Recurrence Plot Spacial Pyramid Pooling Network for Appliance Identification in Non-Intrusive Load Monitoring

1st Marc Wenninger
Department of Computer Science
Technische Hochschule Rosenheim
Rosenheim, Germany

2nd Sebastian P. Bayerl
Department of Computer Science
Technische Hochschule Nürnberg Georg Simon Ohm
Nürnberg, Germany

3rd Andreas Maier
Pattern Recognition Lab
Friedrich-Alexander-Universität Erlangen-Nürnberg
Erlangen, Germany

4th Jochen Schmidt
Department of Computer Science
Technische Hochschule Rosenheim
Rosenheim, Germany

***Abstract*—Parameter free Non-intrusive Load Monitoring (NILM) algorithms are a major step toward real-world NILM scenarios. The identification of appliances is the key element in NILM. The task consists of identification of the appliance category and its current state. In this paper, we present a parameter free appliance identification algorithm for NILM using a 2D representation of time series known as unthresholded Recurrence Plots (RP) for appliance category identification. One cycle of voltage and current (V-I trajectory) are transformed into a RP and classified using a Spacial Pyramid Pooling Convolutional Neural Network architecture. The performance of our approach is evaluated on the three public datasets COOLL, PLAID and WHITEDv1.1 and compared to previous publications. We show that compared to other approaches using our architecture no initial parameters have to be manually tuned for each specific dataset.**

***Index Terms*—NILM, V-I trajectory, Recurrence Plot.**

I. Introduction

The world-wide electricity consumption is ever increasing and with 63% its majority is still generated from fossil fuels [1]. In terms of climate change, increasing the share of renewable sources as well as guiding the demand are the key responsibility in the electricity sector. The introduction of renewable energy sources comes along with many new challenges. The amount of electricity generated from photovoltaic systems and wind turbines is intrinsically tied to the natural conditions. These systems do not produce energy when demanded, but when the conditions are good. It is desirable that the demand will take such fluctuations into account, which requires understanding the individual electricity consumption. It has been shown that increasing awareness of the individual electricity consumption in private households can have an effect on the consumed electricity. Studies on the impact of electricity consumption information systems show a decrease in consumption between 5% and 15% [2]. The decrease is attributed to the increased awareness of the actual consumption. A study on resource consumption in private households concludes that householders desire a deeper understanding of their behavior on resource usage [3].

Attributing the total electricity consumption of an household to individual appliances requires power sensing technologies in the form of smart-meters or smart-plugs. While smart-meters only provide the total consumption at a certain sampling frequency, smart-plugs can directly monitor individual appliances. In literature these different levels of metering are referred to as Non-Intrusive-Load-Monitoring (NILM) and Intrusive-Load-Monitoring (ILM). In order to lower the complexity that lies in analyzing the electricity consumption, a range of machine learning algorithms have been developed that aid the identification of individual appliances as well as extracting usage patterns.

In a NILM scenario, the individual loads are not directly available and have to be computed from the total load. Otherwise, detailed information about each appliance are not available. The mathematical decomposition of a total load $P(t)$ into M sub loads $p_m(t)$ is known as disaggregation and is described by:

$$P(t) = \sum_{m=1}^{M} p_m(t) \quad . \tag{1}$$

The NILM process usually consists of two steps: identification of the currently running appliances m followed by an energy attribution estimation. The identification of the appliance is the primary NILM challenge. In machine learning terms, the appliance identification is a classification problem, where a point in the aggregate $P(t)$ can be assigned to a set of running appliances $\{m_1, m_2, \dots\}$. The appliance identification is also called appliance recognition, appliance classification or appliance event detection. In literature NILM algorithms are typically divided into event-based and non-event based. Event-based approaches classify only certain segments of the signal, called events. Although the term is widely used, it is not clearly defined. Hart describes events as step-like changes in the signal [4]. Wild et al. define events as the “transition from one steady state to another steady state which definitely differs from the previous one” [5]. Following this definition, event-based algorithms work on the transient state, while non-event based rely on the steady-state signal. Despite this definition, event-based methods based on steady-state features have been described [6], [7]. In such cases, the term event is not strongly coupled to all step-like changes in the signal. It is rather coupled to the switching ON/OFF event and the steady-state that follows the event.

A wide range of features and classification algorithms have been proposed in the NILM context. This classification task is commonly performed on derived features. The most common features used are real power (P) and reactive power (Q) consumption [8]. It can be used on low sample rates data. PQ features are a common approach when using the steady-state signal of an appliance and have been used by [9]–[12].

When higher sample rates are available, the signals’ harmonics can be used as a feature. Abrupt short current pulses created by modern appliances provide an an exploitable signature. This feature is known as harmonic distortion (HAR) and is used by [13]–[18].

Lam et al. [19] describe a feature known as voltage-current (V-I) trajectory. The V-I trajectory is a 2-dimensional feature composed of the voltage and current of one normalized wave cycle. This high sample frequency feature therefore requires sample rates well above the utility frequency of 50 Hz or 60 Hz. The idea is, that the mutual locus of the steady state instantaneous voltage and current form provide unique shapes for each appliance. V-I trajectory has already been used by [6], [7], [19]–[28].

A wide range of classification algorithms have been used to associate an observed feature with an application type or category. Over the years Neural Networks (NN) [6], [7], [14], [16], [18], [29] k-Nearest Neighbor (kNN) [30]–[33], Support Vector Machines (SVM) [16], [28], [30], [34]–[37], Hidden Markov Models (HMM) [12], [38], [39], Naïve Bayes [10], and Gaussian mixture models [11], [34] have been used for the classification task.

