DEEP NEURAL NETWORKS FOR APPLIANCE TRANSIENT CLASSIFICATION

Peter Davies, Jon Dennis, Jack Hansom, William Martin, Aistis Stankevicius and Lionel Ward

Green Running, St Magnus House, 3 Lower Thames Street, London EC3R 6HD
{p.davies,j.dennis,j.hansom,w.martin,a.stankevicius,l.ward}@greenrunning.com

ABSTRACT

Smart plugs are useful devices for measuring the appliance load, but intrusive. It has long been the goal of energy companies and researchers to monitor the load of all household appliances in a non-intrusive manner, using only a single smart meter. We show that deep neural networks can be extremely effective in this regard. Automatic feature learning can pick out distinctive load-dependent shapes in the time series power data, enabling identification of appliance signatures. We find that properly arranged deep neural networks are capable of multi-class appliance classification, outperforming a traditional multiclass classification algorithm. We evaluate on the public PLAID dataset, and compare results with features extracted from sampling frequencies in the 1Hz-1kHz range. We show that adding features extracted from high frequency sampling significantly improves classification performance over data obtained at typical smart meter frequencies.

***Index Terms*—** Deep neural networks,non-intrusive load monitoring,real power,trajectories,transient

1. INTRODUCTION

Event based non-intrusive load monitoring (NILM) [1] attempts to isolate transient changes in an aggregate building electricity load signal. These changes are assumed to correspond to a single appliance changing its load against a steady state background. NILM approaches using smart meter data (up to 1Hz sampling rate) commonly use edge detection algorithms in a similar vein to Hart et al. [1] to isolate events, and label the event using a limited set of features such as the real power change. High frequency sampling (>10kHz) allows a richer set of features for event classification, such as those generated by short-time Fourier transform [2]. An approach often favoured by researchers is to use high frequency features derived from the current waveform [3, 4] associated with the switched appliance.

When dealing with aggregate data it's necessary to extract a delta-current waveform, capturing the change in current load by taking the difference of the steady state before and after the event. Conversely, when appliances are submetered directly, as in public event datasets such as PLAID [5], it is sufficient to take samples of current waveform directly [6]. The current waveform can then either be plotted as a function of the voltage waveform to form an image-like "IV trajectory" [7], or be used directly as 1 dimensional features [6].

Unlike previous work which addresses either sub-1Hz data akin to smartmeter readings, or the super-10kHz regime, this paper explores the intermediate sampling range, in particular between 1Hz - 1.2kHz. We show that it is possible to accurately classify multiple classes of appliance using low frequency sampling of electrical event transients. The features we use can be seen in Figure 1, and are defined in Section 3.

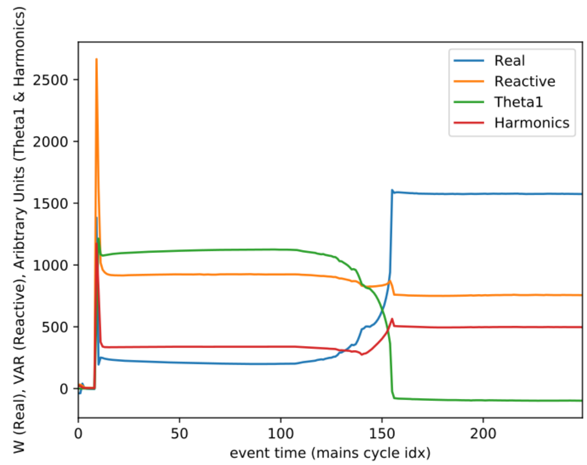


Fig. 1. Plot of transients trajectories extracted from microwave onset event. Event was buffered to 250 mains cycles, with trajectories calculated at 60Hz (4.17 seconds).

