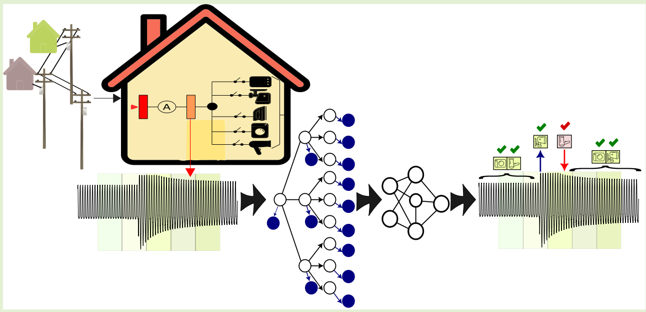
ST-NILM: A Wavelet Scattering-Based Architecture for Feature Extraction and Multilabel Classification in NILM Signals

Everton Luiz de Aguiar, Lucas da Silva Nolasco, André Eugenio Lazzaretti, *Member, IEEE*, Daniel Rodrigues Pipa, *Member, IEEE*, and Heitor Silvério Lopes



***Abstract*—Nonintrusive load monitoring (NILM) is a relevant tool for improving energy consumption habits, contributing to energy conservation and distribution system planning. In recent years, high-frequency strategies using deep learning have been presented in the literature, achieving the state-of-the-art results for detection, feature extraction, and classification of aggregated electrical loads, particularly with the architecture defined as deep neural network model for detection, feature extraction, and multilabel classification (DeepDFML). DeepDFML used a deep convolutional network (DCN) whose trained weights were shared for different output fully connected networks. The performance of DeepDFML depended on the availability of data and data augmentation (DA) strategies. Given this scenario, we propose the ST-NILM, a new integrated architecture based on the scattering transform (ST). ST-NILM has a DCN with analytical wavelet-based nontrained weights, shared with fully connected output networks that perform event detection and multilabel classification of aggregate loads. We compared ST-NILM and DeepDFML for the LIT-SYN dataset. ST-NILM achieved equivalent detection results to DeepDFML for two and three aggregated loads and performed better for single loads. The hardware implementation shows that ST-NILM consumes less memory, less GPU load, and substantially less computational effort than DeepDFML. ST-NILM presents comparable or even superior results than other state-of-the-art deep-learning-based methods.**

***Index Terms*— Deep learning, multilabel classification, nonintrusive load monitoring (NILM), wavelet scattering.**

I. Introduction

NONINTRUSIVE load monitoring (NILM), initially presented by [1], consists in extracting information from each appliance in a house, by observing electrical quantities (voltage, current, or power) aggregated to the energy input. In the NILM approach, individual consumption information is obtained for each appliance without using sensors for each piece of equipment. Separate information may provide consumers with: 1) knowledge of their own consumption habits; 2) analyze consumption in real-time and in historical series; 3) prospect future consumption; and 4) plan and distribute the use of more powerful loads (electric heating, air conditioning, etc.). As a result of this, one could save energy consumption. Two important tasks regarding NILM are: *disaggregation* and *classification*. The disaggregation separates each load curve (typically, current or power) from an aggregated single signal (in most cases located at the switchboard). The classification corresponds to identifying which appliances are turned on/off and when these events happened from the analysis of the single aggregated electrical signal.

One of the most relevant stages of feature extraction is defining the power signature (PS). PS is a particular representation that characterizes the behavior of each appliance based on their features, and it depends on the feature extractor chosen. The first PSs considered only changes (events) that occurred after a time interval in the aggregate power curve that was in a steady-state. These approaches were limited to linear loads and restricted to loads with an operating regime with varying power (electric irons, heaters,

Manuscript received 20 November 2023; revised 23 January 2024; accepted 27 January 2024. Date of publication 13 February 2024; date of current version 2 April 2024. The work of André Eugenio Lazzaretti was supported by the CNPq for the Research under Grant 306569/2022-1. The associate editor coordinating the review of this article and approving it for publication was Dr. Olga Fink. *(Corresponding author: Everton Luiz de Aguiar.)*

The authors are with the Department of Electronics, Federal University of Technology–Paraná, Curitiba 81280-340, Brazil (e-mail: evertonaguiar@alunos.utfpr.edu.br; lucasnolasco@alunos.utfpr.edu.br; lazzaretti@utfpr.edu.br; danielpipa@utfpr.edu.br; hslopes@utfpr.edu.br).

Digital Object Identifier 10.1109/JSEN.2024.3360188

etc.). However, using these early methods was challenging for determining the PS of switched or nonlinear electronic circuits [2].

In addition to the above-mentioned early methods, the PS can be obtained through the features extracted from the electrical time signal (voltage, current, or energy). Ibrahim et al. [3] classified NILM methods into model-based or data-driven. Based on the idea proposed in [4], one can categorize the feature extractors for data-driven NILM methods into: conventional physical definitions (CPDs), time–frequency analysis (TFA), and voltage–current ($V-I$)[1] trajectories. More recently, deep convolutional network (DCN) model-based NILM methods have gained attention in the recent years because the extracted features are highly discriminative, and they are not engineered (no need for preknowledge about the appliances) [5].

The DCN methods are composed of convolutional neural networks (CNNs). CNNs have convolutional filters whose weights are trained. This training process requires large amounts of data and computational power. Furthermore, the structure of DCN topologies is generally determined empirically, and their coefficients have no physical meaning. On the other hand, in [6], the scattering transform (ST) was proposed, which is a time-shifting-invariant and stable to small time-warping operator. The ST is similar to CNN [7], sharing its advantages (high discriminability), but with the following advantages: 1) there is no need for weight training; 2) the ST coefficients have physical meaning (location in time and frequency), which is particularly interesting in the NILM problem; and 3) as a consequence of the coefficients not being trained, ST needs less training data than CNN.

The above-mentioned advantages of DCN methods combined with the ST encouraged us to propose a framework for extracting features from high-frequency NILM signals, using a new architecture with the ST replacing the CNN in a multitask application, called ST-NILM. Hence, the main contributions of this article are given below.

1) We propose a single framework that integrates detection and classification of aggregated loads.
2) Our proposal uses a convolutional network that does not need a training step, thus requiring less data and no need for data augmentation (DA) approaches.
3) Unlike other state-of-the-art approaches, our proposal uses a multilabel nontrainable feature extraction framework to disaggregate and classify multiple aggregated loads.
4) The architecture of the proposed convolutional network is determined analytically through the selection of the appropriate ST parameters for NILM.
5) Our proposal requires less computational cost than the state-of-the-art methods, with superior results for most cases.
6) An ablation study on ST-NILM parameters that evaluates the impact of varying the ST parameters on the classification and detection of electrical loads.
7) An embedded version of our proposed framework was also developed for demonstrating its feasibility for a real-time application.

The remaining of this article is organized as follows. We present the related works in Section II. We introduce ST-NILM in Section III, highlighting all the steps that compose it. We show the preprocessing process in Section III-A and present the scattering network (and its parameterization) in Section III-B. We detail the dataset used in Section IV. The results obtained are shown in Section V, including an ablation study on the ST-NILM parameters and the embedded implementation of our proposed framework. Finally, general conclusions and future works are presented in Section VI.

II. Related Works

DCN architectures are categorized according to the data sampling frequency. Several works use low-frequency datasets (sampling frequency below 3 Hz) to train CNN architectures for disaggregation or NILM classification [8], [9], [10], [11], [12], [13]. High-frequency strategies, main focus of this work, using DCN can also be found in [14], [15], [16], [17], [18], [19], [20], [21], [22], and [23].