In the following we propose an appliance identification method based on the voltage-current (V-I) trajectory. We transform the V-I trajectory into two unthresholded recurrence plots and use a Spacial Pyramid Pooling Convolutional Neural Network in order to identify appliances. A similar approach has been proposed by [6], [7]. While their approach requires handcrafted parameters for each tested dataset, the approach described here relies on the same parameters for each tested dataset while having equivalent or superior classification performance. Our main contributions are:

- A highly generalized appliance identification algorithm for NILM, based on parameter-free recurrence plots.
- The calculation of the recurrence plot is implemented as a Neural Network layer, simplifying the processing pipeline and improving the required computation time by using the GPU.
- Evaluation on three public datasets COOLL, PLAID and WHITEDv1.1 [40]–[42], including comparison to similar work. The evaluation shows that our approach results in near perfect scores on COOLL and WHITEDv1.1 and very good scores on PLAID while using the same parameters for all three datasets.

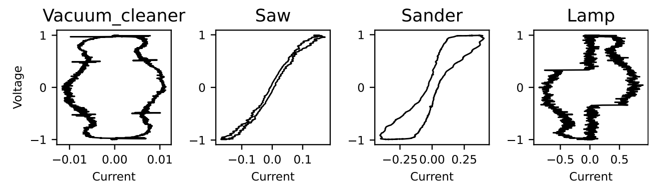

Fig. 1. Standard V-I trajectory plots where current and voltage are plotted against each other. Examples are taken from the COOLL dataset. The data are scaled to range -1 to $+1$.

Our code has been published as open source under MIT license[1].

II. Method

The classification of appliances is a crucial part in the NILM process. This section describes how V-I trajectories were extracted and how unthresholded recurrence plots were calculated from these. We also provide the implementation details for their GPU-based computation. The section concludes with the description of the neural network architecture used in subsequent experiments.

A. V-I trajectory

We utilize an aligned V-I trajectory. The V-I trajectory is a single voltage and current cycle when the appliance is in steady state. Voltage and current are usually plotted on two axes as shown in Fig. 1. The different shapes of the plots are clearly visible. Lam et al. [19] describe several shape features based on the plotted V-I trajectory: asymmetry, looping direction, area, curvature of mean line, self-intersection, slope of middle segment, area of left and right segments, and peak of middle segment. While such hand crafted features help to understand the different characteristics, it has been shown that by using a deep neural network the classification of V-I plots can directly be learned by a classification algorithm [6], [7], [20], [24], [25].

[20], [24], [25] use a pixelated, binary version of such plots, resulting in an image classification like problem.

The feature extraction process used in this work is shown in Fig. 2. We first extract one cycle in the data by searching for zero crossings in the voltage signal using the algorithm in Listing 1. In case of the datasets used for the evaluation of this work, we search for a steady-state cycle in the signal first. This is done by taking 20 cycles after the switch-on event, and then using the last full-cycle available by looking

