Designing a Machine Learning based Non-intrusive Load Monitoring Classifier

Leo Ogrizek[1], **Blaz Bertalanic**[1,2], **Gregor Cerar**[1], **Marko Meza**[2], **Carolina Fortuna**[1]

[1]*Jozef Stefan Institute, Ljubljana, Slovenia*
[2]*Faculty of Electrical Engineering, University of Ljubljana, Slovenia*
E-mail: lo7909@student.uni-lj.si

Abstract. Non-Intrusive load monitoring provides the users with detailed information about the electricity consumption of their appliances and gives energy providers a better insight about the usage of their clients. It can also be used in improving care of elderly, legal services and optimizing energy consumption. While there is plenty of work in NILM appliance classification, in this paper we investigate the design tradeoffs in the process of developing a machine learning based classification model from the perspective of feature engineering, model selection and optimisation. Our work shows that well engineered features have a greater impact on model performance than the selection of the machine learning technique. According to the results, the improvement in f1 score between non-engineered and the proposed engineered features is up to 42% while improvement between the worst non-optimised model and the best optimised one is 19%.

1 Introduction

Non-Intrusive load monitoring (NILM) is a process of estimating the electricity consumption of individual appliances in a household from their combined electricity consumption. By providing detailed information to the user, NILM can help reduce energy usage by 5-15%[1]. It is also beneficial for the energy providers as the additional information allows them to tune the production of electricity and adjust the pricing plans better.

Data about appliance usage has additional applications outside of energetics. It provides an insight into daily activities of residents which can be used in various fields including health and medical (discovering sleep disorders, remotely monitoring elderly), legal (monitoring curfews) and commercial (customer profiling). The benefit of measuring electricity consumption on a single point rather than on every device is ease of deployment and lower installation and maintenance costs.

For application such as elderly care and curfews, events triggered by the residents, for example a toaster being used, need to be automatically detected and distinguished from automatic events such as for example air conditioning being activated by a thermostat. When unusual, possibly anomalous patterns in the usage are detected, such as decreased home activity, then an alert needs to be sent out to their caretakers, allowing them to intervene sooner. This is possible through the use of NILM classification [2]. In a similar manner, when anomalous behaviour is detected from people under house arrest, an alert can be dispatched to the appropriate authorities automatically.

NILM can also be used for predicting trends in energy consumption. Having a future view allows energy producers to prepare and tune the production in accordance to the projected requirements. Accurate consumption predictions reduce excess energy production and in turn, reduce the negative effects on the environment. NILM by itself can reduce power consumption by increasing awareness, some NILM powered systems can also turn off unneeded devices [3, 1]. Inside small communities NILM allows for better implementation of smart energy distribution systems and peer-to-peer energy trading, by providing a local producer information about whether energy is needed or can be returned into the grid [4].

Machine learning (ML) techniques have been shown to be suitable for addressing all NILM related problems [5]. This is realized by formulating a NILM classification problem to be solved and selecting high quality data that can be used by the ML techniques to discriminate between the target classes. The data, ML techniques, the training and evaluation process result in a model able to distinguish the target classes based on previously learned patterns in the data.

In this paper we aim to understand the design tradeoffs for developing a NILM classifier. The contributions of this paper are as follows:

- We systematically analyze the performance of feature engineering, model selection and optimization for NILM on the UK-Dale dataset.
- We propose a best feature set constructed to capture the shape of the time series and we show it performs by 42 percentage points better than the baseline raw time series.

The paper is organized as follows. Section 2 presents related work, Section 3 formulates the problem and identifies the dataset, Section 4 discusses the feature selection process while Section 5 elaborates on the model selection. Finally, Section 6 concludes the paper.

2 Related Work

In the last decade a large body of work on NILM classification has been published. The traditional way of developing such classifiers used various signal processing technique. For instance, in [6] the authors classify appliances based on transient effects by using FFT transforms on higher frequency measurements.

As the performance of the ML techniques improved, they have been increasingly considered as potential techniques for realising NILM. In [5], the authors provide an overview of the field by analysing suitable features and ML techniques for different types of data, i.e. low frequency, high frequency, etc. They also provide an overview on model selection. In [7] the authors develop a KNN classifier that is trained using U/I trajectories as features. In [8], the authors compare four algorithms for disaggregating appliances using multi label classification.

More recently, authors are using deep learning to solve the NILM classification problem. In [9] low frequency data is used to detect on/off events. Classifications are made based on these events using a deep neural network classifier trained on average power consumption, minimum and maximum time an appliance is powered on. In [10] the authors use transient power signals to train a convolutional neural network to classify devices while in [11] a recurrent neural network is trained on denoised data.