In different feature extraction methods, a 2-D feature image is generated from the 1-D NILM signal, allowing the use of well-known image processing and deep learning techniques for NILM classification. In [14], the 2-D image was generated by a time–frequency short-time Fourier transform (STFT). The spectrogram was applied as the input of a CNN, particularly designed for that work. The good location both in time and frequency allowed to deal with nonstationary multicomponent signals, but some classification accuracy results were below the average of other methods, i.e., around 70% for the PLAID dataset. A similar approach is presented in [15], in which spectrograms obtained from an STFT were used as input of the CNN. The strategy was devised to filter out background noise caused by other loads in the target load. In [16], the 2-D representation was weighted pixelated $V-I$ images, obtained from the normalized $V-I$ curve in steady-state. The image was then inserted as the entry of a CNN, which performs the classification. The overall FScore was also below the average of the other compared methods (<78%), and the authors needed to use two datasets together (PLAID and WHITED) to reach those results. Mulinari et al. [24] proposed to use the 2-D Fourier transform to extract features from the $V-I$ curve of electric loads. FScores of up to 97.7% were achieved by [24], but the proposed method did not contemplate load detection and disaggregation.

Two multilabel approaches were presented in [17] and [18]. Both have the advantages of being multilabel classification strategies, presented as alternative approaches to the traditional 2-D image applied to the input of a CNN. In [17], a transition event is first located, followed by the Fryze power theory in the aggregated current to extract the features together with a similarity matrix based on the Euclidean distance to reinforce the discriminability. A 2-D image is then generated, being the input of a CNN with multilabel classification. In [18], on the other hand, a multilabel approach was proposed to improve the classification of loads of the same type but different

[1] We represent electrical current with the capital letter “I” to maintain compatibility with related literature.

brands. Given one cycle of the voltage and current, a weighted recurrent graph (WRG) generates a 2-D image for posterior classification. Accuracy results were better than other baseline methods that produce 2-D images from V–I trajectories, but still lower than those presented in [17] (both for the PLAID dataset). Moreover, the results of [18] were obtained from submetered data (i.e., not aggregated).

Accuracy results above 98% were obtained in [19], which presented a technique called 2-D phase encoding (2DPEP) to generate a 2-D image from the NILM signal. Using time-domain feature extractors, the authors overcame the state-of-the-art classification metrics for several distinct datasets based on sliding windows. Despite these promising results, Himeur et al. [19] had a high dependency on the event detection algorithm and the proposed approach used other classification methods based on classical machine learning, increasing the complexity. Chen et al. [25] proposed both temporal and spectral to define a dual PS for each load. Each PS, represented as 2-D image, enters a DCN structure followed by a fully connected network. The FScore results overcame the V–I method, but the authors did not compare these results with another CNN-based approach. Furthermore, the proposal of [25] was time-shifting-covariant.

In [20], a discrete wavelet transform to obtain a 2-D image from the aggregated current signal is proposed. The authors used the image as an input to a sequence-to-sequence CNN. The wavelet transform has the advantage of time-warping stability, but the data used in [20] were submetered, and not naturally aggregated. The same limitation is found in [20] and [21]. Particularly, Moranin et al. [21] used a deep convolutional autoencoder to extract features from individual hospital loads. Nevertheless, there is no disaggregation since the CNN input is obtained from a submetering network. In [22], a multiagent strategy was proposed to improve NILM classification, achieving accuracy results above 95% on the LIT-dataset. Although this result was superior to the related literature, applying the multiagent strategy in realistic cases may be compromised due to the high computational complexity. Mukaroh et al. [23] also presented a CNN-based classification model for NILM, reaching 92% global accuracy on the LIT-dataset, but with limitations in the feature extraction for loads with similar transitory shapes.

A real-time CNN-based method proposed in [26] reached up to 99.2% accuracy, with a 100-Hz sampling frequency data. This method uses a three-stage structure: 1) event detection; 2) CNN classification; and 3) power estimation, applying machine learning to detect the turn-on events and an heuristic algorithm to estimate the real-time power. However, there were some limitations, such as: 1) authors used a private dataset, which reduces reproducibility; 2) only three appliances of greater power consumption were used in the tests; and 3) the algorithm had problems in identifying loads with steep step-up transients.

Gomes and Pereira [27] applied a pinball loss function (PB) to different DCN architectures and compared the results with the mean squared error (MSE) loss function. The authors performed experiments with manually summed data and native aggregated data. Both with pinball loss function and MSE, all the results were better with the sum of loads. As the DCN was not time-shifting-invariant, Gomes and Pereira [27] reported problems with time-shifting signals. Jia et al. [28] extracted features with bidirectional dilated convolution networks, increasing the length of receptive fields. Although the results outperformed other methods, the authors did not address the computational cost of training, and the noncausal attribute of the extractor did not allow real-time application. Both [29] and [30] applied long short term memory (LSTM) to disaggregate NILM loads. Hwang and Kang [29] and Laouali et al. [30] presented promising applications for real-time. However, the lack of reproducibility of the private dataset of [30], the average FScore below 90% of [30], and the low-frequency data features' discriminability are challenging to overcome.

Mallat [6] proposed the ST to define a time–frequency representation of a 1-D signal. ST has an architecture analogous to a CNN. However, in the ST, the filter coefficients are not trained but are analytically determined using wavelets. In addition, the structure of the ST (number of layers and convolutional filters) can also be determined analytically. As a consequence, ST needs less training data than CNN for classification tasks [7], [31]. ST is invariant to time-shifting and stable to time-warping, and these properties are particularly interesting for classifying electrical charges [32]. Some classical feature spectral extraction methods, such as the STFT [33], were time-shifting-invariant representations but were not stable to time-warping [6]. Methods based on Mel-Spetrograms [34] extracted features more closely related to human perception of sound, but their particular filter bank structure may not adequately represent electrical signals. Two reassignable synchrosqueezing approaches were presented by [35] and [36]. Daubechies et al. [35] proposed the synchrosqueeed wavelet transform (SSWT), which was based on the continuous wavelet transform (CWT). They added CWT components with the same instantaneous frequency to obtain a more focused representation. The method [35] considerably decreased the dimensionality of the CWT representation but depended on prior knowledge of the nature of the input signal. In addition, unlike ST, SSWT method [35] was a time-shifting-variant representation. Oberlin et al. [36] proposed a synchrosqueezing method based on the STFT, named Fourier-based synchrosqueezing transform (FSST). The approach of [36] outperformed the existing TFA techniques in terms of its ability to extract detailed information about the frequency content of a signal. However, unlike ST, FSST was not stable to time-warping. Michau et al. [37] proposed the denoising sparse wavelet network (DeSpaWN) method, composed of a cascade architecture using a deep learning framework. The DeSpaWN method, unlike SSWT [35], did not need a prior, as the coefficients were learned, and hard-thresholding allowed for a sparse representation. DeSpaWN, however, requires training a reduced number of parameters, unlike ST.

Several authors have used ST in areas other than NILM. For instance, Wang et al. [38] proposed a framework for recognizing musical performance techniques, reaching Global FScores up to 79.9%. Souli et al. [39] extracted features with ST and used a DCN to classify voices, reaching the state-of-the-art accuracy results of up to 99.62%. de Aguiar et al. [7] and Aguiar et al. [32] proposed approaches based on ST to extract features and classify NILM signals. Aguiar et al. [32] evaluated the performance of ST under nonideal dataset conditions: subsampling, reduced number of examples, and reduced number of measurements. The results outperformed the DWT and V–I methods for most of the analyzed scenarios, but the authors did not propose an event detection method.

Finally, the deep neural network model for detection, feature extraction, and multilabel classification (DeepDFML)-NILM [40] uses a DCN for detection, feature extraction, and multilabel classification of high-frequency NILM signals for the publicly available LIT-dataset. The detection results were above 90% for most cases, while the classification accuracies were comparable with other state-of-the-art works (around 97%). However, due to the high number of parameters, the proposed method requires DA strategies, in addition to significantly increasing the computational complexity.

