Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 21, Iss. 1, pp. 19-35, 2023

FEATURES EXTRACTION AND SELECTION WITH THE SCATTERING TRANSFORM FOR ELECTRICAL LOAD CLASSIFICATION

Everton Luiz de Aguiar, André Eugenio Lazzaretti, and Daniel Rodrigues Pipa
Federal University of Technology (UTFPR)
{eaguiar,lazzaretti,pipa}@utfpr.edu.br

Abstract –The Scattering Transform (ST) presents itself as an alternative approach to the classic methods that involve neural networks and deep learning techniques for the feature extraction and classification of electrical signals. Among its main advantages, one can emphasize that the coefficients of the ST are determined analytically and do not need to be learned, as typically performed in Convolutional Neural Networks (CNNs). Additionally, ST has time-shifting and small time-warping invariance, which reduces the need for precise temporal localization (detection) for subsequent classification. This paper originally proposes six feature extraction and selection methods applied to classification of Non-intrusive Load Monitoring (NILM) high-frequency signals. We visually analyze the separability among classes for the proposed Feature Extractors and validate the performance of the proposed methods varying several parameters for ST calculation, such as signal length, number of examples, and sampling frequency. The results outperform other state-of-the-art feature extraction techniques, reaching up to 100% of FScore for a publicly available dataset, demonstrating the feasibility and promising aspects of the ST for NILM problems.
Keywords – NILM, Feature Extraction, Scattering Transform.

1. INTRODUCTION

Non-intrusive load monitoring, initially proposed in [1], comprises a set of computational signal processing solutions applied to residential electrical appliances signals. Typically, it is desired to recognize (non-intrusively) the nature of each electrical load in a home (classification task) or to identify the respective energy consumption of each equipment from the aggregated signal (disaggregation task).

The two main tasks of NILM are classification and disaggregation. The classification identifies the electric appliance or the set of electric appliances activated in a given instant of time, and the disaggregation is the separation of the individual electrical signals that composes an aggregate signal. The steps for the NILM classification are (i) event detection, (ii) feature extraction, (iii) training of the classification model, and (iv) prediction [2].

Regarding feature extraction, one of the most relevant stages is identifying the power signature (PS). PS is a particular representation that characterizes the behavior of each appliance. One can obtain the PS through the features extracted from the electrical time signal (voltage, current, or energy). Based on the idea proposed in [3], one can categorize the feature extractors into (i) Conventional Physical Definitions (CPD), (ii) Time-Frequency Analysis (TFA), and (iii) Voltage-Current (VI) Trajectories. The CPD-based feature extractors are simpler to implement but have poor discriminability and less capability of extracting transient information [1]. TFA feature extractors are more discriminant than CPD since they generally deal with high-frequency data but are time-shifting covariant, having a dependency on event detection [4], but generally they are dependent on the event detection techniques [3].

In the last five years, the scientific community has presented many feature extractors based on Deep Convolutional Networks (DCNNs), which we number as a fourth category of feature extractors (iv), in addition to CPD, TFA and VI Trajectories. DCNN methods are highly discriminative and overcame state-of-the-art performances for NILM in the last five years, both in terms of classification and disaggregation [5, 6]. However, the performance of the DCNN methods depends on the network architecture, and mostly, on the amount of data available for training. This means that better results are achieved at the expense of more time to learn the features of the signal and a larger database. These dependencies add to the overall complexity of the classification task.

The Scattering Transform (ST), first proposed in [7], is a convolutional network composed of multi-scale filter banks that implement cascaded wavelets transforms. Such filters have analytically determined coefficients, i.e., they do not require the training process for convolutional layers. Each layer of the convolutional network gives rise to coefficients that can be used as features of the electrical signal for classification. This allows the feature extraction to deal better than CNN with smaller datasets (with trained coefficients) and simplify the classification task. In addition, the overall complexity is reduced. To the best of our knowledge, the only works that use ST instead of CNN for NILM classification are [8] and [9]. In [8], the authors proposed a feature selection strategy based on the energy of the first-order coefficients of ST, applied to several scenarios. Each scenario took different time regions from the PLAID [10] and LIT-SYN [11] datasets. Authors reached state-of-the-art classification metrics (accuracy above 99% for all evaluated scenarios). In [9], the authors proposed a framework for extracting features from NILM signals based on ST. The proposed framework contemplated classification task experiments with parametric variations, such as signal length, number of examples by class, and sampling frequency.

Despite the superior overall results, two research gaps result from the work of [8]. First, the authors did not analyze the impacts of other feature selection strategies on classification performance. Secondly, they did not perform classification experiments on isolated loads but only on aggregate loads. In this sense, this paper is an extension of [9]. Using the framework proposed in [9] as a reference, we propose and analyze in this work six different methods for selecting features using the Scattering Transform, filling the first research gap from [8]. In addition, we discuss and show the discriminability of the proposed method among similar classes using isolated loads from a publicly available dataset, filling the second research gap in [8]. Hence, the main contributions of this paper are:

- A new ST-based framework for NILM signals classification;
- Experimental classification performance analysis under non-ideal dataset conditions, such as fewer examples per class and lower sampling frequency;
- Proposition and performance analysis of six ST-based analytical features selection techniques;
- A visual (low-dimensional) analysis of the discriminability of the classes applying ST-based features extractor.

