Imaging Time-Series for NILM*

Lamprini Kyrkou, Christoforos Nalmpantis, and Dimitris Vrakas

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

Abstract. Non Intrusive Load Monitoring is the field that encompasses energy disaggregation and appliance detection. In recent years, Deep Neural Networks have improved the classification performance, using the standard data representation that most datasets provide; that being low-frequency or high-frequency data. In this paper, we explore the NILM problem from the scope of transfer learning. We propose a way of changing the feature space with the use of an image representation of the low-frequency data from UK-Dale and REDD datasets and the pretrained Convolutional Neural Network VGG16. We then train some basic classifiers and use the metric F1 score to test the performance of this representation. Multiple tests are performed to test the adaptability of the models to unseen houses and different datasets. We find that the performance is on par and in some cases outperforms that of popular deep NN algorithms.

Keywords: NILM · Energy Disaggregation · Transfer Learning · Artificial Neural Networks.

1 Introduction

Energy demands have risen greatly in the past 40 years. More and more electrical devices are becoming essential in a household; computers, tablets, cell phones etc. That, of course, means that more energy is being spent. Therefore a need arises for efficient monitoring of the energy being consumed. With the progress of technology, it has become possible to monitor the energy consumption of a house with the use of smart meters. The idea is to apply a smart meter in each appliance of the house and have real-time information about energy consumption. The application of smart meters on the appliance level is still quite costly and therefore other ways have to be explored.

A more cost-efficient way of monitoring power consumption of a house would be to have to install only one meter per household, which will monitor the total energy being consumed. The real world application of this is possible with the

* 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.

help of algorithms that are able to infer the total energy signal to the device level sub-signals that compose it. This is also known as an energy disaggregation task and the field that studies it is Non-Intrusive Load Monitoring (NILM).

The most popular NILM approaches in machine learning are producing models whose input for the training is high or low-frequency data of the mains of a house and the labels are the true energy consumption values of a chosen appliance. Usually, both the training data and labels are chronologically arranged and one of the goals of the chosen model is to find correlations between the aggregated signal and the appliance signal, given a certain time frame. In NILM, appliances within a household often change and get replaced by others as time passes. Therefore, there are many occasions where a model trained to classify an appliance in a certain time frame of a house cannot predict it accurately on a different time frame. That being said, it is worthwhile investigating whether the application of a transfer learning technique can be used to classify electrical appliances (ON/OFF state) given the aggregated power signal of a house.

Transfer learning [15,3] does not rely on training and test data being in the same feature space or even having the same distribution, in contrast to the more classical approaches of machine learning techniques. It can be beneficial in problems where there is a shortage of data, such as an accurate classifier or regressor is not able to be trained, or there is a need for a more general model; a model that generalizes in different distributions. There are several ways transfer learning can be achieved. In this work, we focus on feature representation. The idea is to encode the existing knowledge to another feature space. E.g, encode a time series to an image representation. This approach, to our knowledge, has not been previously explored in the existing literature concerning NILM problems. It has been however applied to other time series classification tasks and has been proven to yield good results

In this study, the idea involves transformation of low-frequency data (1 Hz data), from popular datasets UK-Dale [9]and REDD [11] to images using Wang and Oates' [20] time series to image algorithm, which uses Gramian Angular Field Matrices (GAF), and afterwards changing the feature space with the help of the pretrained Convolutional Neural Network VGG16 image classification algorithm [19] to vectors. The final step is, feeding those vectors, accompanied with their respective device labels, to a classification algorithm, such as a decision tree algorithm, and training it to be able to recognize whether the appliance is ON or OFF.

In the following sections, the process of the feature transformation and transference is analyzed, as well as a set of experiments for the appliance "fridge" of the UK-Dale and REDD datasets is compared directly to previous implementations.

2 Related Work

Hart [6] was the first to work on this problem and used combinatorial techniques in order to monitor changes on the appliances states of a household given the aggregated signal. Since then there have been many approaches to the problem. Factorial Hidden Markov Models (FHMM) solutions [16,1,10,17] have been the leading implementations the past two decades, while Deep learning Artificial Neural Network (ANN) architectures have become popular in the last decade [12,8,2,13,21]. Nalmpantis and Vrakas [14] in their review describe some of the most important recent Machine Learning approaches for the problem of NILM. The approaches are compared in detail, presenting a qualitative and quantitative analysis.

De Baets et al. [4] at their work represent the Vi trajectory of appliances as images and train a Siamese Neural network from which a new feature space is derived. This new representation is the input of the DBSCAN algorithm that ultimately is able to recognize appliances in a household that are left unlabeled. They use the high-frequency data from the datasets PLAID [5] and WHITED [7]. Their approach seems to be successful in recognizing unknown appliances in a household.