Despite the relatively good results presented so far, some of the main limitations with the architectures based on DCN applied to NILM are given below.

1) The proposed architectures generally do not integrate detection, disaggregation, and classification, i.e., the network performs only one or two of these tasks —with the exception of DeepDFML [40].
2) As they use deep learning techniques, these solutions demand a large amount of data for training, since the computational effort grows with the increase in the number of coefficients (weights) of learned convolutional filters. The ST, on the other hand, contributes to reducing the computational effort, by replacing the convolutional weights by analytical wavelet coefficients. Hence, this work proposes an original and unified architecture for high-frequency NILM signals, combining the advantages of ST with DCN approaches for signal detection (identification) and classification.

III. Proposed ST-NILM

Since our approach is based on the DeepDFML [40], we first present in this section a brief overview of that architecture. The DeepDFML was inspired by YOLO [41], and it detects the instant in time when each electrical load is turned on or off and predicts the combination of loads present in the input signal. The architecture of DeepDFML has a shared network with five layers and several learned coefficients. The size of the convolutional filters, the number of layers, the pool size, and all other parameters were empirically determined.

We present DeepDFML in Fig. 1(b). The DeepDFML method has a shared convolutional network after the preprocessing stage and three subnetworks (one for each task) at the output. Two subnetworks deal with classification tasks (event type and load identification). The event-type subnetwork performs the binary classification of the turn-on or turn-off event of an appliance. The load identification subnetwork performs the multilabel binary classification of the aggregated appliances in the input sample. The event sample detection subnetwork deals with a regression task, and its output is a real number from 0 to 1 that informs the event's position inside a grid time interval.

Fig. 1(a) shows the proposed ST-NILM architecture. In the ST-NILM, we replace the shared DCN of the DeepDFML with the scattering network. The scattering network is also a convolutional network but, unlike CNN, the filters are fixed, not learned. Fig. 1 shows the proposed ST-NILM architecture. The most relevant differences between our proposal (ST-NILM) and DeepDFML-NILM are given below.

1) ST-NILM does not have trained weights on the shared convolutional network [named scattering network in Fig. 1(a)].
2) The determination of the scattering network weights is analytical (determined, in our proposal, by Morlet wavelets) and, therefore, it is possible to calculate them a priori; we perform load detection through the event-type classification network. This network detects whether the load has been turned on or off within a grid interval (less than 166 ms). We consider this detection resolution sufficient, so we do not use DeepDFML's event sample detection network on ST-NILM.

We detail all the parts that compose the proposed architecture as follows.[2]

A. Preprocessing

In ST-NILM, we use the same preprocessing strategy used in DeepDFML-NILM. To illustrate the preprocessing strategy proposed in [40] and used for ST-NILM, we present the block diagram of Fig. 2.

In Fig. 2, the green blocks represent the original time series samples contained in the LIT-SYN subset, and the red blocks represent the preprocessing stage. To facilitate direct results' comparisons, we apply the data segmentation proposed in [40]. In this strategy, the original LIT-SYN subset signals are cut into smaller sections containing centered segments of approximately 50 cycles of 60 Hz plus two unmapped margins (15% of the centered segment at each edge of the whole segment). The cutoff points of the original signals come from the electrical loads connection annotations contained in the original subset. Each output segment of the cutting block must have at least one turn-on event. The 50-cycle segments go through a grid separation block (Fig. 2), which divides them into five regions of 10 cycles, each called grids. From the labels and events' annotations originally contained in the dataset, we determine multitask labels for each of the grids as follows.

1) *Event-Type Label:* 3-bit binary one-hot encoding that determines whether a load has turned on or off in the range of a grid;
2) *Multilabel Classification:* Binary one-hot encoding that indicates which loads are connected in the range of a grid.

B. Scattering Network

The proposed method replaces the shared deep CNN applied in [40] with the scattering network, which has no learned

[2] All the codes for ST-NILM are publicly available at https://github.com/LucasNolasco/ST-NILM

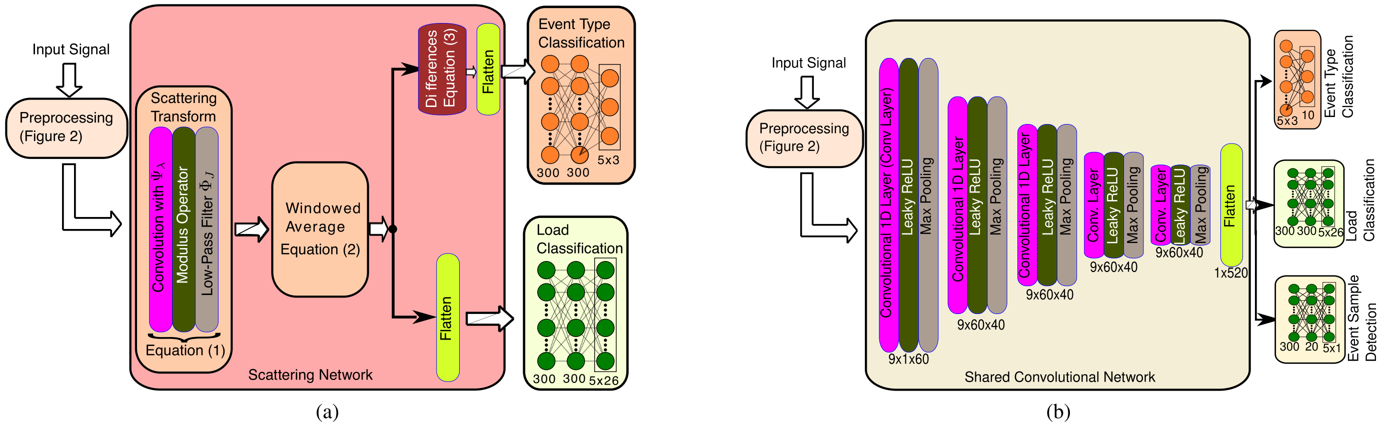


Fig. 1. (a) Proposed ST-NILM and (b) DeepDFML architectures: the proposed ST-NILM architecture is inspired by DeepDFML, but by replacing the shared convolutional network of (b) by the scattering network of (b) we also adapt the fully connected network for event-type classification to obtain better results. (a) Proposed ST-NILM. (b) DeepDFML, proposed by [40].

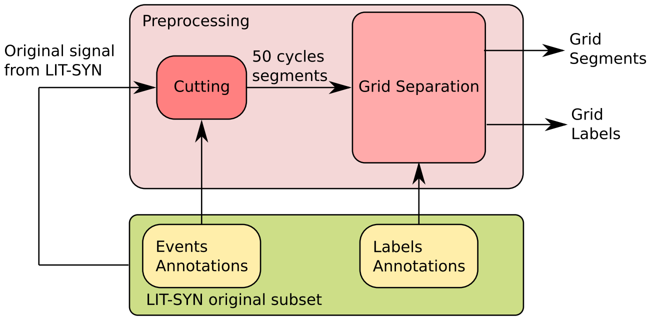


Fig. 2. Preprocessing strategy applied to ST-NILM and proposed in [40].

coefficients. The path of the input signal in the scattering network block follows: 1) ST; 2) windowed average; and 3) flatten. We will address each of these steps as follows.

1) Scattering Transform: The ST has a structure similar to the DCN and represents the modulus and mean of a CWT. ST uses functional blocks that calculate the modulus of convolutional operations between wavelets and n-dimensional signals. For the case of the present work, we will use 1-D signals at the input of the ST.

Let a time-domain electrical current signal. Guth et al. [42] showed that phase collapse improves the classification performance of convolutional networks in which there are time-shifting-invariant classes (1-D case). Phase collapse is the phase information elimination performed by some nonlinearity. The ST performs phase collapse explicitly through the modulus applied to the convolution of the time-domain signal of the wavelet [42]. Real-valued CNNs, for example, perform phase collapse implicitly [42].

