On Neural Filter Selection for ON/OFF Classification of Home Appliances

Anže Pirnat and Carolina Fortuna
ap6928@student.uni-lj.si,carolina.fortuna@ijs.si
Jožef Stefan Institute, Ljubljana, Slovenia.

ABSTRACT

Non-intrusive load monitoring (NILM) enables the extraction of appliance-level consumption data from a single metering point. Appliance ON/OFF classification is a particular type of such appliance level data extraction recently enabled by deep learning (DL) techniques. To date, a study on the influence of neural filter selection on the performance and computational complexity for appliance ON/OFF classification is missing. In this paper, we start from a widely used DL architecture, adapt it for the appliance ON/OFF classification problem and then study the influence of the filters on the model performance and model complexity. Through this study we develop a model, PirnatCross, that excels at cross-dataset performance, offering an average improvement in average weighted F1 score of 17.2 percentage points vs a SotA model and VGG11 baseline respectively, when trained on REFIT and evaluated on UK-DALE and vice versa. Also, PirnatCross consumes 6-times less energy compared to a SotA model.

KEYWORDS

non-intrusive load monitoring (NILM), ON/OFF appliance classification, deep learning (DL), convolutional recurrent neural network (CRNN), multi-label classification

1 INTRODUCTION

Mitigating the impact of climate change is an urgent challenge that requires collective action to keep the global average temperature below 1.5°C in relation to pre-industrial levels. Reducing unnecessary electrical energy consumption and consequently limiting electrical energy production is a crucial step towards achieving our goals, as it is estimated that such activities account for over 40% of the total CO_2 equivalent generated by human activities [1]. Beside reducing energy consumption, we are increasingly adopting renewable power plants due to their significantly lower CO_2 emissions compared to fossil fuel-based ones [2]. However, renewable energy resources have a major drawback; dependency on renewable resources which are far less predictable, posing a challenge to the stability of the power system [11]. To address this issue, demand response strategies are being implemented to adjust electricity consumption to better match supply [1]. Consequently, efforts are being made to monitor and manage energy consumption more efficiently in residential buildings, making it relevant to track device activity (ON/OFF events) [3].

To avoid the high cost and invasiveness of monitoring each individual device with an electricity meter, researchers have developed a more economically efficient method known as non-intrusive load monitoring (NILM). This method involves obtaining appliance-level data using just one metering point to measure the total electricity consumption of a household. By using classification techniques for NILM, it is possible to determine the states (ON/OFF) of devices within a household and monitor their activity for demand response applications. As in a typical household it is possible to have several appliances working simultaneously, a suitable approach for determining the activity states of appliances is multi-label classification, where the state of each appliance is used as the class label and the recorded readings from a single household meter serve as input samples. Li *et al.* were among the first to propose multi-label classification for NILM disaggregation. More recently, Tanoni *et al.* [12] employed gated recurrent unit (GRU) in their CRNN for weakly-supervised training, mixing the amount of strongly and weakly labeled data to confirm the effectivness of such approach. Also Zhou *et al.* [14] proposed a new model called TTRNet, which uses a transpose convolution before a recurrent layer, a method, which has also shown better results in other works [8]. The existing works based on DL techniques typically lack a DL computational complexity/energy consumption analysis that is relevant in designing such models [2]. For instance, in [5] they analyzed the carbon footprint of various architectures and concluded that convolutional layers are power hungry because they operate in three dimensions, unlike fully connected layers which operate in two dimensions.

Existing studies typically develop and evaluate their method on a only a few datasets that are often limited in size. For instance [12] relied on two publicly available datasets and developed and evaluated a model for each of the two: REFIT [9] and UK-DALE [6]. While this approach is appropriate for relative method performance assessment, some studies have discussed also the importance of cross-dataset evaluation. For example, Han *et al.* [4] described significant dataset biases and high class imbalance of in-the-wild datasets as a fundamental bottleneck in facial expression recognition. Their results showed that cross-dataset evaluation can reduce dataset bias and improve the performance.