The contributions of this paper are as follows:
(i) We present a method to downsample high frequency electrical data into four channels: P (real power) Q (reactive power) θ_1 (current fundamental phase w.r.t. voltage) and H (r.m.s. of harmonics' power). P and Q are commonly used in the literature, however H and θ_1 are less often seen; they correspond to orthogonal components of reactive power, and increase the separability of different appliances.
(ii) We investigate the improvements in event classification performance as sampling rate is increased through the 1Hz - 1.2kHz range.
(iii) We compare random forest, fully connected net, and deep convolutional neural net performance on the same features. We obtain the best results when feeding the multi-variate time-series features described in (i) into a 1D convolutional neural network in a manner analogous to RGB channels.

2. RELATED WORK

Disaggregation has been traditionally applied using event classification and some form of state tracking [1], often in the form of a hidden Markov model (HMM) [8] or a deep recurrent neural network [9]. Here we focus on classifying transient events, and do not take into account the past, present or future states of a home when performing our classification. This is useful because accurate appliance event classification is the cornerstone of accurate NILM tracking systems.

Neural networks consist of layers of weak classifiers (neurons) whose activation (α) is a dot product of a set of learned weights with input features $\mathbf{w}^T\mathbf{x}$. A non-linear function is applied to this activation such as a $max(\alpha, 0)$ threshold (known as ReLU) or $tanh(\alpha)$. By stacking layers of neurons, neural networks can be trained with the backpropagation algorithm to approximate extremely complex functions. Neural networks have been applied to whole-home energy disaggregation [9] in the form of recurrent long short-term memory networks (LSTMs) [10], denoising autoencoders [11] and other recurrent networks [12].

Convolutional neural networks (CNNs) [13] were designed for signal processing tasks such as image and speech recognition. The CNN architecture embeds assumptions of translation invariances and local input features being more relevant to each other than distant ones. A number of filters are applied across the image in strides, the output of which is then fed into multiple convolutional and Maxpool (or other data aggregation) layers.

The convolutional ("Conv") layers consist of neurons that take inputs from a local patch of the input features; these input features can consist of multiple channels such as RGB channels in images. Resultantly, the networks learn to detect patterns that are common across the input channels, and the hidden layers represent maps of activation of the filters across the image. Higher levels of the network learn features composed of lower level features and have been shown to learn increasingly abstract concepts [14]. A variety of algorithms have been used for event classification in NILM, such as SVM [15], sparse coding [16] and clustering [17]. Wong et al. [18] used an alternative state-based model based on particle filtering.

Martinez et al. [19] demonstrated that a CNN may be combined with a downstream LSTM [10] in order to track appliances over time. In this paper we focus on the upstream CNN part, attempting to provide accurate appliance identification using simple, easy to collect features. Lange et al. [20] performed multiclass appliance identification using a CNN on at dataset sampled at 60Hz. Similarly, Penha et al. [21] used a CNN on data sampled at 1Hz. In Section 5, we replicate these experiment with the PLAID dataset, and show that higher frequency power measurements lead to greater appliance identification accuracy.

Barsim et al. [6] used an ensemble of neural networks, each trained to identify a single appliance on the current and voltage waveforms, obtaining a high test accuracy. We use the same dataset, and show that simpler features may be fed into a multiclass neural network and obtain similar results. We mitigate the training size issue encountered by Barsim et al. by augmenting the data with "jittered" examples, as described in Section 3.2. Similarly, Baptista et al. [7] and De Baets et al. [22] performed appliance classification using CNNs on the PLAID dataset using the current and voltage trajectories. In Section 5 we evaluate results with simpler power features, that are extracted from 1.2kHz downsampled data rather than the full 30kHz sampling.

3. DATA

We use the public PLAID [5] dataset for our experiments, consisting of labelled transient event data sampled at 30KHz with 11 appliance classes sampled from 55 houses. We investigate onset events because these contain interesting transient structures generated as components in the appliance intialise by charging, heating up, spinning up etc. We do not focus on offsets because, with a few exceptions, most appliances switch off with a sudden step down to zero as power is cut; as such we do not expect to be able to separate classes based on their offset transients (with a few exceptions such as washing machine motors). We therefore select the 1,299 of PLAID's 1,793 events that are labelled as "off-on" type. The dataset is gathered in the USA, where mains voltage is supplied at 60Hz.