Each building block of ST is composed of a *convolution structure* + *modulus* + *average*. This structure is very similar to a DCN, which generally has the association of *convolution* + *nonlinearity* + *pooling*. ST uses modulus operation for the following reasons [6].

1) Modulus extracts the envelope of a complex signal, which partly explains the time-shifting invariance. Consider the modulus of the convolution of the signal $x = x[n]$ with the wavelet $\Psi[n]$, given by $|x * \Psi|$. Then $\int(|x * \Psi|))dt$ is the L1-norm, which is time-shifting-invariant.
2) The modulus is a contractive operator, increasing feature space discriminability. Mallat [6] showed that the modulus operator is the only possible nonlinearity choice with these characteristics.

Now we will define the main parameters of the ST: the bandwidth and central frequencies of the filters, the number of wavelets per octave Q, the maximum scale (J), and the order (m) (or the number of filter banks).

1) *Choosing the number of wavelets per octave Q:* The more wavelets per octave, the greater the number of ST coefficients, and the more selective the filters. Studies in the literature using ST chose Q values between 8 and 12. We chose $Q = 10$ for the ST-NILM.
2) *Choosing maximum scale J:* We choose J based on the nature of the harmonic content of electrical signals. Consider a discrete signal $x[n]$, with sampling frequency f_s and a total number of samples equal to T. So, the highest representable frequency, by Nyquist's theorem, is $(f_s/2)$. Because of that, the maximum frequency scale $(f_{\max})$ analyzed must represent a frequency lower than half the sampling frequency. Considering a previous ablation study based on the classification performance and reduced number of coefficients, we choose $J = 10$, so that the the largest filter will be concentrated in a time interval of size 2^{10} samples. Ramírez-Ramírez et al. [43] showed that the harmonic distortion of the combination of residential loads for frequencies greater than 1020 Hz (order 17) represents less than 4% of the distortion found, for example, at 180 Hz (order 3). With that in mind, we consider the choice of $J = 10$ reasonable for our application in electrical signals.
3) *Choosing order m:* Literature shows that the scattering coefficients (S) with two layers already represent up to 98% of the energy of the input signal, and all the second-order coefficients together represent only 20% of the first-order coefficients' total energy [44]. In addition, equivalent electrical appliances classification results were presented in [45] for $m = 1$ and $m = 2$. We chose $m = 1$,

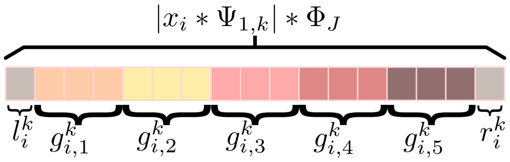


Fig. 3. Grids division for windowed average.

considering that the ST with $m = 1$ has fewer coefficients than the ST with $m = 2$, resulting in less computational effort.

2) Windowed Average: We use a windowed average to select the features shared by the two output deep networks. The first-order coefficients of the ST, in each path, are signals located in time, given by

$$S_i^k = |x_i * \Psi_{1,k}| * \Phi_J. \tag{1}$$

Fig. 3 shows the division for windowed average. Let N be the number of samples of S_i^k. We divide the range $n \in [0, N]$ into five integer parts $(g_{i,1}^k \dots g_{i,5}^k)$, centered on the same range. The remaining samples from this entire division were at the ends of the range, and we named them *unmapped region* (represented with l_i^k, r_i^k). The idea is to combine the result of the ST using the same grid division applied by DeepDFML when encoding the model outputs.

Next, the average of each $\bar{l_i^k}$, $\bar{r_i^k}$, and $\bar{g_{i,n}^k}$ segments is calculated. We define $f_n^k = \bar{g_{i,n}^k}$, $f_{i,l}^k = \bar{l_i^k}$, and $f_{i,r}^k = \bar{r_i^k}$, and therefore, the selected features for each wavelet are

$$Sf_i^k = \left[f_{i,l}^k \; f_{i,1}^k \; \cdots \; f_{i,5}^k \; f_{i,r}^k\right]. \tag{2}$$

3) Event-Type Classification and Load Classification Networks: The event-type classification network determines what type of event is found within a grid's time interval using one-hot encoding on a multiclass classification task. The input signal for this task is derived from the differences between consecutive grid interval averages. We define the differences for the kth wavelet filter and the ith sample as

$$D_i^k = \begin{bmatrix} d_{i,0}^k \\ d_{i,1}^k \\ d_{i,2}^k \\ d_{i,3}^k \\ d_{i,4}^k \\ d_{i,5}^k \\ d_{i,6}^k \end{bmatrix} = \begin{bmatrix} f_{i,1}^k - f_{i,l}^k \\ f_{i,2}^k - f_{i,l}^k \\ f_{i,3}^k - f_{i,1}^k \\ f_{i,4}^k - f_{i,2}^k \\ f_{i,5}^k - f_{i,3}^k \\ f_{i,r}^k - f_{i,4}^k \\ f_{i,r}^k - f_{i,5}^k \end{bmatrix}. \tag{3}$$

For the event-type classification, the feature set is formed by concatenating the coefficients D_i^k, $k = [0 \dots K]$ such that K is the total number of wavelets with order less than or equal to one. A flatten layer is applied to D_i^k to obtain a 1-D vector with all the selected features.

The output for this subtask has dimension $(n_{\text{grids}} \times 3)$, where n_{grids} is the number of grids, and 3 is the number of possibilities for classifying events: 1 for turn-on, 2 for turn-off, and 0 for a false event. This subtask network uses a sigmoid layer as an activation function and the categorical cross-entropy as a loss function. This cost function, in turn, takes the following form:

$$\text{CCE} = -\sum_{i=1}^{M} p\left(x_i\right) \cdot \log q\left(x_i\right) \tag{4}$$

in which M is the total number of examples, $p(x_i)$ is the expected probability of x_i to be from the class being analyzed, and $q(x_i)$ is the probability found by the model.

In addition, there is a load classification network, which serves to determine which loads are present in the sample to be classified. This network represents a multilabel classification task, which allows the identification of multiple loads simultaneously. The input signal of this network is the set of concatenated coefficients Sf_i^k, given by applying a flatten layer to Sf_i^k to obtain a 1D vector with all the selected features. The output has dimension $(n_{grid} \times n_{\text{loads}})$, where n_{grids} is the number of grids obtained from the windowed average, and $n_{\text{loads}} = 26$ is the number of electrical appliances in the LIT-SYN dataset. Since a multilabel classification was implemented, sigmoid was chosen as the activation function using the binary cross-entropy as a loss function. This cost function takes the following form:

$$\text{BCE} = -\frac{1}{M}\sum_{i=1}^{M} p\left(x_i\right) \log q\left(x_i\right) + \left(1 - p\left(x_i\right)\right) \log\left(1 - q\left(x_i\right)\right). \tag{5}$$

It is worth noting that this subtask did not require weighting to improve the classification performance.

Sections V-A and V-B provide a more detailed explanation of the training and evaluation processes.

IV. Dataset

We chose the dataset for the experiments, considering the application in real homes. Since turning on multiple appliances simultaneously in real-world homes is expected, we need a high-frequency dataset with as many aggregate loads as possible. With this, we can adequately train the ST-NILM with the annotated data. The dataset used in this work was created by the Laboratory for Innovation and Technology in Embedded Systems (LIT),[3] and it is composed of three subsets [46]: Natural, Simulated, and Synthetic. Here, we use the Synthetic subset for the following reasons: 1) the subset contains annotated samples of up to eight aggregate loads; 2) the Natural subset was not available by the time this work was done; 3) the Simulated subset does not contain real-world data; and 4) the Synthetic subset contains real-world data, from real-world appliances, with precise annotations.