Our work complements [5] by providing a quantitative consideration of the design trade-offs when designing a NILM classifier using classical explainable machine learning models.

3 Problem statement

We define our NILM classification problem as follows. Given an input time series T representing energy measurements from households, there is a function Φ, that maps the time series to a set of target classes C representing different household appliances as in equation 1.

$$C = \Phi(T) \tag{1}$$

where the set of target classes is C = { computer monitor, laptop, television, washer dryer, microwave, boiler, toaster, kettle, fridge }. The assumption is that a disaggregation in individual time series is already performed and the classification is done on the resulting time series. The classifier Φ is realized using classical machine learning techniques and the UK-DALE dataset to develop the model able to discriminate between the classes. We consider the following diverse but explainable set of techniques: SVM, KNN, Random forest (RF), MLP, Logistic regression (LR) from the scikit-learn library version 0.22.2. Deep learning methods were not used due to the limited amount of data in the UK-DALE dataset.

3.1 Dataset summary

The UK DALE (Domestic Appliance-level Electricity) [12] dataset contains 180000 measurements of power usage per device, taken in 6 second intervals or 0.1667 Hz for a total of 300 hours of measurements per device. The devices in the dataset are common household appliances which were already presented in section 3 . Data is spread into 1 hour long segments, each dataset sample contains a time series with 600 data points as depicted in Figure 1. Measurements are disaggregated, taken with a smart plug on every device [12].

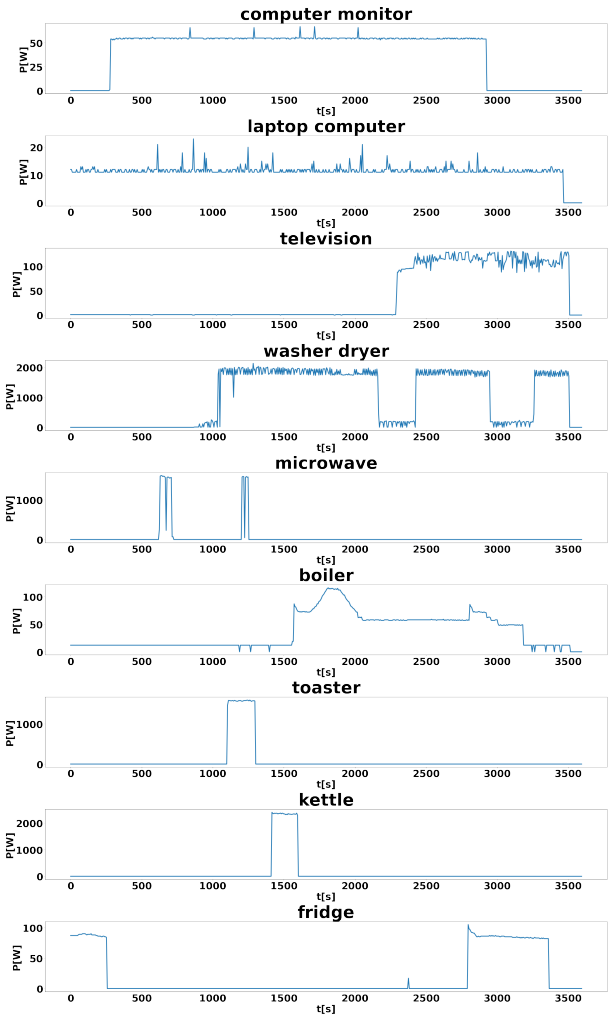

Figure 1: Samples for each appliance, showing power in relation to time over a 1 hour interval.

3.2 Methodology

UK-Dale contains univariate time series that if used raw, may lead to suboptimal model performance. A common practice in the ML community is to construct synthetic features, by statistical summaries, feature interactions, etc. using the raw time series. As some of the considered techniques, such as SVM and Logistic regression are sensitive to scaling, we used standard scaling that ensures that all the features will be relatively proportional.

During the model development process, we use 5-fold cross validation realized using scikit-learn K-fold cross validation. We use the selected ML technique with default parameter configuration as a baseline and then provide an optimized version using the scikit-learn implementation of grid search. For the SVM, grid search

was optimizing the kernel and C value; for K-nearest-neighbours it optimized the number of neighbours; for MLP it optimized the solver, the initial learning rate and the learning rate schedule; for Random forest the number of estimators while for SVM and logistic regression the regularization parameter.