3.1. Feature Engineering

The real power in one mains cycle P is calculated by summing the N data points of instantaneous power within a mains cycle: $P = \frac{1}{N}\sum_{n=1}^{N} v_n i_n$. We first downsample PLAID events to 1.2kHz, resulting in $N = 20$ samples per mains cycle. From this, we extract real power P for each mains cycle, which constitutes our 60Hz data. Our 30Hz, 12Hz, 6Hz, 2Hz and 1Hz data are obtained by instantaneous downsampling of the 60Hz P data. These frequencies and features simulate the data from smart meters, which provide only real power.

For comparison against data from high frequency sampling hardware, we generate additional power features from 1.2kHz data. We calculate three additional channels: reactive power in a mains cycle $Q = \sqrt{|S|^2 - P^2}$ where v_n is the voltage reading at time n, i_n is the current reading at time n, and the apparent power in the mains cycle S (not used as a feature) is $|S| = \frac{1}{N}\sqrt{\sum_{n=1}^{N} v_n^2}\sqrt{\sum_{n=1}^{N} i_n^2}$. We extract θ_1, defined as the difference in phase angles between current and voltage fundamentals $\theta_1 = \angle I_1 - \angle V_1$. We extract an RMS harmonic magnitude channel H as $H = H_{2:nyquist} = \sqrt{I_2^2 + I_3^2 + \cdots + I_{nyquist}^2}$ where I_k is the magnitude of the k-th current harmonic in the mains cycle, i.e. the current frequency components at {120Hz, 180Hz, ...}, with $k = nyquist$ being the Nyquist frequency, which is 600Hz in our 1,200Hz sampling regime. We note that H is the numerator of the equation for total harmonic distortion, as used by Roos et al. [23].

See Figure 1 for an example plot of $\{P, Q, \theta_1, H\}$ features extracted from a microwave cycle.

For each channel C in our feature sets, we take the first derivative C' and add it to the feature set as a sister channel. This results in our 1Hz - 60Hz feature sets each consisting two channels $\{P, P'\}$ and our full 1.2kHz data being downstampled into 8 channels $\{P, P', Q, Q', \theta_1, \theta_1', H, H'\}$ which also have a 60Hz sample rate (US mains frequency).

3.2. Data Augmentation

We augment the data by a method we refer to as jittering, to create additional copies of events from under-represented classes. The augmented events are translated rightwards in time by an amount randomly sampled a uniform distribution over the range of [0%,+20%] of the event duration. The "hole" created at the beginning of the event is padded by repeating the first sample; any excess samples extending beyond the 250 sample limit of the event are dropped. The number of augmented events created from each training event is based on the overall class count: after jittering, classes are balanced in the training set. Including jittered events, a total of 2,420 events are used for training. Jittered events are not used in the test set. Our motivations for applying this data augmentation are (i) to address class imbalance in PLAID and (ii) to provide regularisation to the data; the data generating process is random, and the same electrical event might be captured by the data gathering system with some degree of translation in time.

4. METHODOLOGY

Here we define our experimental methodology, research questions (RQs) and network architecture.