In this paper we aim to better understand the influence of the filters on the model performance and model complexity for multi-label ON/OFF appliance classification through intra and cross-dataset evaluation. Our main contributions are as follows:

- We adapt VGG19, a widely used DL architecture, for the appliance ON/OFF classification and study the influence of the filters on the model performance and model complexity.
- We develop a model, PirnatCross, that excels at cross-dataset performance, offering an average improvement of 17.2 percentage points vs a SotA model and VGG11 baseline respectively, when trained on REFIT and evaluated on UK-DALE and vice versa. Also, PirnatCross consumes 6-times less energy compared to SotA model.

[1] tinyurl.com/CO2-from-electricity1
[2] tinyurl.com/renewable-energy-doubled

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2023, 9–13 October 2023, Ljubljana, Slovenia
© 2023 Copyright held by the owner/author(s).

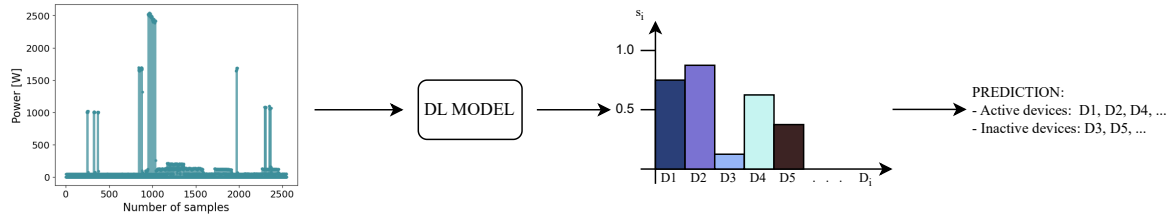

Figure 1: We input the data measured from a household into the DL model and it outputs s_i for each device present in the experiment. If s_i is greater than 0.5 we classify the device as active, if not as inactive.

The paper is organized as follows. Section 2 provides the problem statement, Section 3 presents methodological details, while Section 4 analyses the results of our study. Finally, Section 5 concludes the paper.

2 PROBLEM STATEMENT

Given an input power consumption measured by a smart meter $p(w)$ over a time window w, we aim to develop a multi-label ON/OFF classifier Φ that maps the input to a probability vector $s(w)$ corresponding to the status of the home appliances as:

$$s(w) = \Phi(p(w)) \tag{1}$$

The $|s|$ of the set s, indicates the number of appliances to be recognised. For each window of measurements $p(w)$ input to the model Φ, $s(w)$ will be of the form $[s_1(w), s_2(w), ..., s_N(w)]$, $s_i \in [0, 1]$ and $N = |s|$ where each s_i estimates the probability of appliance d_i to be active as also depicted in Figure 1. When $s_i > 0.5$ the appliance will be classified as ON, otherwise it will be classified as OFF. More than one appliance can be ON at the same time, therefor s contains multiple labels assigned to the current instance. In this paper $N = 5$ in total of which any 1-4 can be active.

The ON/OFF classifier Φ realized as a deep learning network is typically composed of a set of layers $[l_1, l_2, ...l_M]$ where the types of the layers may vary depending on how the respective architecture is designed. For instance $l_i \in [FC, Pool, Conv, GRU, ...]$, where FC stands for fully connected, Pool stands for pooling, Conv for convolutional and GRU for gated recurrent unit. As has been already shown also in [10], the computational complexity varies across the types of the layers.

In developing Φ, we start from the VGG family of architectures as they are widely used in various communities and have already shown promising results for classification on NILM [7]. More precisely we consider VGG19 comprising of 19 layers with trainable parameters, 16 of which are convolutional and 3 are fully connected. The convolutional layers are grouped into five blocks:

- Block 1: 2 x conv. with 64 filters + Max pooling
- Block 2: 2 x conv. with 128 filters + Max pooling
- Block 3: 4 x conv. with 256 filters + Max pooling
- Block 4: 4 x conv. with 512 filters + Max pooling
- Block 5: 4 x conv. with 512 filters + Max pooling

This architecture has been tailored to accommodate time series data, replacing the 2D convolutions and pooling from VGG19, designed for images, with 1D counterparts that are more suitable for time-series. In addition, the convolutional layers in the 5th block have been replaced with transpose convolutional layers to increase the temporal resolution of features to reduce their number as suggested in [14]. We also integrated a recurrent layer after the 5th block, GRU layer to be specific, as it is able to model temporal relationships in the time series and it was shown to achieve good performance in a recent study [12].