The Synthetic subset (LIT-SYN) is regarding to acquisitions collected from a bench, in which real loads are connected, but the network loads' switching instant is controlled. This set contains 1664 waveform acquisitions sampled at 15.36 kHz and 12 bits using the National Instruments (NI) MyRio embedded module, with precisely annotated (< 5 ms) switching events and synchronized voltage and current waveforms. All the

[3] http://dainf.ct.utfpr.edu.br/ douglas/LIT_Dataset

files were recorded using the NI technical data management streaming (TDMS) and, then, converted into binary MATLAB files (MAT) for manipulation in Python environment. This procedure and the data acquisition are fully detailed in [46]. The aggregated ac grid voltage and current acquisitions define different subsets of LIT-SYN, depending on the number of simultaneous loads in the waveform: with single loads (LIT-SYN-1), two loads (LIT-SYN-2), three loads (LIT-SYN-3), and eight loads (LIT-SYN-8).

V. Results

This section presents the results obtained with ST-NILM, comparing them with DeepDFML and related works. In addition, we present the procedures adopted for the experiments and a discussion of the results obtained. For all the implemented baselines, we use the LIT-SYN-1, LIT-SYN-2, LIT-SYN-3, and LIT-SYN-8 subsets. All these subsets present the same sampling rate [46].

A. Experimental Setup

The dataset was divided into training and test sets, with a holdout test set containing 10% of the data. For training, with 90% of the entire dataset, we used a ten-fold cross-validation procedure. The model was trained for each fold considering a validation subset with 10% of the training subset size. The best model was chosen considering the average between the F_1 scores for binary classification on the validation subsets for each fold.

Three detection metrics were used, based on [40], as follows.

1) $PC_{\text{on}} = (A_{\text{on}}/N_{\text{on}})$ (percentage of correction of an ON event for a given method), such that A_{on} is the number of correct ON detections and N_{on} is the total number of turning on events.
2) $PC_{\text{off}} = (A_{\text{off}}/N_{\text{off}})$ (percentage of correction detection of an OFF event for a given method), such that A_{off} is the number of correct OFF detections and N_{off} is the total number of turning off events.
3) $PC_{av} = (PC_{\text{on}} + PC_{\text{off}}/2)$ (the arithmetic mean of PC_{on} and PC_{off}), which indicates the average detection performance.

For classification, we use the F1-Macro metric, given by

$$\text{F1-Macro} = \frac{1}{Y}\sum_{i=1}^{Y}\frac{2 \cdot tp_i}{2 \cdot tp_i + fp_i + fn_i} \tag{6}$$

where tp_i and fp_i are the number of true positives and false positives for appliance i, respectively, and fn_i is the number of false negatives.

B. Results Using a Reduced LIT-SYN Dataset

We present both the detection and classification metrics considering reduced subsets for the training step. We perform the analyzes on reduced datasets for the following reasons: 1) in real-world cases of disaggregation, the amount of data can be limited, so an analysis of the performance of the NILM techniques under these conditions is necessary; 2) to the best of our knowledge, related works do not address the effect of reducing the number of examples on the performance of NILM strategies; and 3) methods with better performance with reduced datasets tend to be more easily implemented in real-time.

We create additional training examples for DA tests using the high accuracy NILM detector (HAND) method [47]. These additional examples correspond to previously obtained false-positive and false-negative detections that can help the learning process and increase the computational cost.

Table I shows the detection metrics for the reduced subset scenarios for training, and without DA. The detection metrics are comparable between ST-NILM and DeepDFML for large training sets (100% and 75%). For reduced training sets (50% and 25%), ST-NILM outperforms DeepDFML on all the calculated detection metrics. We also inserted in Table I the metrics obtained with DeepDFML with DA. ST-NILM, even without DA, obtained detection results equivalent to DeepDFML with DA (maximum difference in the PC metric of 4.8%).

Fig. 4 shows the classification results obtained for reduced training datasets, comparing ST-NILM, DeepDFML [40], in addition to two baselines using STFT [33] and Mel-Spectrogram [34] instead of ST as feature extractors. From each time-series example, we use frames with 512 samples, separated by steps of 1024 samples for both STFT and Mel-Spectrogram and 128 frequency bins with an upper frequency limit of 7800 Hz for Mel-Spectrogram. We choose these parameters based on two analyses: the number of coefficients comparable with ST-NILM and the best classification performance among several sets of parameters. Note that the classification metrics are better for ST-NILM in all nondata-augmented scenarios with reduced training dataset. For comparison purposes, we also insert in Fig. 4 the FScore-Macro obtained with DeepDFML with DA. The classification result obtained with ST-NILM without DA is equivalent or higher to that obtained by DeepDFML with DA for the scenarios with more training examples (maximum difference of 2%) and lower (difference of 14.58%) for the scenario with 25% of training data. The detection performance using the spectral extractors STFT [33] and Mel-Spectrogram [34] was low (reached a maximum of $PC_{av} = 52\%$, similar to the values found by [22]), and for this reason, we chose not to include them as detection performance baselines in Tables I and II.

C. Comparison With Related Works

The most similar architecture to ST-NILM is DeepDFML [40] and, therefore, it was used as the primary baseline for comparing results. This section also compares the results obtained with other related works. The best results achieved by DeepDFML considered a DA strategy to expand the training set. This strategy increased the overall complexity of the model. On the other hand, our proposed method, ST-NILM, does not need DA. We present in Table II the detection results comparing ST-NILM with DeepDFML in all the subsets of the LIT-SYN dataset, also considering scenarios with and without DA. The results of Table II show that: 1) the best ST-NILM results are achieved without DA;

TABLE I
DETECTION RESULTS FOR REDUCED TRAINING SUBSETS

Percentage	$PC_{on}(\%)$			$PC_{off}(\%)$			$PC_{av}(\%)$		
	ST-NILM	DeepDFML		ST-NILM	DeepDFML		ST-NILM	DeepDFML	
		with DA	without DA		with DA	without DA		with DA	without DA
100%	**97.84%**	93.50%	95.10%	91.04%	96.50%	**95.90%**	94.44%	95.00%	**95.50%**
75%	**95.68%**	90.90%	83.50%	84.20%	**90.80%**	81.80%	89.94%	**90.85%**	82.70%
50%	**91.13%**	83.50%	64.10%	**82.31%**	80.20%	64.10%	**86.72%**	81.85%	64.10%
25%	**72.66%**	60.20%	51.40%	**82.08%**	73.30%	47.40%	**77.37%**	66.75%	49.30%

TABLE II
COMPARISON OF DETECTION RESULTS BETWEEN METHODS CONSIDERING EACH SUBSET OF THE LIT-SYN DATASET

Method	**Subset**	$PC_{on}(\%)$ **Data augmentation?**		$PC_{off}(\%)$ **Data augmentation?**		$PC_{av}(\%)$ **Data augmentation?**	
		No	Yes	No	Yes	No	Yes
ST-NILM	LIT-SYN-1	**100.0**	88.6	**100.0**	**100.0**	**100.0**	94.3
	LIT-SYN-2	91.1	92.1	93.5	90.5	95.3	91.3
	LIT-SYN-3	**98.7**	88.7	94.3	94.0	**96.5**	91.4
	LIT-SYN-8	**96.5**	80.5	77.5	73.6	87.0	77.0
		No	Yes	No	Yes	No	Yes
DeepDFML [40]	LIT-SYN-1	100.0	**98.9**	100.0	99.6	100.0	**99.3**
	LIT-SYN-2	**97.8**	**97.9**	**98.3**	**98.7**	**98.2**	**98.3**
	LIT-SYN-3	92.6	**94.2**	**98.1**	**98.3**	95.4	**96.3**
	LIT-SYN-8	88.4	**89.0**	**94.4**	**90.2**	**91.4**	**89.6**

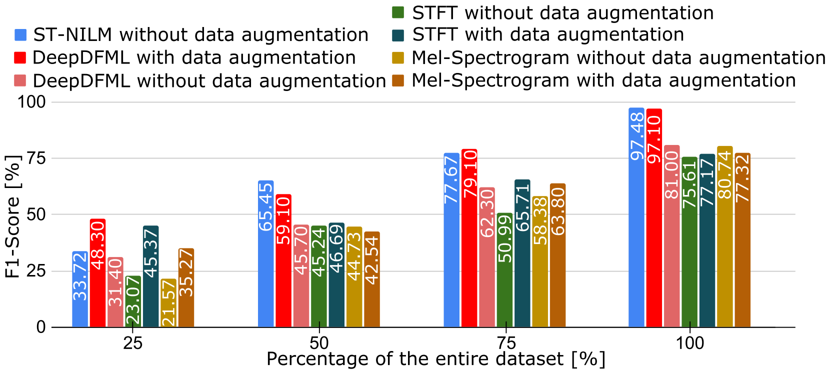


Fig. 4. Classification results: F1-Macro results for reduced training subset scenarios.