Wang et al. [20] propose a novel way of transforming time series into images, in order to test whether this new representation of the data improves classification results. The images are constructed by transforming a time series to its polar coordinate representation. For this the time series used are scaled either to the interval [-1,1] or [0,1]. The rescaled time series $\tilde{X}$ can then be represented as its polar coordinates. This can be achieved by using the value of the time series and the time stamp to encode as the angular cosine and radius respectively. The equation is defined as:

$$\begin{cases} \phi = arccos(\tilde{x_i}, -1 \leq \tilde{x_i} \leq 1, \tilde{x_i} \in \tilde{X}) \\ r = \frac{t_i}{N}, t_i \in N \end{cases} \tag{1}$$

The next step is constructing a Gramian Angular Summation/Difference Fields (GASF/GADF) matrix which contains the correlation between elements of different timestamps. GASF and GADF are defined as follows:

$$GASF = [cos(\phi_i + \phi_j)] = \tilde{X}' \cdot \tilde{X} - \sqrt{I - \tilde{X}^2}' \cdot \sqrt{I - \tilde{X}^2} \tag{2}$$

$$GADF = [cos(\phi_i - \phi_j)] = \sqrt{I - \tilde{X}^2}' \cdot \tilde{X} - \tilde{X}' \cdot \sqrt{I - \tilde{X}^2} \tag{3}$$

In their study they test the above representation on multiple datasets. They use a Tiled Convolutional Neural Network for extracting features. For the classification task they use a Denoising Auto-Encoder. The results are promising as the Mean Squared Error metric is reduced by a maximum of 48% compared to approaches that use raw data.

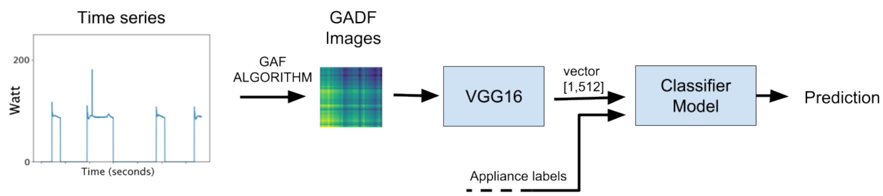

Fig. 1. proposed training/prediction procedure.

3 Implementation

3.1 Data Preprocessing

The process begins with the transformation of the mains aggregated energy signal of a house to multiple vectors of length 64 that will, in turn, become the input of GAF algorithm. In this study only the GADF form is used. The vector size corresponds to 6.4 minutes of data and is chosen so that the time frame is not considerably large, but at the same time is not too small which would cause substantial delays in most machines when producing the image files. Moreover, the time frame chosen cannot be bigger than 6.4 minutes because of computational restrictions. The nilmtk framework for Python 2.7 is used in order to preprocess the data and construct the input files for the GAF algorithm. The set sampling rate for both the mains data and the appliance data is set to 6 seconds. Once the data has been preprocessed the GAF algorithm can begin generating the images. The output of this algorithm are multiple images in .png format and 100x100 pixel size. PAA smoothing is applied on them. The images are also tied with same length vectors of labels for the appliance "fridge", for which the experiments are conducted. The whole procedure can be seen in figure 1.

3.2 Data Extraction

The next step concerns the transference to another input space by using the VGG16 pre-trained model in Keras, Python to categorize the images produced by the GAF algorithm in the previous step. VGG16 is an image classification Convolutional Neural Network (CNN) which is trained for 1000 classes of the ImageNet database [18]. It has been proven very successful on image classification tasks. Its input consists of a 3D tensor that represents the image. Because the model has already been trained there is no need to re-train it with our images. Instead, each image passes through the network and its prediction/ output is the new data space we wanted. Therefore, each image is inferred to 512 categories vectors. After each image for a selected house and time period has been processed, an array of dimensions [numberofimages x 512] is constructed. This will be the input for the training of the classification algorithms that are tested in this study.

3.3 Classification

The labels of each image are averaged and simplified to 1 or 0, depending on whether the averaged number surpasses the threshold of the appliance. If the appliance was on an ON state, during the 6.4 minutes the image represents, the label for it becomes 1. If the appliance was deactivated (and therefore on an OFF state) during the time period of the image, the label for it is 0. Alternatively, one could choose the maximum value of the label vector as the representative point on whether the appliance is in use or not.

For the classification task, the algorithms that were chosen belong to Python's Scikit-learn library. Those are the following: 1) Multi-Layered Perceptron Classifier (MLPC), for which several hidden layer sizes were tested and were concluded to an optimal size of 50. 2) AdaBoost Classifier, with Decision Tree Classifier learners (ABDTC), that each developed to a maximum depth of 2. 1000 estimators were used for the training of this meta-classifier. 3) AdaBoost Classifier (ABC), with its default parameters and 1000 estimators.