In order to estimate the computational complexity of the resulting architecture, referred to as PirnatCross, we must first calculate its complexity as the sum of all floating point operations (FLOPs) that have to be computed for each of its layers. This can be calculated for convolutional, pooling and fully-connected layers with the equations from [10] and for GRU with equation from [13].

As convolutional layers dominate in our adaptation of VGG19, and the computational complexity of a convolutional layer is relatively high compared to other type of layers [10]. Generally, the number of FLOPs used throughout the convolutional layer F_c is equal to the number of filters N_f times the flops per filter $F_c = (F_{pf} + N_{ipf})N_f$. Therefore we aim to study the influence of the number of the filters N_f on the model performance and complexity. Let the starting number of filters in each block of the adapted architecture be the same as in the original VGG19, namely $F = [64, 128, 256, 512, 512]$, analyze the model performance as average F1 score versus computational complexity in FLOPs.

3 METHODOLOGY

This section provides methodological details related to the datasets, the training approach and evaluation process that were employed for the study.

3.1 Datasets

The study is conducted using two datasets: UK-DALE [6] and REFIT [9]. Within each dataset, we monitor the same five appliances d_i that were also used in recent research [12]: fridge, washing machine, dishwasher, microwave, and kettle. The data from the selected devices is obtained and processed using the procedure described by Tanoni *et al.* [12] to form 2 mixed datasets. After processing, the two mixed datasets each consist of the same five devices, with each sample containing a random selection of one to four active devices. Samples with varying numbers of active devices are randomly distributed throughout the datasets. We evaluate the cross-dataset performance of models on two mixed datasets obtained by processing data from, UK-DALE and REFIT, in both directions. Specifically, we train models on REFIT derived dataset and test them on UK-DALE derived datasetand vice versa, by training on UK-DALE derived dataset and testing on REFIT derived dataset.

3.2 Benchmarks

In order to have a more meaningful study, we also evaluate PirnatCross, the adapted VGG19, against a VGG11 baseline and a recently published work TanoniCRNN [12]. For VGG11, we used a learning rate of 0.0001 and the same batch size and epochs. For

TanoniCRNN, we used the hyperparameters specified as optimal in their paper [12].

For PirnatCross we vary the set of the filters F by multiplying with $k \in [0.02, 0.04, 0.06, 0.08, 0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 1.9, 2.1, 2.3, 2.5]$. The learning rate, batch size, and number of epochs were determined through a process of trial and error, informed by previous experiments, and subsequently fine-tuned for each model, to optimize model performance and stability. The resulting values are: learning rate of 0.0003, a batch size of 128, and trained for 20 epochs.

While some models were capable of handling larger batch sizes, we found that performance was not improved by increasing the batch size beyond 128, so we kept it unchanged for all models. We train and evaluate using 5-fold cross-validation.

3.3 Metrics

We use the average weighted F1 score ($\overline{F1score_w}$) as a performance metric because our datasets are not balanced and do not provide equal representation for each device.

$$\overline{F1score_w} = \sum_{i=1}^{N_d} F1score_i \times Weight_i \tag{2}$$

The average weighted F1 score is calculated using three metrics: true positive (TP), false positive (FP), and false negative (FN). TP measures the instances where the device is accurately classified as active, while FP represents cases where the device is erroneously classified as active. FN indicates instances where the device is mistakenly classified as inactive.

From these metrics, we derive the precision ($Precision = \frac{TP}{TP+FP}$) and recall ($Recall = \frac{TP}{TP+FN}$), which are used to calculate the F1 score ($F1score = 2 \times \frac{Precision \times Recall}{Precision+Recall}$). To obtain the average weighted F1 score (2), we first compute the F1 score for each device, then take the average based on their weight ($Weight = \frac{SSD}{SAD}$), which is determined by the support for the specified device (SSD) and the support of all devices (SAD).