2) ST-NILM detection results are more affected using DA than DeepDFML; 3) ST-NILM without DA is better than DeepDFML without DA, i.e., PC_{av} is higher, for tripe loads (LIT-SYN-3); 4) Except for LIT-SYN-2, the metric PC_{on} obtained by ST-NILM without DA outperforms DeepDFML with DA for all other subsets; and 5) notwithstanding, ST-NILM without DA surpassed DeepDFLM and correctly detected all the events for LIT-SYN-1. For LIT-SYN-2 and LIT-SYN-3, the detection performances of ST-NILM and DeepDFLM were equivalent (maximum variation of 5.3%). For LIT-SYN-8 (the case with more aggregated loads), ST-NILM detected better turn-on events and, conversely, DeepDFML was better for turn-off events.

Table III presents a comparison between our work and the main state-of-the-art methods, taking into account the following criteria: 1) *the maximum number of multiple loads*, which is the maximum number of appliances added together in a single aggregated signal used for the proposed framework; 2) whether the method considers multilabel classification or not; 3) whether the method uses *DA* or not, which directly impacts the overall complexity of the proposal; 4) whether the method is *embedded* or not, which indicates the feasibility of real-time implementation; and 5) a classification or disaggregation metric. Table III shows that ST-NILM is the only method (among the analyzed methods) with an embedded application that performs multilabel disaggregation with FScore above 95%, without using DA.

D. Ablation Study on ST-NILM Parameters

To verify the impact of the variation in parameters m, J, and Q on the ST-NILM performance, we perform a series of training processes with different combinations of these parameters. We reproduced all the experiments previously presented with $m = 1$, now with $m = 2$, with Kymatio

TABLE III

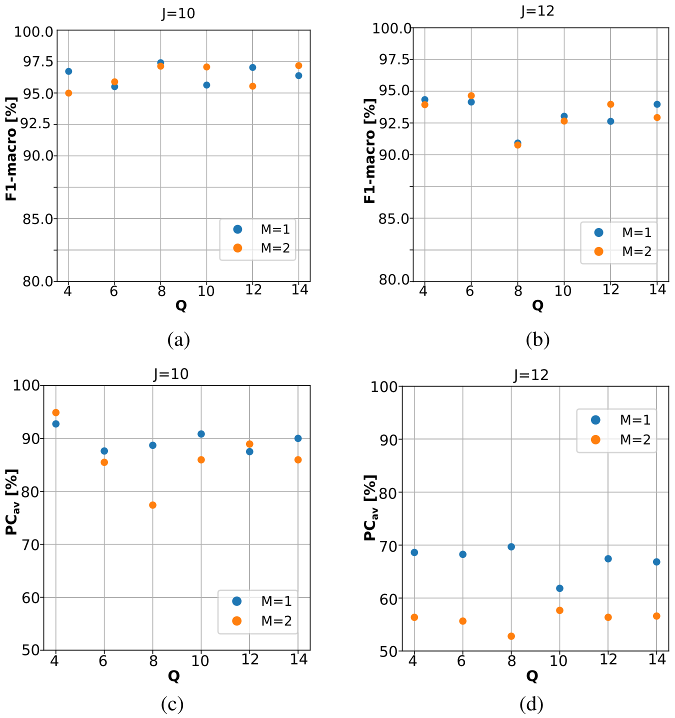
COMPARISONS WITH STATE-OF-THE-ART METHODS. THE ✓SYMBOL INDICATES THAT THE METHOD HAS THE RELATED CHARACTERISTIC AND THE ✗SYMBOL INDICATES THAT THE METHOD DOES NOT HAVE THAT CHARACTERISTIC

Method	Multiple loads	Multi-label?	Dataset	Augm.?	Embedded?	Best FS-core (%)
[14]	1	✗	PLAID	✗	✗	97%
[15]	5	✗	UK-DALE and BLUED	✗	✗	89%
[16]	-	✗	PLAID + WHITED	✗	✗	75%
[17]	3	✓	PLAID	✗	✗	94%
[18]	1	✓	PLAID	✗	✗	94%
[40]	8	✓	LIT-dataset	✓	✓	90%
[25]	3	✗	PLAID and WHITED	✗	✗	96%
[32]	1	✗	COOLL	✗	✗	99%
[7]	8	✗	LIT-dataset and PLAID	✗	✗	97%
[29]	5	✗	REDD	✓	✓	88%
[24]	8	✗	LIT-dataset	✗	✓	96%
This work	**8**	✓	**LIT-dataset**	✗	✓	**97%**

library [48]. For $m = 2$, for each time series, we extract the features considering the following.

1) For the classification model, we use the averages of each first-order grid as features. For the detection model, we apply a flatten layer, as in Fig. 1, and concatenate the difference between the averages of each first-order grid and second-order grid, in all subbands.
2) We apply the same fully connected layers used for $m = 1$ in the tests for $m = 2$.
3) For $m = 2$, we assume one wavelet per octave in the second-order filter bank and Q filters per octave in the first-order filter bank. For $m = 1$, we assume Q wavelets per octave in the first-order filter bank.
4) We used the same classification metrics for $m = 1$ and $m = 2$.

Fig. 5 presents the classification and detection results obtained by tests with parametric variation. Comparing the classification results in Fig. 5(a) and (b), we note that the F1-Macro is not significantly impacted when choosing $m - 2$ instead of $m = 1$ (maximum variation less than 1.8%). We also note that the average classification performance is better for $J = 10$ than for $J = 12$ (3.2% higher F1-Macro average value) and that parameter Q has more impact on classification performance for $J = 12$ than for $J = 10$. The detection results presented in Fig. 5(c) and (d) show that the J parameter is the most impactful for the detection since the average PC_{av} drops more than 25% when exchanging $J = 10$ for $J = 12$. Furthermore, the variance of PC_{av} with different Q is greater with $J = 12$ than with $J = 10$. We noted that the metrics PC_{av} are better for $J = 10$ in most analyzed scenarios. Finally, the analysis of the results leads us to conclude that: 1) increasing the value of J does not necessarily improve the classification performance and aggravates the detection performance and 2) using $m = 2$ instead of $m = 1$ does not result in significant improvement in classification performance and deteriorates detection performance.



(a)

(b)

(c)

(d)

Fig. 5. (a) Classification and (b) detection results considering different ST parameters. Note that classification results do not variate significantly by increasing J. Detection metric PC$_{av}$ deterioration, on the other hand, is much more evident when exchanging $J = 10$ by $J = 12$. (a) F1-Macro for $J = 10$ with different Q values. (b) F1-Macro for $J = 12$ with different Q values. (c) PC$_{av}$ for $J = 10$ with different Q values. (d) PC$_{av}$ for $J = 12$ with different Q values.