ConvNet Configuration			
5layer_MaxPool1	7layer_MaxPool2	9layer_MaxPool3	11layer_MaxPool4
input (2 or 8 channels, at 1 - 60Hz)			
Conv1D-16	Conv1D-16	Conv1D-16	Conv1D-16
Conv1D-16	Conv1D-16	Conv1D-16	Conv1D-16
Conv1D-16	Conv1D-16	Conv1D-16	Conv1D-16
MaxPool	MaxPool	MaxPool	MaxPool
	Conv1D-32	Conv1D-32	Conv1D-32
	Conv1D-32	Conv1D-32	Conv1D-32
	MaxPool	MaxPool	MaxPool
		Conv1D-48	Conv1D-48
		Conv1D-48	Conv1D-48
		MaxPool	MaxPool
			Conv1D-64
			Conv1D-64
			MaxPool
FC-1024			
Dropout			
FC-11			
soft-max			

Fully Connected Net Configuration			
2layer_FC1	3layer_FC2	4layer_FC3	5layer_FC4
input (2 or 8 channels, at 1 - 60Hz)			
Flatten			
FC-1024	FC-1024	FC-1024	FC-1024
Dropout	Dropout	Dropout	Dropout
	FC-1024	FC-1024	FC-1024
	Dropout	Dropout	Dropout
		FC-1024	FC-1024
		Dropout	Dropout
			FC-1024
			Dropout
FC-11			
soft-max			

Table 1. Network architectures for convolutional neural net (ConvNet), left, and fully connected net, right.

4.1. RQ1: How well do random forest, fully connected net and convolutional deep net perform appliance classification?

In order to establish a baseline, we first train a RF on the same features as we will be using throughout the paper on our 8 features channels extracted at 1.2kHz. These features, defined in Section 3, are: $\{P, P', Q, Q', \theta_1, \theta_1', H, H'\}$. We tune RF parameters manually to attain better performance, although parameter optimisation techniques are not used as this is deemed out of scope of this study. We compare the results of the RF with CNNs consisting 5, 7, 9 and 11 layers and FC-nets with 2, 3, 4 and 6 layers, with the architectures defined in Section 4.3.

4.2. RQ2: Do high frequency transients make appliance classes easier to separate?

We now investigate the level of performance that is attainable using features that are available from smart meter data, specifically $\{P, P'\}$ as defined in Section 3 at 1Hz, 2Hz, 6Hz, 12Hz, 30Hz and 60Hz frequency. We compare the results from the best FC-net and CNN architectures with those of the RF.

4.3. Network Architecture

The architectures we use for CNNs and FC-nets are presented in Table 1. We choose to base our CNN architecture on VGGNet [24] because of its simplicity and empirical success in a wide-range of domains. VGGNet exclusively uses kernels with a width of 3 and a stride of 1 and "valid" padding. Pairs or triplets of Conv layers are followed by a Maxpool. This strategy led to stronger performance than previous work [25] which had used filters with much larger receptive fields near the input. VGG leaves the learning of features that span a wider receptive field to deeper layers.

In our networks, all trainable layers use batch-normalisation [26] ("batchnorm") followed by a ReLU activation, except for the final classification layer which uses softmax and does not use batchnorm. Like VGGNet, our network architecture consists of blocks of two or three convolutional layers followed by a MaxPool. After a number of Conv-Maxpool blocks, all of our CNNs have final layers: a flatten layer, a fully connected hidden layer with batch normalisation and dropout, then finally a softmax classification layer for outputting confidences for the 11 PLAID classes. Our FC-nets use a number of fully connected layers with 1024 neurons, all using ReLU activations, batch normalisation and dropout. A final softmax classifier with 11 classes, one for each appliance in the dataset, follows.

4.4. Training

PLAID [5] consists electrical event data recorded from 11 appliance classes from 55 houses. Much past work used hold-one-house out cross-validation, resulting in training 55 folds, and did not use any validation set. As a trade-off with runtime, we choose to use 11 folds. We also choose to use a validation set so that training rates can be adjusted when models begin to overfit. Our pipeline splits the houses in a 9-1-1 ratio, such that each fold iteratively takes events from 45 houses for training, 5 houses for validation and 5 houses for testing. The test set predictions for the 11 folds are accumulated, and Precision, Recall and F1 score per-class are calculated.