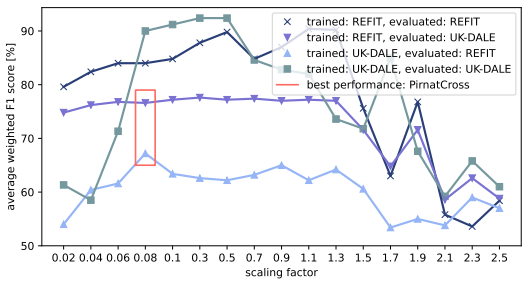

Figure 2: Average F1 scores on intra and cross-dataset training and evaluation as a function of filter scaling factor.

4 RESULTS

In this section we first determine the optimal filter configuration for variations of the PirnatCross architecture to achieve high average weighted F1 score. We then follow with a computational complexity and carbon footprint assessment. Finally, we then benchmark the performance of models in cross-dataset evaluation on REFIT and UK-DALE datasets.

4.1 Analysis of Tuning the Filters in PirnatCross

Figure 2 depicts the performance of the PirnatCross architecture where the original number of filters in the set F has been scaled by scaling factors $k \in [0.02, 0.04, ..., 2.5]$. The upper two curves present the average weighted F1 score for models trained and evaluated on REFIT and UK-DALE separately, so without cross-dataset evaluation. The second lowest curve presents the average weighted F1 scores for models trained on REFIT and cross evaluated on UK-DALE while the lowest curve presents the results on training on UK-DALE and cross evaluating on REFIT. In our experiments, we observe only the cross evaluation models, they show a rapid improvement in performance for scaling factor values from 0.02 to 0.08. From scaling factor value 0.08 to 0.9, we see a decline in performance in one example and a small improvement in the others, while beyond 0.9 the results gradually decline. For scaling factors above 1.3 a rapid drop in performance can be observed.

Marked with light blue in Figure 2 and also depicted in Figure 3 is the PirnatCross version of the proposed architecture having F scaled by 0,08 and thus resulting in the $F_1 = [5, 10, 20, 40, 40]$ filter configuration of the blocks. PirnatCross1 performs optimally in terms of avg F1 score.

PirnatCross1 also contain 5 blocks as the original VGG19. The first two comprising of two convolutional layers and the subsequent two comprising of four convolutional layers. The final block consists of four transpose convolutional layers and all blocks include an average pooling layer after the convolutional layers. Preceding the output layer, our model incorporates a GRU layer with a size of 64. Additionally, two fully-connected layers, each consisting of 4096 nodes, are included in the architecture. The output layer of our model comprises five nodes corresponding to the states s_i of the 5 appliances d_i considered in this study. All layers utilize the ReLU activation function, except for the output layer which employs the sigmoid activation function.

4.2 Computational Complexity and Carbon Footprint Analysis

Table 1 summarizes the weights, FLOPs, energy and carbon footprint numbers for PirnatCross versus the TanoniCRNN and VGG11 baselines. The results take into account the fact that the models were trained on Nvidia A100 graphics card, located in Slovenia where 250g of CO_2 equivalent is produced with each kWh of electricity. The specific equations used to calculate, energy and carbon footprint are defined in our previous work [10].

It can be seen from the second row of the table that PirnatCross achieves superior energy efficiency compared to other models, exhibiting energy consumption 6-times smaller compared to SotA TanoniCRNN and 6.6-times smaller compared to VGG11.

Table 1: Computational complexity and carbon footprint analysis for the proposed architecture and selected baselines.

NN	weights	FLOPs	energy	carbon footprint
PirnatCross	$17.4 \cdot 10^6$	$\mathbf{185 \cdot 10^6}$	**329 kJ**	**22,9 g CO_2 eq.**
TanoniCRNN [12]	$\mathbf{0.75 \cdot 10^6}$	$1.11 \cdot 10^9$	1967 kJ	136.7 g CO_2 eq.
VGG11	$185.6 \cdot 10^6$	$1.21 \cdot 10^9$	2150 kJ	149.3 g CO_2 eq.