E. Embedded System

With the objective of evaluating the practical implementation of the proposed method, a subject also underexplored in the literature (Table III), ST-NILM was tested on a Jetson NVIDIA TX1 to evaluate its performance in an embedded system environment. This platform has a 256-core GPU, a four-core ARM Cortex-A57 CPU, and 4 GB of RAM under a Linux operating system. The inference performance, presented in Table IV, was obtained from an average of 841 examples. Compared with DeepDFML, ST-NILM had lower RAM usage (62.7% vs 94.8%), lower GPU load, and substantially fewer floating point operations (0.264 GFLOPS for ST-NILM and 7.643 GFLOPS for DeepDFLM).

TABLE IV
RESOURCES CONSUMPTION ON THE EMBEDDED SYSTEM

Resource monitored	System state	Value
RAM	Idle	808/3984 Mb (20.3%)
	ST-NILM	**3379/3984 Mb (84.8%)**
	[40]	3777/3984 Mb (94.8%)
Avg. CPU load	Idle	8.2%
	ST-NILM	44.0%
	[40]	30.9%
Avg. GPU load	Idle	0.0%
	ST-NILM	**38.0%**
	[40]	50.6%
FLOPS	[40]	7.643 G
	ST-NILM	**0.264 G**

VI. CONCLUSION

In this article, we presented a DNN architecture to classify electrical signals in a unified way. The proposed network, ST-NILM is based on the DeepDFML architecture, which has a shared CNN with many learned parameters. However, ST-NILM has an untrained shared convolutional network based on ST that has analytically computed filters given by wavelets.

We tested ST-NILM with the LIT-SYN dataset and repeated the same tests with the DeepDFML architecture for comparison purposes. Classification results with smaller datasets were substantially better for ST-NILM than DeepDFML. Furthermore, classification results show that ST-NILM, even without DA, outperforms DeepDFML with DA. The good classification performance of ST-NILM, unlike DeepDFML, is not dependent on DA. The improvements in classification performance over DeepDFML can be explained by the fact that there is no need for the scattering network to learn the filter coefficients, only the output networks. The lack of training in the convolutional network contributed to the ST-NILM obtaining better results than DeepDFML for smaller training sets. ST-NILM achieved detection results equivalent to DeepDFML for LIT-SYN2 and LIT-SYN-3, and performed better for single loads (LIT-SYN-1). The resources used by the hardware implementation showed that ST-NILM consumes less memory, less GPU load, and consumes substantially less computational effort. The results obtained for the embedded system with LIT-SYN (a real-world dataset) indicate that future tests with ST-NILM and real-time data can be promising.

As future works, we intend to do the below.

1) Improve the GPU implementation of ST-NILM.
2) Improve the feature selection techniques, to improve PC_{off} in the case of more aggregate loads.
3) Improve the classification and detection comparisons with other time–frequency-based feature extractors.
4) Evaluate the ST-NILM with other real-world datasets.
5) Evaluate the ST-NILM with real-time data.

REFERENCES

[1] G. W. Hart, "Nonintrusive appliance load monitoring," *Proc. IEEE*, vol. 80, no. 12, pp. 1870–1891, Jun. 1992.

[2] C. Laughman et al., "Power signature analysis," *IEEE Power Energy Mag.*, vol. 1, no. 2, pp. 56–63, Mar. 2003.

[3] M. S. Ibrahim, W. Dong, and Q. Yang, "Machine learning driven smart electric power systems: Current trends and new perspectives," *Appl. Energy*, vol. 272, Aug. 2020, Art. no. 115237, doi: 10.1016/j.apenergy.2020.115237.

[4] N. Sadeghianpourhamami, J. Ruyssinck, D. Deschrijver, T. Dhaene, and C. Develder, "Comprehensive feature selection for appliance classification in NILM," *Energy Buildings*, vol. 151, pp. 98–106, Sep. 2017.

[5] J. Kelly and W. Knottenbelt, "Neural NILM: Deep neural networks applied to energy disaggregation," in *Proc. 2nd ACM Int. Conf. Embedded Syst. Energy Efficient Built Environ.*, 2015, pp. 55–64.

[6] S. Mallat, "Group invariant scattering," *Commun. Pure Appl. Math.*, vol. 65, no. 10, pp. 1331–1398, Oct. 2012.

[7] E. de Aguiar, A. Lazzaretti, B. Mulinari, and D. Pipa, "Scattering transform for classification in non-intrusive load monitoring," *Energies*, vol. 14, no. 20, p. 6796, Oct. 2021.

[8] K. Chen, Q. Wang, Z. He, K. Chen, J. Hu, and J. He, "Convolutional sequence to sequence non-intrusive load monitoring," 2018, *arXiv:1806.02078*.

[9] H. Chen, Y.-H. Wang, and C.-H. Fan, "A convolutional autoencoder-based approach with batch normalization for energy disaggregation," *J. Supercomput.*, vol. 77, no. 3, pp. 2961–2978, Mar. 2021.

[10] A. Moradzadeh, B. Mohammadi-Ivatloo, M. Abapour, A. Anvari-Moghaddam, S. G. Farkoush, and S.-B. Rhee, "A practical solution based on convolutional neural network for non-intrusive load monitoring," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 10, pp. 9775–9789, Oct. 2021.

[11] K. Chen, Y. Zhang, Q. Wang, J. Hu, H. Fan, and J. He, "Scale- and context-aware convolutional non-intrusive load monitoring," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 2362–2373, May 2020.

[12] L. Massidda, M. Marrocu, and S. Manca, "Non-intrusive load disaggregation by convolutional neural network and multilabel classification," *Appl. Sci.*, vol. 10, no. 4, p. 1454, Feb. 2020.

[13] M. Kaselimi, E. Protopapadakis, A. Voulodimos, N. Doulamis, and A. Doulamis, "Multi-channel recurrent convolutional neural networks for energy disaggregation," *IEEE Access*, vol. 7, pp. 81047–81056, 2019.

[14] S. Houidi, D. Fourer, and F. Auger, "On the use of concentrated time-frequency representations as input to a deep convolutional neural network: Application to non intrusive load monitoring," *Entropy*, vol. 22, no. 9, p. 911, Aug. 2020.

[15] Q. Wu and F. Wang, "Concatenate convolutional neural networks for non-intrusive load monitoring across complex background," *Energies*, vol. 12, no. 8, p. 1572, Apr. 2019.

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

[17] A. Faustine and L. Pereira, "Multi-label learning for appliance recognition in NILM using fryze-current decomposition and convolutional neural network," *Energies*, vol. 13, no. 16, p. 4154, Aug. 2020.

[18] 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, p. 3374, Jul. 2020.

[19] Y. Himeur, A. Alsalemi, F. Bensaali, and A. Amira, "An intelligent nonintrusive load monitoring scheme based on 2D phase encoding of power signals," *Int. J. Intell. Syst.*, vol. 36, no. 1, pp. 72–93, Jan. 2021.

[20] D. Yang, X. Gao, L. Kong, Y. Pang, and B. Zhou, "An event-driven convolutional neural architecture for non-intrusive load monitoring of residential appliance," *IEEE Trans. Consum. Electron.*, vol. 66, no. 2, pp. 173–182, May 2020.

[21] A. Morán, S. Alonso, D. Pérez, M. A. Prada, J. J. Fuertes, and M. Domínguez, "Feature extraction from building submetering networks using deep learning," *Sensors*, vol. 20, no. 13, p. 3665, Jun. 2020.

[22] A. E. Lazzaretti et al., "A multi-agent NILM architecture for event detection and load classification," *Energies*, vol. 13, no. 17, p. 4396, Aug. 2020.

[23] A. Mukaroh, T.-T.-H. Le, and H. Kim, "Background load denoising across complex load based on generative adversarial network to enhance load identification," *Sensors*, vol. 20, no. 19, p. 5674, Oct. 2020.