[1] https://github.com/walwe/rpspp

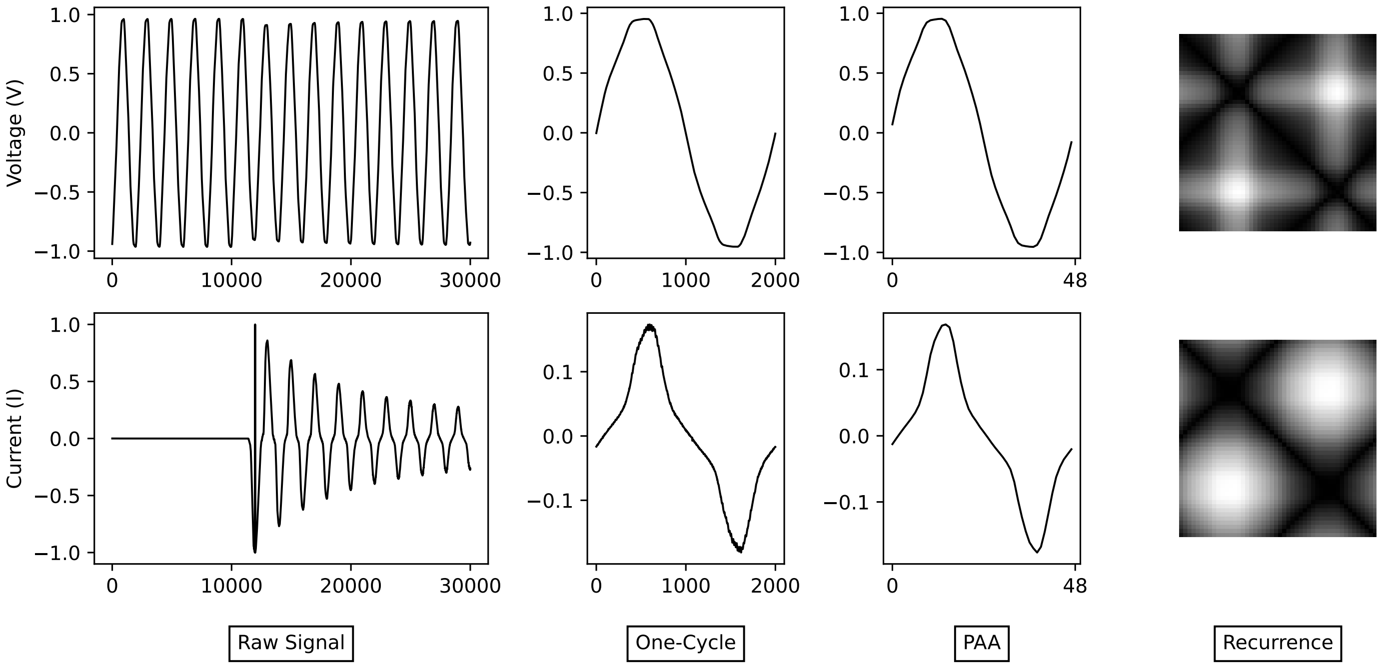

Fig. 2. Feature extraction process, shown for two different signals: First we extract a voltage zero crossing aligned cycle. The cycle is then subsampled using a piecewise aggregate approximation (PAA) in order to reduce dimensionality. The voltage and current cycles are then transformed into a Recurrence Plot.

Listing 1
PYTHON NOTATION OF SEARCH ALGORITHM FOR SINGLE VOLTAGE AND CURRENT CYCLE

```
# Input:
#   v_data = Voltage data
#   i_data = Current data
#   period_length = period length of one cycle

# Find all positive values
positive = v_data > 0

# Find Zero crossing indices
zc_idx = bitwise_xor(positive[1:], positive[:-1])

# Start on uphill slope
if v_data[zc_idx[0]] > v_data[zc_idx[0] + 1]:
    zc_idx = zc_idx[1:]

# Check for even number of zero crossings
if len(zero_crossings) % 2 == 1:
    zc_idx = zc_idx[:-1]

# Assure full period for last zero crossing
if zc_idx[-2] + period_length >= len(v_data):
    zc_idx = zc_idx[:-2]

# Choose last cycle
start = zc_idx[-2]
stop = zc_idx[-2] + period_length

v_cycle = v_data[start:stop]
i_cycle = i_data[start:stop]
```

for the zero crossing in order to ensure we are in steady-state. The corresponding current measurements are extracted in alignment with the voltage zero crossings. In order to reduce the number of data points, we scale each cycle to 48 values using a piecewise aggregate approximation (PAA) [43]. PAA is a fast dimensionality reduction algorithm, that approximates pieces of the signal by computing the mean value of equal frames of the original signal. We use the implementation provided by the Python package *pyts* [44].

B. Recurrence Plots

Recurrence is a property of many natural processes and systems. States that have been observed more than once are often followed by similar states. These recurrences can be represented in a recurrence matrix $\boldsymbol{R} = (R_{i,j})$ defined as:

$$R_{i,j}(\epsilon) = \Theta(\epsilon - ||\vec{x_i} - \vec{x_j}||), i,j = 1, ..., N \quad . \tag{2}$$

N represents the number of samples $\vec{x_i}$, ϵ is a threshold, and Θ the Heaviside or step function where $\Theta(x) = 0$ if $x \leq 0$ and $\Theta(x) = 1$ otherwise. An appropriate distance measure which fits the time series to be analyzed has to be chosen for $||.||$. Common choices are the Euclidean or cosine distance.

For states that are in the ϵ-neighborhood (3) applies:

$$\vec{x_i} \approx \vec{x_j} \iff R_{i,j} \equiv 1 \quad . \tag{3}$$

Recurrence Plots (RP) are a visual representation of the recurrence matrix that can be used for qualitative assessment by humans [45]. The RP is generated by binarizing the recurrence matrix using a threshold.

When using powerful classification algorithms, such as Neural Networks, denser representations that encode more information about the nature of a time series might be useful. Behavior similar to cutting-off or binarizing with an ϵ parameter can be learned by a Neural Network. A denser representation can be obtained by calculating the pairwise distances to obtain a distance matrix $\boldsymbol{D}$:

$$D_{i,j} = ||\vec{x} - \vec{y}|| \quad . \tag{4}$$

These pairwise distances are then plotted. This modification of the RP is also known as *unthresholded recurrence plot* or *distance plot* as described in [45]. We use the Euclidean distance for the distance calculation and follow the processing steps in [46], using a threshold cut-off at three times the standard deviation σ of all distances in the recurrence matrix:

$$D_{i,j} = D_{i,j}(d \leq 3\sigma) = \begin{cases} 3\sigma & d \geq 3\sigma \\ d & d < 3\sigma \end{cases} \quad . \tag{5}$$

We made our PyTorch recurrence plot implementation available online[2].

Unthresholded recurrence plots have been shown to work as feature inputs for a vast range of time series classification problems [46]. [7] use a Weighted Recurrence Plot (WRG) introducing a weight parameter δ that is used to tune the cut-off. Both parameters are also fine tuned as parameters during neural network training [6].

C. Classification

The generated recurrence plots are classified using a Convolutional Neural Network (CNN) combined with Spacial Pyramid Pooling (SPP). Using a CNN and SPP has been shown to work effectively in visual recognition tasks [47]. Figure 3 shows the full network architecture, which consists of four blocks: Recurrence, CNN, SPP, and a fully connected layer (FC).