The models were evaluated using standard classification metrics: $precision = \frac{TP}{TP+FP}$, $recall = \frac{TP}{TP+FN}$ and $f1 = \frac{2*precision*recall}{precision+recall}$ for every class and their mean scores for every fold, where TP, FP and FN stand for true positives, false positives and false negatives. The final results presented in this paper are the mean scores of all 5 folds.

4 Feature selection

By manually inspecting the appliances in the dataset, we noticed that they differ in overall power consumption and how constant their energy usage is (see Figure 1). For feature engineering we chose those features, that best capture the shape of the time series. The first created features were the mathematical moments and the squares of their values, minimum, maximum, peak-to-peak, sum and median. These features alone do not describe the energy usage spikes very well, so a second series was created from every sample with the derivatives of all the intervals between measurements. The number of times this derivative changes sign, which will be referred to as nturns, describes how many power spikes a device has during the measurement period. The sum of absolute values of all the derivatives, which will be referred to as d_abs describes the intensity of said spikes. Features were calculated for each sample separately to avoid data leakage between training and testing dataset.

Table 1: Comparison of feature sets using SVM.

Feature set	Precision	Recall	f1
raw	0.486	0.460	0.421
min, max, median	0.746	0.664	0.661
mean, std	0.714	0.646	0.642
all	0.851	0.837	0.835
best	**0.851**	**0.835**	**0.834**

We performed a guided evaluation of the features described in this section with the best results, the raw time series as baseline and standard statistics summarized in Table 1. Using SVM to explore the performance of various feature vectors, our experiments showed that the most useful features are those that describe the shape of the time series. Inputting raw data which makes every timestamp a feature yields an f1 score of 0.421. The results are better than random because of the difference in size of the features within different classes.

Using the minimum, maximum and median value from the time series results in an f1 score of 0.661. These results are significantly better because they describe the shape of the data, rather than assuming a timestamp is relevant to the end results. Using synthetic features, shifting the series would still result in the same features, unlike when raw data is used. Because measurements can be started/finished at any time, this is a better classifier. For the same reason, using the mean and standard deviation also yields better results than raw data: f1 score of 0.642. Using all synthetic features results in an f1 score of 0.835. This is considerably better because the features describe more properties of the time series. However, some of the features are redundant and can be ignored to improve computation time.

The final feature set determined to be the best is *[mean, skew, kurtosis, peak-to-peak, variance, nturns, d_abs and standard deviation]*. This means the baseline feature vector of length 600 was reduced to a feature vector of length 8. These features give almost identical results as using all the synthetic features, with an f1 score of 0.834, but with lower required computing performance.

Table 2: Per class performance, SVM, best feature set

class	precision	recall	f1
computer monitor	0,882	0,720	0,780
laptop computer	0,897	0,800	0,838
television	0,958	0,927	0,941
washer dryer	0,952	0,700	0,804
microwave	0,679	0,710	0,687
boiler	0,945	0,937	0,940
toaster	0,705	0,940	0,806
kettle	0,684	0,806	0,739
fridge	0,961	0,980	0,970

A per class breakdown shown in Table 2 of the best results reveals that the fridge, boiler and television are classified best, while microwave and kettle are classified poorly. This is because the former have very distinct energy consumption patterns (fridge has low consumption and it is usually turned off can be distinguished from small devices by large and long lasting energy consumption spikes, boiler has a low consumption when not in use but consumes a lot of power when turned on, television uses a medium amount of energy with much more emphasised variations in consumption), while the latter share properties between each other and other low consumption devices such as the toaster and monitor (all are powered on for short periods of time, they also share highly variable energy usage).

Certain samples contain devices that are powered off throughout the whole sample, such samples can not be distinguished from one another because they all have the same characteristic of no consumption.

5 Model selection and optimisation

As discussed in Section 3.2 and summarized Table 3 the model performance depends both on the selected ML technique as well as on the parameters. It can be seen from the table that the best performing models are based on MLP and SVM.

For the SVM, the default radial kernel proved to be optimal, however the default C value of 1 was far too

Table 3: Impact of algorithm optimisation

Algorithm	Precision	Recall	f1
SVM default	0.704	0.672	0.647
SVM optimised	**0.851**	**0.835**	**0.834**
KNN default	0.784	0.776	0.772
KNN optimised	0.787	0.776	0.776
MLP default	0.786	0.769	0.766
MLP optimised	**0.849**	**0.830**	**0.828**
LR default	0.742	0.691	0.674
LR optimised	0.788	0.770	0.768
RF default	0.822	0.815	0.813
RF optimised	**0.822**	**0.815**	**0.813**

low, resulting in a separating hyperplane with a larger margin, that misclassifies more points. After optimisation the best C value was determined to be around 38000. This resulted in a significant increase in accuracy with the f1 score going from 0.647 to 0.834 as can be seen from the first two rows of the table.