Note that some of the above models are trained on UK-Dale's dataset data for house 1, while others for several houses of the REDD dataset. Moreover, the training data and labels are always shuffled before the training of an algorithm begins and a small portion of it is left out as a validation set, while a different set is put aside as the test set. The test set is the one which is used for evaluation of the models, while the validation set is used for fine tuning the models' parameters.

3.4 Evaluation Metrics

The metric that is used to evaluate the models is F1 score:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

F1 score is better than accuracy for evaluating models as it is a function of precision and recall that calculated a balance point of the two. It is a very useful metric for problems that suffer from class imbalance as it can make that apparent.

3.5 Specifications

All experiments can be reproduced by our implementation code in our GitHub repository: https://github.com/LampriniKyrk/Imaging-NILM-time-series. The hardware that is used for all of the model training and testing is an AMD Ryzen 5 1600x processor, an AMD R7 260x GPU and 8GB ddr4 RAM. Note that the

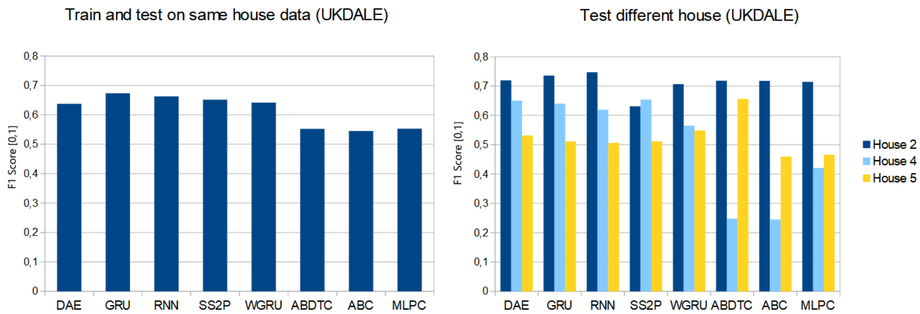

Fig. 2. The bar chart on the left shows the results for all the models that are trained for the data of house 1 of UK-Dale and tested on data from the same house. The bar chart on the right shows the results for the same models as the left but the data tested belong to the unseen houses 2, 4 and 5 of the UK-Dale dataset.

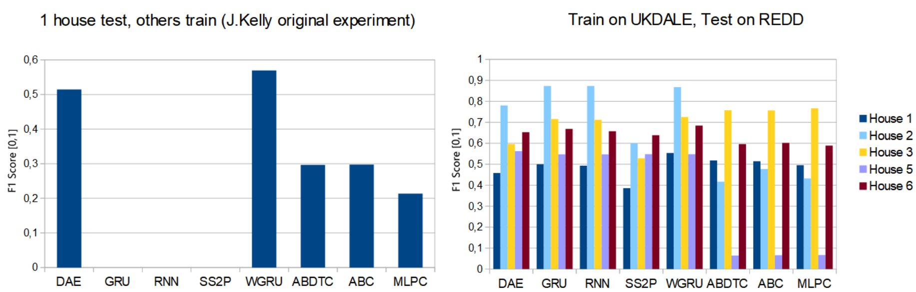

Fig. 3. The bar chart on the left shows the results for all the models that are trained for the data of house 1, 2 and 4 of UK-Dale and tested on data from house 5 of UK-Dale. The bar chart on the right shows the results for the models that are trained for the data of house 1 of UK-Dale dataset and tested for the houses of the REDD dataset.

image production takes a significantly long time, even when running in multiple threads, as the hard disk serves as the bottleneck.

On the experiment section, it is defined in which house and time period data each algorithm is trained on and what the test set for each experiment entitled.

4 Experiments and Results

In this section, we discuss the experiments and results of our novel NILM solution and compare it directly to previous work. The proposed method is compared with some popular ANN architectures as they were implemented in a previous work of Krystalakos et al.[12]. The following networks were used for this task: Gated Recurrent Units Network (GRU)[12], Recurrent Neural Network (RNN)[8], Windowed GRU (WGRU)[12], Short-Sequence2Point (SS2P)[21], and

Denoising Auto-Encoder (DAE)[8]. The comparison between the aforementioned ANN architectures and our approach is direct, meaning the training and testing of the models are done for the exact same time periods and houses.