4.3 Cross-Dataset Analysis

Tables 2 and 3 present the per device breakdown of the F1 scores for PirnatCross, TanoniCRNN and VGG11 when trained on REFIT and evaluated on UK-DALE and vice versa.

When we trained on REFIT and evaluated on UK-DALE, the scores for the four models were as follows: PirnatCross achieved a score of 0.766, TanoniCRNN achieved a score of 0.752 and VGG11

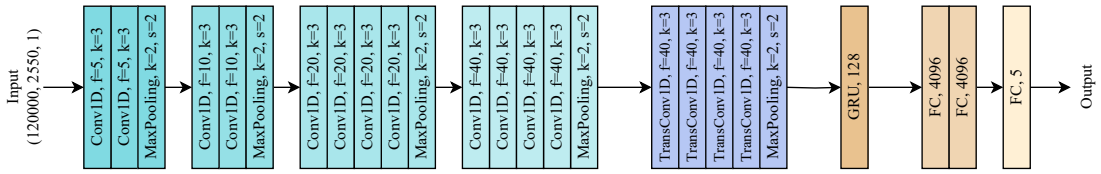

Figure 3: The proposed architecture PirnatCross made for maximum performance.

Table 2: F1 scores for PirnatCross1, TanoniCRNN [12] and VGG11 trained on REFIT and evaluated on UK-DALE.

devices	PirnatCross	TanoniCRNN [12]	VGG11
fridge	0,944	0,972	0,462
washing machine	0,650	0,690	0,544
dish washer	0,646	0,648	0,294
microwave	0,728	0,756	0,512
kettle	0,786	0,622	0,420
weighted avg	**0,766**	0,752	0,456

Table 3: F1 scores for PirnatCross1, TanoniCRNN [12] and VGG11 trained on UK-DALE and evaluated on REFIT.

devices	PirnatCross	TanoniCRNN [12]	VGG11
fridge	0,730	0,232	0,508
washing machine	0,668	0,666	0,366
dish washer	0,596	0,468	0,360
microwave	0,526	0,630	0,506
kettle	0,800	0,782	0,408
weighted avg	**0,672**	0,542	0,438

achieved a score of 0.456. However, when we trained on UK-DALE and tested on REFIT, the scores were notably lower for all four models. PirnatCross achieved a score of 0.672, TanoniCRNN achieved a score of 0.542, and VGG11 achieved a score of 0.438.

This outcome may be explained by the fact that REFIT has a significantly higher level of data noise compared to UK-DALE as shown in prior work [12]. Consequently, the testing results obtained from UK-DALE are expected to show higher F1 scores. Moreover, we observed that, overall, our model PirnatCross consistently outperformed the other models in both testing scenarios, achieving the highest weighted average F1 scores overall.

5 CONCLUSIONS

To address the challenge of cross-dataset usage scenario on NILM ON/OFF classification, we propose PirnatCross, with an aim to present the maximum performance and the energy efficiency. The results of our evaluation on the REFIT and UKDALE datasets reveal that PirnatCross achieve an average performance improvement of 7.2 over SotA and 27.2 percentage points over baseline, underscoring its superior effectiveness in handling data from diverse sources. Additionally PirnatCross consumes 6-times less energy compared to SotA model. To develop PirnatCross, we employed our methodology. In the case of classification on NILM this included beginning with the VGG19 architecture and implementing several modifications, such as replacing the convolutional layers with transpose convolutional layers in the 5th block, incorporating a GRU layer after it, and adjusting the number of filters based on our analysis. Our analysis revealed that an increase in the number of filters in convolutional layers and consequently an increase in the number of FLOPs did not necessarily lead to an improvement in classification accuracy. Instead, we observed a point of steady improvement in performance, followed by a gradual decline and a significant drop in performance when the number of filters exceeded a certain threshold. This information is crucial for optimizing the architecture of NILM models, and keeping track of the carbon footprint.