The network inputs are the two channels, voltage and current, which have been resized using PAA to a vector of length 48. Each channel is converted into a recurrence plot which is then fed into a series of CNN layers. The CNN part of the network contains four layers. All CNN layers use a 3×3 kernel and have a Rectified Linear Unit (ReLU) activation function. The first layer has the same number of input and output channels and a two-dimensional max pooling layer of size 3×3. The other CNN layers all use 32 channels. Two-dimensional batch normalization is applied in the last two layers. The last CNN layer also has a two-dimensional dropout with a probability of 0.2 applied. We found the batch normalization as well as the dropout to be crucial for the network to generalize quickly and to prevent overfitting.

The SPP layer is a combination of four two-dimensional adaptive max pooling filters of different sizes (1×1, 2×2, 4×4, 8×8). Adaptive max pooling is a dimensionality reduction mechanism that outputs a fixed size vector by adapting the max pooling kernel size. In our case, the output size of the CNN block and therefore the input size of the SPP layer is of size $32 \times 16 \times 16$, where 32 are the number of CNN channels. The size has been reduced from 48×48 to 16×16 by the 3×3 max pooling in the first layer of the CNN block. Since 16×16 is an integer multiple of the SPP filter sizes, adaptive max pooling is implemented by calculating the stride and kernel size as shown in Listing 2.

[2]https://github.com/walwe/pytorch-recurrence

Listing 2
Calculation of stride and kernel size for adaptive max pooling

```
stride = in_size // out_size
kernel_size = in_size - (out_size -1)*stride
```

The max pooling filter outputs are stacked to form a new vector. The SPP layer therefore outputs a fixed length vector of size $32 \cdot (1 \cdot 1 + 2 \cdot 2 + 4 \cdot 4 + 8 \cdot 8) = 2720$, where 32 is the number of channels produced by the last CNN layer. $1 \cdot 1, 2 \cdot 2, \ldots$ are the output sizes of the adaptive max pooling layers applied to the channels. The layer has the purpose of recognizing different features in the recurrence plot by partitioning the plot into finer and coarser levels. This way, the network is capable of recognizing features in the plot of different sizes. The flattened and stacked SPP layer output is then fed into the FC layer. The SPP layer also works as a dimensionality reduction before the FC layer. The last layer's output size is equivalent to the number of classes to be classified.

III. Experiments

In the following we present an evaluation of the proposed method. For direct comparison, the evaluation was done in close alignment with the evaluation presented by [7]. The evaluation presented here is as described by the authors and their experiment has been reproduced as closely as possible by using the same dataset split approach and random seed.

A. Data

The data used for evaluation of the proposed appliance identification method are three public datasets: COOLL, PLAID and WHITEDv1.1 [40]–[42]. These datasets have been widely used for the evaluation of appliance identification methods.

COOLL (Controlled On/Off Loads Library) is a dataset containing controlled On/Off loads of 42 appliances of 12 categories. The data was recorded at 100 kHz sampling frequency, which is a very high sampling rate compared to other datasets. The dataset contains 20 samples per appliance, recorded in a laboratory in France. Similar to other publications, we only make use of the appliance categories with at least 2 appliances and therefore 40 samples. The categories used are: Drill, Fan, Grinder, Hair dryer, Hedge trimmer, Lamp, Sander, Saw, Vacuum cleaner.

The second dataset used is PLAID 2017 (Plug Load Appliance Identification Dataset). The dataset was collected in the US. It was recorded at a sampling frequency of 30 kHz. It contains a total of 1793 records from 11 different appliance categories.

The third dataset used for evaluation is WHITEDv1.1 (Worldwide Household and Industry Transient Energy Data Set). It contains 1259 recordings of 47 appliance categories with a total of 110 different appliances. The dataset contains the first 5 seconds of the appliance start-ups, recorded at a

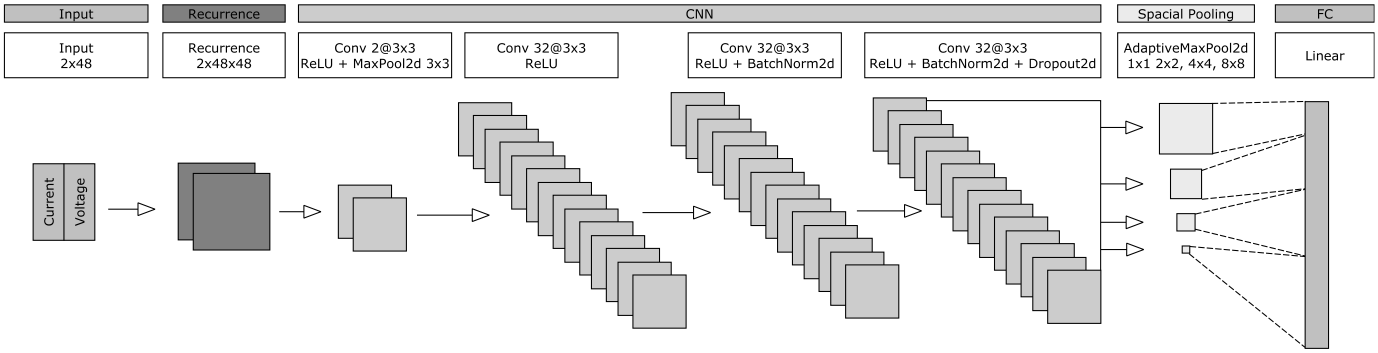

Fig. 3. Deep Neural Network used for appliance identification. The network is composed of four parts: Recurrence, CNN, Spacial Pyramid Pooling, and FC.

sampling frequency of 44.1 kHz. Data from 21 of the available categories is used.