The ANN architectures that were used as a base for the comparison follow. GRU consists of two convolutional layers, followed by two bidirectional GRU and two fully connected layers. RNN consists of one convolutional layer, followed by two bidirectional LSTM layers and two fully connected layers. WGRU consists of one convolutional layer and two bidirectional GRU layers who are followed by two fully connected layers. The last four layers have dropout between them. SS2P consists of a total of seven layers: five convolutional layers followed by two fully connected layers. All layers of the SS2P architecture have dropout between them. DAE consists of a convolutional layer, three fully connected layers followed by one convolutional output layer. All layers of the DAE architecture have dropout between them.

We train multiple models based on the new feature space on the task of predicting ON/OFF states of the appliance "fridge". In the following sub-sections, the results are presented, categorized by experiment type.

In the first experiments category, all models are being trained on house 1 of the UK-Dale dataset and the dates 1/4/2013 to 1/4/2014. The test also occurs for the data of the same house, though the time frame is 1/4/2016 to 1/4/2017. As it can be seen in figure 2 the proposed approach gives comparable results with those from the ANN models. This experiment uses the F1 metric and evaluates how well the algorithm performs on data of the same house.

The second category of experiments uses the same trained models as above but the test occurs on different houses of the same dataset. In these experiments, we test how well the models generalize on unseen houses of the same dataset. The metric we use for this evaluation is F1 score. Our approach seems to be on par with the comparative models for the houses 2 and 5. Testing on house 5 the AdaBoost classifier with low depth decision trees (ABDTC) seems to outdo all of the ANN models. On house 4 our models are not as accurate as the ANN models, which could be due to the fact that house 1 and house 4 differ greatly based on the number of appliances each house has. House 1 has a total of 54 appliances, while house 4 has only 6.

In the third experiment category, the models are trained with data from multiple houses of the UK-Dale dataset. The houses used for the training data are 1, 2 and 4. From house 1 only the time frame from 1/4/2013 to 1/4/2014 is used for the training, as it has the most data out of all the houses and therefore if all where to be used a memory error could occur. This test evaluates if it is possible for a model to learn from multiple houses and be able to predict accurately on an unseen house. Some ANN models were not able to converge. Our models have a significantly lower F1 score from the best ANN, which is the Windowed GRU model.

The last experiment category tests how well a model trained in UK-Dale generalizes for the data of the REDD dataset. That way, if a model performed well on the unseen houses of the different dataset it is safe to say that it has the potential to be accurate on data regardless of the country of origin. Note that UK-Dale is a UK based dataset while REDD is a US-based dataset. As it is shown in figure 3 most models, whether they're ANN's or the different input space approaches, perform well on most houses. Our approach is on par with the ANN's on most houses and on house 3 it outperforms them.

5 Conclusions and Future work

Transferring of NILM data to a different feature space shows promise as the results seem on par with most of the modern NILM approaches. It certainly has room for improvement as the algorithms we use for classification are not state of the art nor are configured in the best way. Furthermore, the accuracy of the approach could potentially be improved if smaller time windows were to be used (e.g 3 minutes or less), as the problem that arises from bigger time windows is that we don't always get a good estimation of the true appliance state. Moreover, the approach could be explored for regression and multi-label tasks, in addition to the classification approach that is researched in this study. Closing, we believe that continuing this research by combining the new feature space with state of the art algorithms, it is possible to achieve even better results.

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. Dai, W., Chen, Y., Xue, G.R., Yang, Q., Yu, Y.: Translated learning: Transfer learning across different feature spaces. In: Advances in neural information processing systems. pp. 353–360 (2009)
4. De Baets, L., Develder, C., Dhaene, T., Deschrijver, D.: Detection of unidentified appliances in non-intrusive load monitoring using siamese neural networks. International Journal of Electrical Power & Energy Systems **104**, 645–653 (2019)
5. Gao, J., Giri, S., Kara, E.C., Bergés, M.: Plaid: a public dataset of high-resoultion electrical appliance measurements for load identification research: demo abstract. In: proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings. pp. 198–199. ACM (2014)
6. Hart, G.W.: Nonintrusive appliance load monitoring. Proceedings of the IEEE **80**(12), 1870–1891 (1992)
7. Kahl, M., Haq, A.U., Kriechbaumer, T., Jacobsen, H.A.: Whited-a worldwide household and industry transient energy data set. In: 3rd International Workshop on Non-Intrusive Load Monitoring (2016)
8. 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

9. 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)
10. 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
11. 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)
12. 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
13. Lange, H., Bergés, M.: The neural energy decoder: energy disaggregation by combining binary subcomponents (2016)
14. Nalmpantis, C., Vrakas, D.: Machine learning approaches for non-intrusive load monitoring: from qualitative to quantitative comparation. Artificial Intelligence Review pp. 1–27 (2018)
15. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10), 1345–1359 (2010)
16. 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
17. 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)
18. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
20. Wang, Z., Oates, T.: Imaging time-series to improve classification and imputation. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)
21. 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)