We use Keras with Tensorflow backend and Adam optimiser, running on an Nvidia GTX1080 GPU. An advantage of our approach using 1D representations of events is that our CNNs are small enough for full-batch learning on the PLAID dataset and the whole training set can be passed through our models within the space of the GTX1080's VRAM. Our initial training rate is 0.0015, and we make use of a Keras callback to reduce the learning rate by 70% whenever a plateau in validation accuracy lasting longer than 20 epochs is observed. If a plateau of longer than 70 epochs is observed, training is halted early, otherwise it is allowed to continue for 500 epochs. As our classification layer uses softmax activation, we apply cross-entropy loss.

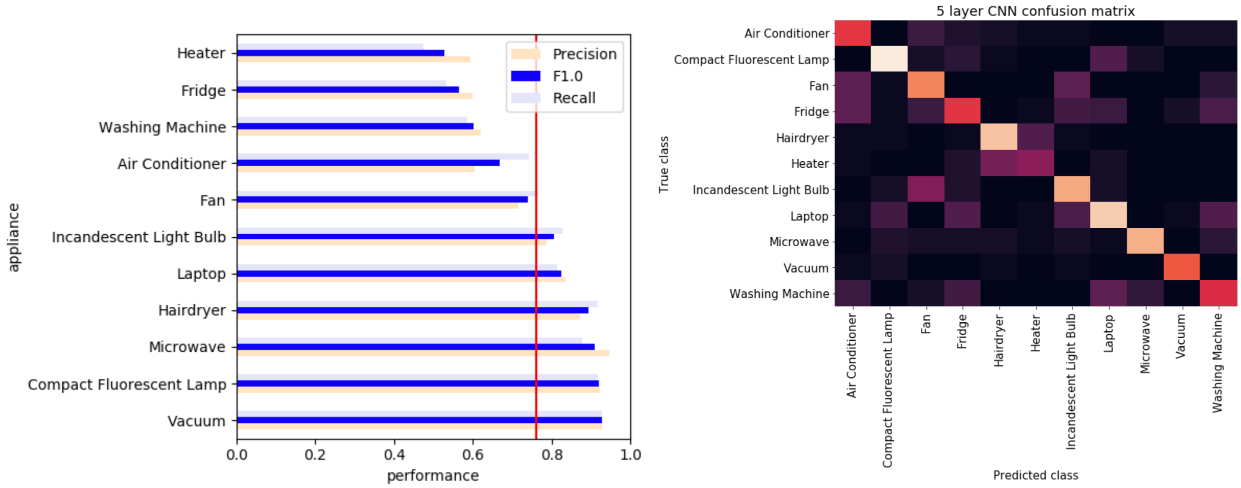
5. RESULTS

We present a comparison of results between random forest, FC and CNN at a random of frequencies in Figure 3. This contains results for RQ1 and RQ2.

5.1. RQ1: How well do random forest, fully connected net and convolutional deep net perform appliance classification?

From the final column (1.2kHz) of Figure 3 we can see that the best performing model is a 5-layer CNN with one Maxpool layer, which achieves the best performance F1-score of 0.7619. All CNNs perform significantly better than the RF (0.6921) and the best performing FC-net (the 4-layer which scored 0.7021).

A per-class breakdown of the best performing CNN model can be found in Figure 2. We observe best performance for devices with visually unique looking onset events, such as microwave (example onset plotted in Figure 1), vacuum and compact fluorescent lamp.



(a) Class F1 scores, red bar shows macro-average F1 score.

(b) Log-transformed confusion matrix, brightness indicates density.

Fig. 2. Per-appliance F1 scores and confusion matrix of best performing CNN model with 5 convolutional layers at 1,200Hz.

