 2015). https://doi.org/10.1038/sdata.2015.7
6. Kelly, J., Knottenbelt, W.: The uk-dale dataset, domestic appliance-level electricity demand and whole-house demand from five uk homes. Scientific data **2**, 150007 (2015)

7. Kolter, J.Z., Jaakkola, T.: Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation (6 2018). https://doi.org/10.1184/R1/6603563.v1
8. Kolter, J.Z., Johnson, M.J.: Redd: A public data set for energy disaggregation research. In: Workshop on Data Mining Applications in Sustainability (SIGKDD), San Diego, CA. vol. 25, pp. 59–62 (2011)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
10. Krystalakos, O., Nalmpantis, C., Vrakas, D.: Sliding window approach for online energy disaggregation using artificial neural networks. In: Proceedings of the 10th Hellenic Conference on Artificial Intelligence. pp. 7:1–7:6. SETN '18, ACM, New York, NY, USA (2018). https://doi.org/10.1145/3200947.3201011, http://doi.acm.org/10.1145/3200947.3201011
11. Lange, H., Bergés, M.: The neural energy decoder: energy disaggregation by combining binary subcomponents (2016)
12. Mauch, L., Yang, B.: A new approach for supervised power disaggregation by using a deep recurrent lstm network. In: 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP). pp. 63–67. IEEE (2015)
13. Nalmpantis, C., Vrakas, D.: Machine learning approaches for non-intrusive load monitoring: from qualitative to quantitative comparation. Artificial Intelligence Review pp. 1–27 (2018)
14. Paradiso, F., Paganelli, F., Giuli, D., Capobianco, S.: Context-based energy disaggregation in smart homes. Future Internet **8**(1) (2016). https://doi.org/10.3390/fi8010004, http://www.mdpi.com/1999-5903/8/1/4
15. Parson, O., Ghosh, S., Weal, M., Rogers, A.: Non-intrusive load monitoring using prior models of general appliance types. In: Twenty-Sixth AAAI Conference on Artificial Intelligence (2012)
16. Todorovski, L., Džeroski, S.: Combining classifiers with meta decision trees. Machine learning **50**(3), 223–249 (2003)
17. Zeifman, M.: Disaggregation of home energy display data using probabilistic approach. IEEE Transactions on Consumer Electronics **58**(1), 23–31 (2012)
18. Zhang, C., Zhong, M., Wang, Z., Goddard, N., Sutton, C.: Sequence-to-point learning with neural networks for non-intrusive load monitoring. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
19. Zhong, M., Goddard, N., Sutton, C.: Signal aggregate constraints in additive factorial hmms, with application to energy disaggregation. In: Advances in Neural Information Processing Systems. pp. 3590–3598 (2014)