B. Validation

The evaluation is performed in a stratified k-fold and a leave-one-group-out cross-validation. The latter is done in order to test the generalization performance. The PLAID dataset provides groups in the form of locations called house 1–55 that are used for the leave-one-group-out validation. COOLL and WHITED do not provide such groups as part of the dataset. Therefore, the appliances were assigned to random groups based on their provided appliance name as proposed by De Baets [48]. The COOLL dataset for example provides appliances' names in the form of: Drill_1, Drill_2, …, Drill_6. The maximum number of different appliances in one category is 8, thus the number of created groups/houses for COOLL is 8. A new label in the range 0–7 is randomly assigned to the 6 Drill groups. Therefore 6 of the 8 houses contain a Drill, but no house will share the exact same Drill, resulting in a real generalization task. This grouping results in 8 groups for COOLL and 9 for WHITEDv1.1. Faustine et al. [7] present their WRG model results, claiming to use the same data split approach. Based on our review or their published source code[3], we conclude that the results they present are not based on the group split described, but rather a k-fold like split. Therefore, corrected results for the WRG algorithm based on our evaluation are also presented.

The final evaluation is done using a macro F_1-score. The F_1-score is calculated as:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{\text{TP}}{\text{TP} + 0.5(\text{FP} - \text{FN})} \quad , \qquad \text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad , \text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad . \tag{6}$$

where TP, FP and FN are the number of true positives, false positives, and false negatives. We then calculate the macro F_1-score for each fold and finally average all folds:

$$F_{\text{macro}} = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{N} \sum_{a=1}^{N} F_{k,a} \tag{7}$$

[3]https://github.com/sambaiga/WRG-NILM

TABLE I
LEAVE ONE GROUP OUT RESULTS (F_{MACRO}) ON COOLL, WHITEDV1.1 AND PLAID. WRG RESULTS HAVE BEEN REEVALUATED. DE BAETS RESULTS ARE AS PRESENTED IN THEIR PUBLICATIONS.

Algorithm	COOLL	WHITEDv1.1	PLAID
De Baets [48]	N/A	**0.7546**	0.7760
WRG [7]	0.447	0.3954	0.8921
Ours	**0.5329**	0.4310	**0.8942**

TABLE II
5-FOLD CROSS VALIDATION RESULTS (F_{MACRO}) OF OUR EXPERIMENTS ON COOLL, WHITEDV1.1 AND PLAID. DE BAETS IS NOT LISTED AS THEY DO NOT PROVIDE RESULTS.

Algorithm	COOLL	WHITEDv1.1	PLAID
WRG [7]	0.8957	**0.9984**	0.8082
Ours	**0.9213**	0.9924	**0.8456**

where K is the number of folds, N the number of appliance categories and $F_{k,a}$ the F_1-score of each category a in fold k.

IV. RESULTS

The results of the evaluation of the appliance identification task show an improvement on COOLL and PLAID in both evaluations in comparison to De Baets [48] and WRG [7]. Results are presented in Table I and II. The proposed Spacial Pyramid Pooling CNN network is capable of achieving the highest F_1-score for the PLAID and COOLL dataset in both evaluation scenarios. We show De Baets results as published by the authors. The authors did not publish results for the COOLL dataset. As they did not publish their source code, it cannot be assured that their experimental setup is comparable. Our reevaluation of the WRG algorithm shows very different results compared to the authors' published results due to the difference in generating the groups for the leave-one-group-out evaluation.

The leave-one-group-out evaluation of our approach for COOLL and WHITEDv1.1 results in a F_1-score of 0.5329 and 0.4310. While on COOLL our approach achieves a higher score compared to WRG, the absolute scores are low. The confusion matrix in Fig. 4 shows that the Drill, Grinder, Hedge trimmer, Saw and Vacuum are getting confused. The current signature of all these appliances can in some cases be very similar, leaving no grounds for separation. On the other

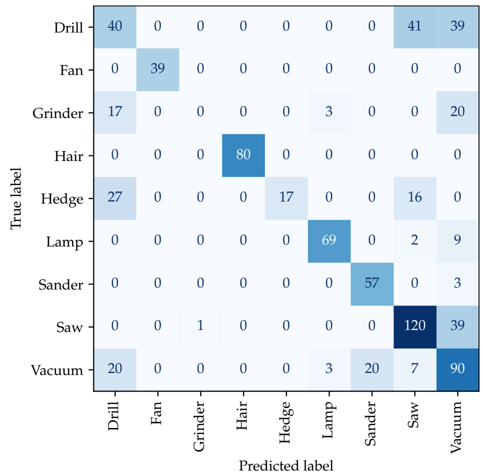

Fig. 4. Confusion matrix of leave-one-house-out test split result on COOLL dataset. Cross validation results have been aggregated.