In K-nearest-neighbours grid search found that checking 4 closest neighbours is better than the default 5, however this resulted in only a very small improvement of f1 from 0.772 to 0.776 as can be seen from rows 3 and 4 of the table.

MLP initially had convergence issues. The first candidates for optimisation were thus the initial learning rate and the learning rate schedule. Different solvers were also tested. After optimisation, the parameters were changed from default adam solver with initial learning rate of 0.001 and constant learning rate to sgd solver with 0.45 initial learning rate and adaptive learning rate. We used one hidden layer with 100 neurons. Optimisation resulted in a large increase in f1, from 0.766 to 0.828 as can be seen from rows 5 and 6 of the table.

Like SVM, logistic regression also suffered from a low C value. After optimisation, it was changed from the default of 1 to 20000. The solver was changed from lbfgs to liblinear. This resulted in a large increase in f1 going from 0.674 to 0.768 as can be seen from rows 7 and 8 of the table.

In random forest the number of estimators was changed from 100 to 500 but that only resulted in a change of f1 from 0.813 to 0.815 as can be seen from rows 9 and 10 of the table.

6 Conclusions

In this paper we showed the tradeoffs in developing an accurate appliance classification system and proposed a new feature set for improved classifier performance.

First, we showed that by using statistical and custom quantities able to capture the shape of the raw time series can improve the F1 score performance by up to 42 percentage points compared to the raw time series baseline. Our experiments showed that the best performing features were mean, skew, kurtosis, peak-to-peak, variance, the number of times the derivative changed sign on the observed interval, sum of absolute values of the derivative on every interval within the time series and standard deviation.

Second, we also showed that the choice of the machine learning technique and the optimal parameters are also important. The best preforming model using optimised SVM has an f1 score 0.187 better than the worst performing non optimized SVM. However, all optimized SVM, MLP and random forest models work well with f1 scores between 0.81 and 0.84 for the considered problem.

References

[1] S. Darby, "THE EFFECTIVENESS OF FEEDBACK ON ENERGY CONSUMPTION," p. 24, 2006.

[2] J. Alcalá, J. Ureña, and A. Hernández, "Activity supervision tool using Non-Intrusive Load Monitoring Systems," Sep. 2015, pp. 1–4, iSSN: 1946-0759.

[3] I. Abubakar, S. N. Khalid, M. W. Mustafa, H. Shareef, and M. Mustapha, "Application of load monitoring in appliances' energy management – A review," *Renewable and Sustainable Energy Reviews*, vol. 67, pp. 235–245, Jan. 2017.

[4] "Smart home energy management systems survey," Nov. 2014, pp. 167–173.

[5] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors*, vol. 12, no. 12, pp. 16 838–16 866, 2012.

[6] S. R. Shaw, S. B. Leeb, L. K. Norford, and R. W. Cox, "Nonintrusive load monitoring and diagnostics in power systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 7, pp. 1445–1454, 2008.

[7] A. Kelati, H. Gaber, J. Plosila, and H. Tenhunen, "Implementation of non-intrusive appliances load monitoring (NIALM) on k-nearest neighbors (k-NN) classifier," *AIMS Electronics and Electrical Engineering*, vol. 4, no. 3, pp. 326–344, 2020.

[8] D. Li and S. Dick, "Whole-house Non-Intrusive Appliance Load Monitoring via multi-label classification," in *2016 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2016, pp. 2749–2755, iSSN: 2161-4407.

[9] M. A. Devlin and B. P. Hayes, "Non-intrusive load monitoring and classification of activities of daily living using residential smart meter data," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 3, pp. 339–348, 2019.

[10] D. d. Paiva Penha and A. R. Garcez Castro, "Home Appliance Identification for Nilm Systems Based on Deep Neural Networks," *IJAIA*, vol. 9, no. 2, pp. 69–80, Mar. 2018. [Online]. Available: http://aircconline.com/ijaia/V9N2/9218ijaia06.pdf

[11] J. Kim, T.-T.-H. Le, and H. Kim, "Nonintrusive Load Monitoring Based on Advanced Deep Learning and Novel Signature," *Computational Intelligence and Neuroscience*, vol. 2017, p. e4216281, Oct. 2017, publisher: Hindawi. [Online]. Available: https://www.hindawi.com/journals/cin/2017/4216281/

[12] J. Kelly and W. Knottenbelt, "The uk-dale dataset, domestic appliance-level electricity demand and whole-house demand from five uk homes," *Scientific data*, vol. 2, no. 1, pp. 1–14, 2015.