ACKNOWLEDGEMENTS

This work was funded in part by the Slovenian Research Agency under the grant P2-0016. The authors would like to thank Blaž Bertalanič for insightful discussions.

REFERENCES

[1] Jamshid Aghaei and Mohammad-Iman Alizadeh. 2013. Demand response in smart electricity grids equipped with renewable energy sources: a review. *Renewable and Sustainable Energy Reviews*, 18, 64–72. DOI: https://doi.org/10.1016/j.rser.2012.09.019.

[2] Eva García-Martín, Crefeda Faviola Rodrigues, Graham Riley, and Håkan Grahn. 2019. Estimation of energy consumption in machine learning. *Journal of Parallel and Distributed Computing*, 134, 75–88. DOI: https://doi.org/10.1016/j.jpdc.2019.07.007.

[3] R. Gopinath, Mukesh Kumar, C. Prakash Chandra Joshua, and Kota Srinivas. 2020. Energy management using non-intrusive load monitoring techniques – state-of-the-art and future research directions. *Sustainable Cities and Society*, 62, 102411. DOI: https://doi.org/10.1016/j.scs.2020.102411.

[4] Byungok Han, Woo-Han Yun, Jang-Hee Yoo, and Won Hwa Kim. 2020. Toward unbiased facial expression recognition in the wild via cross-dataset adaptation. *IEEE Access*, 8, 159172–159181.

[5] Gigi Hsueh. 2020. *Carbon Footprint of Machine Learning Algorithms*. Senior Projects Spring 2020. 296. https://digitalcommons.bard.edu/senproj_s2020/296.

[6] Jack Kelly and William Knottenbelt. 2015. The uk-dale dataset, domestic appliance-level electricity demand and whole-house demand from five uk homes. *Scientific data*, 2, 1, 1–14.

[7] Weicong Kong, Zhao Yang Dong, Bo Wang, Junhua Zhao, and Jie Huang. 2020. A practical solution for non-intrusive type ii load monitoring based on deep learning and post-processing. *IEEE Transactions on Smart Grid*, 11, 1, 148–160. DOI: 10.1109/TSG.2019.2918330.

[8] Luca Massidda, Marino Marrocu, and Simone Manca. 2020. Non-intrusive load disaggregation by convolutional neural network and multilabel classification. *Applied Sciences*, 10, 4. DOI: 10.3390/app10041454.

[9] David Murray, Lina Stankovic, and Vladimir Stankovic. 2017. An electrical load measurements dataset of united kingdom households from a two-year longitudinal study. *Scientific data*, 4, 1, 1–12. DOI: 10.1038/sdata.2016.122.

[10] Anže Pirnat, Blaž Bertalanič, Gregor Cerar, Mihael Mohorčič, Marko Meža, and Carolina Fortuna. 2022. Towards sustainable deep learning for wireless fingerprinting localization. In *ICC 2022 - IEEE International Conference on Communications*, 3208–3213. DOI: 10.1109/ICC45855.2022.9838464.

[11] Ali Q. Al-Shetwi, M.A. Hannan, Ker Pin Jern, M. Mansur, and T.M.I. Mahlia. 2020. Grid-connected renewable energy sources: review of the recent integration requirements and control methods. *Journal of Cleaner Production*, 253, 119831. DOI: https://doi.org/10.1016/j.jclepro.2019.119831.

[12] Giulia Tanoni, Emanuele Principi, and Stefano Squartini. 2022. Multi-label appliance classification with weakly labeled data for non-intrusive load monitoring. *IEEE Transactions on Smart Grid*, 1–1. DOI: 10.1109/TSG.2022.3191908.

[13] Minjia Zhang, Wenhan Wang, Xiaodong Liu, Jianfeng Gao, and Yuxiong He. 2018. Navigating with graph representations for fast and scalable decoding of neural language models. *Advances in neural information processing systems*, 31.

[14] Mengran Zhou, Shuai Shao, Xu Wang, Ziwei Zhu, and Feng Hu. 2022. Deep learning-based non-intrusive commercial load monitoring. *Sensors*, 22, 14. DOI: 10.3390/s22145250.