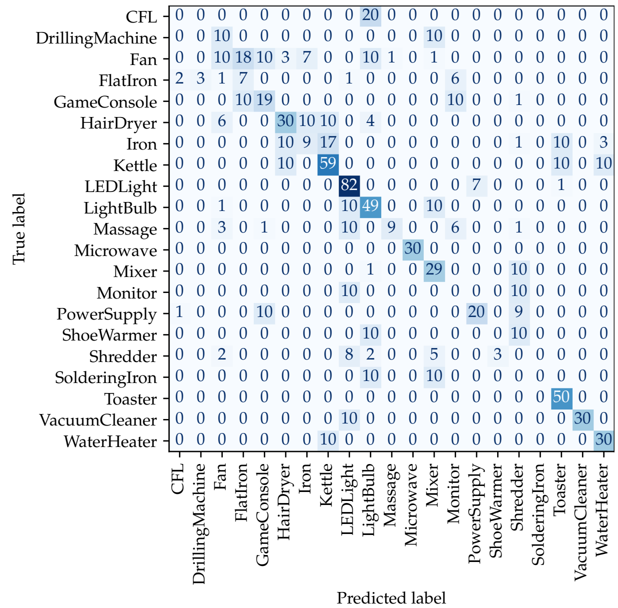

Fig. 5. Confusion matrix of leave-one-house-out test split result on WHITEDv1.1 dataset. Cross validation results have been aggregated.

classes, the classification works very well. On WHITEDv1.1 the approach presented by De Baets achieves the highest score of 0.7546. Similarly to the COOLL dataset, both WRG and our approach do not result in good scores. The confusion matrix in Fig. 5 shows that the CFL class is always misclassified as LighBulb. In general there are some clear patterns of misclassifications resulting from evaluations folds were the training split was insufficient for the test split. In these cases, our algorithm is not capable of learning a good enough representation of the appliance category in order to classify the unseen appliance.

On PLAID both WRG and our approach outperform De Baets with an F_1-score of 0.8921 and 0.8942. Figure 6 shows that Air Conditioner and Fan are getting confused as well as Hairdryer and Heater. The misclassification pattern shows that very similarly operating appliances are hard to separate by their V-I trajectory. Air Conditioner and Fan both have a similar ventilation mechanism. Hairdryer and Heater both use a heating coil. The V-I trajectory recurrence plot seems to not provide a good enough feature in order to separate these very close appliance categories.

The results on the 5-fold cross validations are much better. For the COOLL and PLAID dataset we outperform the WRG algorithm. On COOLL we achieve a F_1-score of 0.9213 while WRG achieves 0.8957. On WHITEDv1.1 both approaches achieve near perfect scores, where WRG has a score of 0.9984 and ours 0.9924. Our score for the PLAID dataset is 0.8456 compared to WRG with a score of 0.8082.

The evaluation using the leave-one-group-out for the COOLL and WHITEDv1.1 dataset is a very hard task for neural networks as the datasets are small and the risk of overfitting is high. While the 5-folds validation results in very good scores, the leave-one-house-out task only results in good scores for PLAID, likely due to the PLAID dataset being much larger compared to the others.

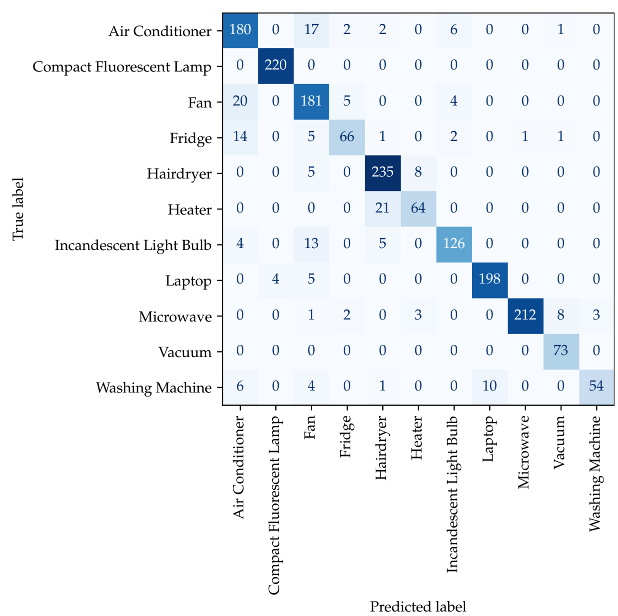

Fig. 6. Confusion matrix of leave-one-house-out test split result on PLAID dataset. Cross validation results have been aggregated.

V. Conclusion

We presented a new Deep Neural Network architecture using Spacial Pyramid Pooling for appliance identification in NILM. Our approach uses an unthresholded Recurrence Plot (RP) that is created as part of the Neural Network. The use of an unthresholded RP leaves the dimensionality reduction to the Neural Network. Unlike previous work we have reduced the network complexity and removed additional hyperparameters that had to be tuned for each dataset. The presented evaluation on three public datasets shows an improved F_1-score for the

PLAID dataset in a leave-one-house-out scenario. In a 5-folds cross validation we achieve near perfect results for COOLL and WHITEDv1.1. We have corrected the results on the WRG algorithm presented by Faustine et al. [7]. While there is still some improvement possible on the PLAID dataset, we expect that further improvement will have to rely on more than a single cycle, in order to introduce richer features. Since the evaluation has been performed on single appliance measurements, the performance on aggregated data is still unclear. In future work we would like to extend the approach to work on aggregated measurements.

ACKNOWLEDGMENT

Major parts of this work were funded by the German Federal Ministry of Education and Research (BMBF), grant 01LY1506, and supported by the Bayerische Wissenschaftsforum (BayWISS).

REFERENCES

[1] BP p.l.c., "BP statistical review of world energy 2020," 2020, https://www.bp.com/content/dam/bp/business-sites/en/global/corporate/pdfs/energy-economics/statistical-review/bp-stats-review-2020-full-report.pdf.

[2] T. Hargreaves, M. Nye, and J. Burgess, "Making energy visible: A qualitative field study of how householders interact with feedback from smart energy monitors," *Energy Policy*, vol. 38, no. 10, pp. 6111 – 6119, 2010, the socio-economic transition towards a hydrogen economy - findings from European research, with regular papers.