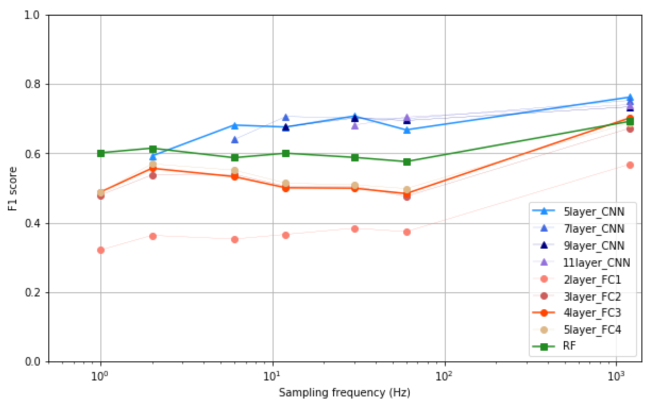


Fig. 3. Comparison of model performance across the range of sampling frequencies. Lines for the best preforming CNN and FC-net are shown in bold.

The models all perform poorly on the heater class, however, which is confused with hairdryer. This is because heater onset events are generated by single heating element turning on, which corresponds to a simple step shaped transient. Such appliance classes are very difficult to separate since fundamentally they contain a heating element whose onset appears as a plain step (e.g. hairdryer, which is composed of a very lower power fan and a high power heater).

5.2. RQ2: Do high frequency transients make appliance classes easier to separate?

In Figure 3 we plot the results for each of the models at different sampling frequencies. As discussed in Section 4, frequencies lower than 1.2kHz contain $\{P, P'\}$ only to simulate smart meters of increasingly high sampling rates, and the 1.2kHz data contains $\{P, P', Q, Q', \theta_1, \theta_1', H, H'\}$ channels for comparison against high frequency features. We can see that F1-score performance for all models degrades substantially when the frequency is dropped to 60Hz (thus removing the extra features).

At intermediate frequencies, we observe contrasting model behaviour. CNNs benefit as sampling frequency of real power is increased, whilst the RF and FC-nets seem unable to learn from these features, with performance either plateauing or falling through the 1Hz - 60Hz range. CNN models perform consistently better than the RF and FC-nets, with the exception of the 2Hz sampling frequency. In the 1Hz case, CNN architectures are not used; at these very slow sample rates the input is only 5 pixels long, and the Max-pool and Conv layers reduce the feature activation maps to below 1 pixel length. At this frequency, the RF achieves an F1-score of 0.6014, whereas the best FC drops off to 0.4878.

6. CONCLUSIONS AND FUTURE WORK

Appliance classification is a key component of event-based NILM, as accurate information must be fed to downstream tracking models such as hidden Markov models or recurrent neural networks. In general, our results show that appliance classification is possible to some extent at low "smart meter" like sampling frequencies, but that when using CNNs performance increases greatly with sampling resolutions. Further improvements in performance are available if higher sampling frequencies are used to obtain additional power features which can distinguish appliance transient signatures. In our case, the addition of 60 Hz Q, θ_1 and H channels (calculated from 1,200Hz i, v samples) significantly improved the ability of all models to separate classes.

Our CNN architectures show good separation of appliances on the PLAID dataset, with a peak mean F1-score of 0.7619. We find that 1D CNN architectures with multiple current-voltage derived feature channels as input can provide good appliance transient classification performance, substantially better than RF or FC-nets. We find that for CNNs, increasing real power sampling frequency yielded improved classification performance in Section 5, however FC-nets and RF did not appear to benefit. We believe this is caused by CNNs' inherent translation invariance, which FC-nets and RF lack.

The current state of the art F1 for PLAID event classification is 0.88 [6] with ensembles of one-vs-one models and 0.78 with a single 2D CNN model [7], both using raw current features at 30kHz. We show that similar single-model performance is possible even with a 1D CNN taking set of features derived from a lower sampling frequency when using a suitable neural network architecture.

References

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

[2] Jonathan Allen, "Short term spectral analysis, synthesis, and modification by discrete fourier transform," *Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 3, pp. 235–238, 1977.