[24] B. M. Mulinari, L. d. S. Nolasco, E. Oroski, A. E. Lazzaretti, R. R. Linhares, and D. P. B. Renaux, "Feature extraction of V-I trajectory using 2-D Fourier series for electrical load classification," *IEEE Sensors J.*, vol. 22, no. 18, pp. 17988–17996, Sep. 2022.

[25] J. Chen, X. Wang, X. Zhang, and W. Zhang, "Temporal and spectral feature learning with two-stream convolutional neural networks for appliance recognition in NILM," *IEEE Trans. Smart Grid*, vol. 13, no. 1, pp. 762–772, Jan. 2022.

[26] C. Athanasiadis, D. Doukas, T. Papadopoulos, and A. Chrysopoulos, "A scalable real-time non-intrusive load monitoring system for the estimation of household appliance power consumption," *Energies*, vol. 14, no. 3, p. 767, Feb. 2021.

[27] E. Gomes and L. Pereira, "PB-NILM: Pinball guided deep non-intrusive load monitoring," *IEEE Access*, vol. 8, pp. 48386–48398, 2020.

[28] Z. Jia, L. Yang, Z. Zhang, H. Liu, and F. Kong, "Sequence to point learning based on bidirectional dilated residual network for non-intrusive load monitoring," *Int. J. Electr. Power Energy Syst.*, vol. 129, Jul. 2021, Art. no. 106837.

[29] H. Hwang and S. Kang, "Nonintrusive load monitoring using an LSTM with feedback structure," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2022.

[30] I. Laouali, A. Ruano, M. D. G. Ruano, S. D. Bennani, and H. E. Fadili, "Non-intrusive load monitoring of household devices using a hybrid deep learning model through convex hull-based data selection," *Energies*, vol. 15, no. 3, p. 1215, Feb. 2022.

[31] J. Bruna and S. Mallat, "Invariant scattering convolution networks," 2012, *arXiv:1203.1513*.

[32] E. L. Aguiar, A. E. Lazzaretti, and D. R. Pipa, "Performance of scattering transform feature extraction for electrical load classification," in *Proc. 15th Congr. Computatioal Intell.*, Joinville, Brazil, Jan. 2021, pp. 1–8.

[33] E. Sejdic, I. Djurovic, and J. Jiang, "Time-frequency feature representation using energy concentration: An overview of recent advances," *Digit. Signal Process.*, vol. 19, no. 1, pp. 153–183, Jan. 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S105120040800002X

[34] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, no. 4, pp. 357–366, Aug. 1980.

[35] I. Daubechies, J. Lu, and H.-T. Wu, "Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 243–261, Mar. 2011, doi: 10.1016/j.acha.2010.08.002.

[36] T. Oberlin, S. Meignen, and V. Perrier, "The Fourier-based synchrosqueezing transform," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Jul. 2014, pp. 315–319.

[37] G. Michau, G. Frusque, and O. Fink, "Fully learnable deep wavelet transform for unsupervised monitoring of high-frequency time series," *Proc. Nat. Acad. Sci. USA*, vol. 119, no. 8, pp. 1–10, Feb. 2022.

[38] C. Wang, E. Benetos, V. Lostanlen, and E. Chew, "Adaptive scattering transforms for playing technique recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 30, pp. 1407–1421, 2022.

[39] S. Souli, R. Amami, and S. B. Yahia, "A robust pathological voices recognition system based on DCNN and scattering transform," *Appl. Acoust.*, vol. 177, Jun. 2021, Art. no. 107854.

[40] L. D. S. Nolasco, A. E. Lazzaretti, and B. M. Mulinari, "DeepDFML-NILM: A new CNN-based architecture for detection, feature extraction and multi-label classification in NILM signals," *IEEE Sensors J.*, vol. 22, no. 1, pp. 501–509, Jan. 2022.

[41] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6517–6525.

[42] F. Guth, J. Zarka, and S. Mallat, "Phase collapse in neural networks," 2021, *arXiv:2110.05283*.

[43] A. Ramírez-Ramírez, V. F. Jiménez-Reyes, J. A. Vélez-Enríquez, S. Leal-Ortiz, and J. García-Guzmán, "Study of harmonic content and influence of common electrical appliances used in residential installations," in *Proc. IEEE Int. Conf. Eng. Veracruz (ICEV)*, Oct. 2019, pp. 1–8.

[44] J. Bruna and S. Mallat, "Classification with scattering operators," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1561–1566.

[45] E. L. Aguiar, A. E. Lazzaretti, and D. R. Pipa, "Features extraction and selection with the scattering transform for electrical load classification," *Learn. Nonlinear Models*, vol. 21, no. 1, pp. 19–35, Feb. 2023.

[46] D. P. B. Renaux et al., "A dataset for non-intrusive load monitoring: Design and implementation," *Energies*, vol. 13, no. 20, p. 5371, Oct. 2020.

[47] M. N. Meziane, P. Ravier, G. Lamarque, J.-C. Le Bunetel, and Y. Raingeaud, "High accuracy event detection for non-intrusive load monitoring," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 2452–2456.

[48] M. Andreux et al., "Kymatio: Scattering transforms in Python," 2018, *arXiv:1812.11214*.

Everton Luiz de Aguiar received the B.Sc., M.Sc., and D.Sc. degrees in electrical engineering from the Federal University of Technology–Paraná, Curitiba, Brazil, in 2012, 2014, and 2023, respectively.

He is currently a Professor with the Federal University of Technology–Paraná. His research interests include deep learning, power electronics, and digital signal processing.

Lucas da Silva Nolasco received the B.Sc. and M.Sc. degrees in computer engineering, in 2022 and 2023, respectively. His research interests include embedded systems and digital signal processing.

André Eugenio Lazzaretti (Member, IEEE) received the B.Sc., M.Sc., and D.Sc. degrees in electrical engineering from the Federal University of Technology–Paraná, Curitiba, Brazil, in 2007, 2010, and 2015, respectively.

He is currently a Professor with the Department of Electronics, Federal University of Technology–Paraná. His research interests include machine learning, deep learning, and digital signal processing.

Daniel Rodrigues Pipa (Member, IEEE) received the B.Sc. degree in electrical engineering from the Universidade Tecnológica Federal do Paraná (UTFPR), Curitiba, Brazil, in 2004, and the M.Sc. and D.Sc. degrees in electrical engineering from the Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil, in 2009 and 2012, respectively.

From 2005 to 2012, he was with the Research and Development Center (CENPES), Petrobras, Rio de Janeiro, in the fields of signal, image, and video processing applied to equipment integrity. In 2011, he was a Visiting Researcher with the University of California at San Diego, La Jolla, CA, USA. He has been an Associate Professor of Electrical Engineering with UTFPR since 2012. His current research interests include statistical signal processing, signal and image reconstructions, inverse problems, compressive sensing, and related areas.

Heitor Silvério Lopes received the B.Sc. and M.Sc. degrees in electrical engineering from the Federal University of Technology–Paraná (UTFPR), Curitiba, Brazil, in 1984 and 1990, respectively, and the Ph.D. degree from the Federal University of Santa Catarina, Florianopolis, Brazil, in 1996.

In 2014, he spent a sabbatical year at the Department of Electrical Engineering and Computer Science of the University of Tennese, Knoxville, TN, USA. Since 2003, he has been a Research Fellow in the area of computer science with the Brazilian National Research Council, Curitiba. Currently, he is a tenured Full Professor with the Department of Electronics and the Graduate Program on Electrical Engineering and Applied Computer Science (CPGEI), UTFPR. He is the Founder and the current Head of the Bioinformatics and Computational Intelligence Laboratory (LABIC), Curitiba. His major research interests include computer vision, deep learning, evolutionary computation, and data mining.