[3] M. Chetty, D. Tran, and R. E. Grinter, "Getting to green: Understanding resource consumption in the home," in *Proc. of the 10th International Conference on Ubiquitous Computing*, ser. UbiComp '08. New York, NY, USA: ACM, 2008, pp. 242–251.

[4] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[5] B. Wild, K. S. Barsim, and B. Yang, "A new unsupervised event detector for non-intrusive load monitoring," in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2015, pp. 73–77.

[6] A. Faustine, L. Pereira, and C. Klemenjak, "Adaptive weighted recurrence graphs for appliance recognition in non-intrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 398–406, 2021.

[7] A. Faustine and L. Pereira, "Improved appliance classification in non-intrusive load monitoring using weighted recurrence graph and convolutional neural networks," *Energies*, vol. 13, no. 13, 2020.

[8] I. Abubakar, S. Khalid, M. Mustafa, H. Shareef, and M. Mustapha, "Recent approaches and applications of non-intrusive load monitoring," *ARPN journal of engineering and applied sciences*, vol. 11, pp. 4609–4618, 04 2016.

[9] M. Weiss, A. Helfenstein, F. Mattern, and T. Staake, "Leveraging smart meter data to recognize home appliances," in *2012 IEEE International Conference on Pervasive Computing and Communications*, 2012, pp. 190–197.

[10] A. Reinhardt, P. Baumann, D. Burgstahler, M. Hollick, H. Chonov, M. Werner, and R. Steinmetz, "On the accuracy of appliance identification based on distributed load metering data," in *Proc. SustainIT*, Jan. 2012, pp. 1–9.

[11] A. Ridi, C. Gisler, and J. Hennebert, "Automatic identification of electrical appliances using smart plugs," in *2013 8th International Workshop on Systems, Signal Processing and their Applications (WoSSPA)*, 2013, pp. 301–305.

[12] A. Zaidi, F. Kupzog, T. Zia, and P. Palensky, "Load recognition for automated demand response in microgrids," in *IECON Proc. (Industrial Electronics Conference)*, 12 2010, pp. 2442 – 2447.

[13] C. Feng, H. Hoe, M. P. Abdullah, M. Y. Hassan, and F. Hussin, "Tracing of energy consumption by using harmonic current," *Proceeding - 2013 IEEE Student Conference on Research and Development, SCOReD 2013*, pp. 444–449, 01 2013.

[14] D. Srinivasan, W. Ng, and A. Liew, "Neural-network-based signature recognition for harmonic source identification," *Power Delivery, IEEE Transactions on*, vol. 21, pp. 398 – 405, 02 2006.

[15] P. Meehan, S. Phelan, C. McArdle, and S. Daniels, "Temporal and frequency analysis of power signatures for common household appliances," in *IET Conference Publications*, vol. 2012, 07 2012, pp. 22–27.

[16] D. Srinivasan, W. S. Ng, and A. C. Liew, "Neural-network-based signature recognition for harmonic source identification," *IEEE Transactions on Power Delivery*, vol. 21, no. 1, pp. 398–405, 2006.

[17] H. Ma and A. A. Girgis, "Identification and tracking of harmonic sources in a power system using a kalman filter," *IEEE Transactions on Power Delivery*, vol. 11, no. 3, pp. 1659–1665, 1996.

[18] K. Janani and S. Himavathi, "Non-intrusive harmonic source identification using neural networks," in *Proc. of International Conference on Computation of Power, Energy, Information and Communication, ICCPEIC 2013*, 04 2013, pp. 59–64.

[19] H. Y. Lam, G. S. K. Fung, and W. K. Lee, "A novel method to construct taxonomy electrical appliances based on load signatures," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 2, pp. 653–660, 2007.

[20] L. De Baets, T. Dhaene, D. Deschrijver, C. Develder, and M. Berges, "Vi-based appliance classification using aggregated power consumption data," in *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2018, pp. 179–186.

[21] E. Gomes and L. Pereira, "Pb-nilm: Pinball guided deep non-intrusive load monitoring," *IEEE Access*, vol. 8, pp. 48 386–48 398, 2020.

[22] Y. Liu, X. Wang, and W. You, "Non-intrusive load monitoring by voltage–current trajectory enabled transfer learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5609–5619, 2019.

[23] D. Baptista, S. S. Mostafa, L. Pereira, L. Sousa, and F. Morgado-Dias, "Implementation strategy of convolution neural networks on field programmable gate arrays for appliance classification using the voltage and current (v-i) trajectory," *Energies*, vol. 11, no. 9, 2018.

[24] L. Du, D. He, R. G. Harley, and T. G. Habetler, "Electric load classification by binary voltage–current trajectory mapping," *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 358–365, 2016.

[25] J. Gao, E. C. Kara, S. Giri, and M. Bergés, "A feasibility study of automated plug-load identification from high-frequency measurements," in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2015, pp. 220–224.

[26] A. L. Wang, B. X. Chen, C. G. Wang, and D. Hua, "Non-intrusive load monitoring algorithm based on features of v–i trajectory," *Electric Power Systems Research*, vol. 157, pp. 134–144, 2018.

[27] L. Li, Y. Zhao, D. Jiang, Y. Zhang, F. Wang, I. Gonzalez, E. Valentin, and H. Sahli, "Hybrid deep neural network–hidden markov model (dnn-hmm) based speech emotion recognition," in *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, 2013, pp. 312–317.