[3] Nasrin Sadeghianpourhamami, Joeri Ruyssinck, Dirk Deschrijver, Tom Dhaene, and Chris Develder, "Comprehensive feature selection for appliance classification in NILM," *Energy and Buildings*, vol. 151, pp. 98–106, 2017.

[4] Lucas Pereira and Nuno Nunes, "A comparison of performance metrics for event classification in Non-Intrusive Load Monitoring," *International Conference on Smart Grid Communications*, vol. 2018-Jan, pp. 159–164, 2018.

[5] Jingkun Gao, Suman Giri, Emre Can Kara, and Mario Bergés, "Plaid: A public dataset of high-resolution electrical appliance measurements for load identification research: Demo abstract," in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, 2014, pp. 198–199.

[6] Karim Said Barsim, Lukas Mauch, and Bin Yang, "Neural Network Ensembles to Real-time Identification of Plug-level Appliance Measurements," 2018.

[7] Darío Baptista, Sheikh Shanawaz Mostafa, Lucas Pereira, Leonel Sousa, and Fernando 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, pp. 2460, 2018.

[8] J Zico Kolter and Tommi Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," in *Artificial Intelligence and Statistics*, 2012, pp. 1472–1482.

[9] Jack Kelly and William Knottenbelt, "Neural nilm: Deep neural networks applied to energy disaggregation," in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, 2015, pp. 55–64.

[10] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[12] Hong-Tzer Yang, Hsueh-Hsien Chang, and Ching-Lung Lin, "Design a neural network for features selection in non-intrusive monitoring of industrial electrical loads," in *11th International Conference on Computer Supported Cooperative Work in Design*. IEEE, 2007, pp. 1022–1027.

[13] Yann LeCun, Yoshua Bengio, et al., "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, pp. 1995, 1995.

[14] Matthew D Zeiler and Rob Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[15] Takekazu Kato, Hyun Sang Cho, Dongwook Lee, Tetsuo Toyomura, and Tatsuya Yamazaki, "Appliance recognition from electric current signals for information-energy integrated network in home environments," in *International Conference on Smart Homes and Health Telematics*. Springer, 2009, pp. 150–157.

[16] J Zico Kolter, Siddharth Batra, and Andrew Y Ng, "Energy disaggregation via discriminative sparse coding," in *Advances in Neural Information Processing Systems*, 2010, pp. 1153–1161.

[17] Hong Yin Lam, GSK Fung, and WK Lee, "A novel method to construct taxonomy electrical appliances based on load signaturesof," *Transactions on Consumer Electronics*, vol. 53, no. 2, 2007.

[18] Yung Fei Wong, T Drummond, and YA Şekercioğlu, "Real-time load disaggregation algorithm using particle-based distribution truncation with state occupancy model," *Electronics Letters*, vol. 50, no. 9, pp. 697–699, 2014.

[19] Roberto Martinez, Daniel Pacheco, and Dane Robergs, "NILM Power Disaggregation via Artificial Neural Networks," pp. 1–4.

[20] Henning Lange and Mario Bergés, "The neural energy decoder: energy disaggregation by combining binary subcomponents," in *3rd international workshop on non-intrusive load monitoring*, 2016.

[21] Deyvision de Paiva Penha and Adriana Rosa Garcez Castro, "Convolutional Neural Network Applied To the Identification of Residential Equipment in Non- Intrusive Load Monitoring Systems," pp. 11–21, 2017.

[22] Leen De Baets, Joeri Ruyssinck, Chris Develder, Tom Dhaene, and Dirk Deschrijver, "Appliance classification using VI trajectories and convolutional neural networks," *Energy and Buildings*, vol. 158, pp. 32–36, 2018.

[23] JG Roos, IE Lane, EC Botha, and Gerhard P Hancke, "Using neural networks for non-intrusive monitoring of industrial electrical loads," in *Proceedings of the 10th Anniversary Conference on Instrumentation and Measurement Technology Conference*. IEEE, 1994, pp. 1115–1118.

[24] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[26] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.