[28] T. Kato, H. S. Cho, D. Lee, T. Toyomura, and T. Yamazaki, "Appliance recognition from electric current signals for information-energy integrated network in home environments," in *Ambient Assistive Health and Wellness Management in the Heart of the City*, M. Mokhtari, I. Khalil, J. Bauchet, D. Zhang, and C. Nugent, Eds. Springer Berlin Heidelberg, 2009, pp. 150–157.

[29] J.-G. Kim and B. Lee, "Appliance classification by power signal analysis based on multi-feature combination multi-layer lstm," *Energies*, vol. 12, no. 14, p. 2804, 2019.

[30] M. Kahl, T. Kriechbaumer, D. Jorde, A. Ul Haq, and H.-A. Jacobsen, "Appliance event detection - a multivariate, supervised classification approach," in *Proc. of the Tenth ACM International Conference on Future Energy Systems*, ser. e-Energy '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 373–375.

[31] M. Berges, E. Goldman, H. Matthews, and L. Soibelman, "Enhancing electricity audits in residential buildings with nonintrusive load monitoring," *Journal of Industrial Ecology*, vol. 14, 11 2010.

[32] M. Figueiredo, A. De Almeida, and B. Ribeiro, "An experimental study on electrical signature identification of non-intrusive load monitoring (nilm) systems," in *Adaptive and Natural Computing Algorithms*, Jan. 2011, pp. 31–40.

[33] S. Giri, P.-H. Lai, and M. Bergés, "Novel techniques for the detection of on and off states of appliances for power estimation in non-intrusive load monitoring," in *Proc. of the 30th International Symposium on Automation and Robotics in Construction and Mining (ISARC 2013): Building the Future in Automation and Robotics*, F. Hassani, O. Moselhi, and C. Haas, Eds. Montreal, Canada: International Association for Automation and Robotics in Construction (IAARC), August 2013, pp. 522–530.

[34] Y.-X. Lai, C.-F. Lai, Y.-M. Huang, and H.-C. Chao, "Multi-appliance recognition system with hybrid svm/gmm classifier in ubiquitous smart home," *Inf. Sci.*, vol. 230, p. 39–55, May 2013.

[35] L. Jiang, S. Luo, and J. Li, "Automatic power load event detection and appliance classification based on power harmonic features in non-intrusive appliance load monitoring," in *Proc. of the 2013 IEEE 8th Conference on Industrial Electronics and Applications, ICIEA 2013*, Jun. 2013, pp. 1083–1088.

[36] S. N. Patel, T. Robertson, J. A. Kientz, M. S. Reynolds, and G. D. Abowd, "At the flick of a switch: Detecting and classifying unique electrical events on the residential power line," in *UbiComp 2007: Ubiquitous Computing*, J. Krumm, G. D. Abowd, A. Seneviratne, and T. Strang, Eds. Springer Berlin Heidelberg, 2007, pp. 271–288.

[37] L. Jiang, S. Luo, and J. Li, "An approach of household power appliance monitoring based on machine learning," in *2012 Fifth International Conference on Intelligent Computation Technology and Automation*, 2012, pp. 577–580.

[38] H. Kim, M. Marwah, M. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements," in *Proc. of the 11th SIAM International Conference on Data Mining, SDM 2011*, Dec. 2011, pp. 747–758.

[39] Z. Kolter and M. J. Johnson, "REDD: A public data set for energy disaggregation research," in *Proc. of SustKDD*, 2011.

[40] T. Picon, M. Nait Meziane, P. Ravier, G. Lamarque, C. Novello, J.-C. Le Bunetel, and Y. Raingeaud, "COOLL: Controlled on/off loads library, a public dataset of high-sampled electrical signals for appliance identification," *arXiv preprint arXiv:1611.05803 [cs.OH]*, 2016.

[41] J. Gao, S. Giri, E. C. Kara, and M. Bergés, "Plaid: A public dataset of high-resoultion electrical appliance measurements for load identification research: Demo abstract," in *Proc. of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, ser. BuildSys '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 198–199.

[42] M. Kahl, A. Haq, T. Kriechbaumer, and H. Jacobsen, "Whited-a worldwide household and industry transient energy data set," in *3rd International Workshop on Non-Intrusive Load Monitoring*, 2016.

[43] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality reduction for fast similarity search in large time series databases," *Knowledge and Information Systems*, vol. 3, Jan. 2002.

[44] J. Faouzi and H. Janati, "pyts: A python package for time series classification," *Journal of Machine Learning Research*, vol. 21, no. 46, pp. 1–6, 2020. [Online]. Available: http://jmlr.org/papers/v21/19-763.html

[45] N. Marwan, M. Carmenromano, M. Thiel, and J. Kurths, "Recurrence plots for the analysis of complex systems," *Physics Reports*, vol. 438, no. 5-6, pp. 237–329, jan 2007.

[46] M. Wenninger, S. P. Bayerl, J. Schmidt, and K. Riedhammer, "Timage – a robust time series classification pipeline," in *Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series*. Cham: Springer International Publishing, 2019, pp. 450–461.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *Lecture Notes in Computer Science*, p. 346–361, 2014.

[48] L. De Baets, J. Ruyssinck, C. Develder, T. Dhaene, and D. Deschrijver, "Appliance classification using vi trajectories and convolutional neural networks," *Energy and Buildings*, vol. 158, pp. 32–36, 2018.