We propose a framework for NILM based on the Scattering Transform. We show that this feature extraction and selection allows the classification results not to be substantially altered with the decrease in the number of training examples, the modification of the sampling frequency, or the absence of single load connection event detection. We propose a testing framework to verify these properties and compare the evaluated classification results with baseline (state-of-the-art) methods.

This paper is divided as follows. Section 2 presents the Related Works. Section 3 presents the Proposed Framework, and its subsections present each building block of the Proposed Framework. Section 4 presents the Experimental Analysis under several scenarios and cases. Section 5 shows the Comparisons With State-Of-The-Art Approaches, and the Conclusions and Future Works are discussed in Section 6.

2. RELATED WORKS

The first NILM Feature Extractors were based on the electrical characteristics of the signals. In [12] and [1], the authors used the average powers obtained directly from the voltage and current curves. The authors decomposed the current harmonic coefficients and used them as features in [13]. Similarly, in [14], the authors used the Total Harmonic Distortion of Current (THDi) to define the features map. In [15], the admittance was used as a parameter to define the NILM Feature Extractor. Despite being relatively easy to implement, methods based on electrical characteristics are not discriminative enough for loads with many transients (switching circuits), or without a cyclic pattern (a washing machine has a cyclical pattern of operation, for example).

Time-frequency-based Feature Extractors use time-frequency transforms to define the features space. Well known time-frequency methods are: Wavelet Transform [4], Short-Time Fourier Transform [16], and Scattering Transform [7]. One can find Feature Extractors for NILM applying Wavelet Transform coefficients in [17–19]. The Scattering Transform, proposed in [7], dealt with the time-shifting variance of the Wavelet Transform, improving pattern recognition tasks results by reducing variability [20]. The authors in [9] proposed a framework to classify electrical loads signals from COOLL Dataset [21] and obtained state-of-the-art results for that dataset. In [8], authors expanded the analysis of [9] to multiple aggregated loads, using PLAID and LIT-SYN datasets with Scattering Transform-based Feature Extractors. However, as previously discussed, the authors did not address in [8] the impact of the Feature Selection Method choice, and the classification results were restricted to one only strategy of feature selection. Besides that, Aguiar et al. [8] did not present results with single loads but only with aggregated loads.

Several other researchers used the VI trajectories to determine extractors of NILM features. These strategies map electrical signals onto a Cartesian plane of voltage and current. From this mapping, a 2D image is obtained, and from this image, the features for NILM are extracted. In [22], the authors used area, asymmetry, and looping detection to determine features. In [23], the slope of the middle segment, the curvature of the mean line, and self-intersection are used. In [24], on the other hand, a weighted pixelated image for each electrical signal is created, and these images served as input of a Convolucional Neural Network (CNN), which performed the classification. Finally, in [25], the authors proposed new features from VI trajectories taken from both transient and steady-state periods. VI trajectories feature extraction methods have two main limitations: the dependency of high-frequency data and event detection algorithms.

The relationship between NILM and Deep Learning started with the work of [26]. In that paper, the authors proposed three deep networks for NILM, applied to a low-frequency dataset. Each proposed architecture had a dedicated output network to classify each load[1]. The authors proposed: (i) an architecture based on recurrent networks (bidirectional LSTM + CNN); (ii) Denoising Autoencoder; (iii) Regress Start Time, End Time, and Power. The three proposed networks extract NILM features and also perform disaggregation. Padding and sliding windows are used in strategies (i) and (ii). On the other hand, strategy (iii) used a probabilistic output for each appliance's power demand, converting this to a single vector per appliance.

Since [26], many other papers proposed deep Convolutional Network architectures for NILM tasks. Gomes and Pereira [6] used a pinball loss function (PB) in different Deep Networks architectures and compared the results with the mean squared error

[1]We refer to *low-frequency dataset* when data has sampling frequency less or equal to 3Hz, and *high-frequency dataset* otherwise.

(MSE) loss function. The input signal was fed in a structure containing two summed systems composed of a 1D convolution followed by a Gated Recurrent Unit (GRU) and an activation function. Authors performed experiments with manually summed data (sum of appliances) and aggregated data (actually measured at energy input). Both with pinball loss function and MSE, all results were better with the sum of loads. The authors, however, reported difficulties with time-shifting loads.

Authors in [27] applied bidirectional dilated convolution networks to extract features from NILM signals. This choice increases the length of receptive fields (receptive field grows exponentially with depth) and improves the prediction result. The features come from residual blocks, each one containing a non-causal dilated convolution with different filter lengths and dilatation factors. The results of [27] outperformed existing methods at the date. However, the authors did not discuss the computational cost of the training stage and the non-causal characteristic of the method, hampering its use in real-time applications.

Authors proposed in [28] a deep CNN Feature Extractor integrated with a classification task. Moradzadeh et al. [28] proposed to classify appliances of households not included in the training stage and achieved accuracy results above 96%, but limited to the low-frequency REDD dataset, proposed by [29]. Himeur et al. [30] proposed a 2D phase encoding of power signal named 2D-PEP to extract features and classify NILM signals. Using time-domain Feature Extractors, the authors overcame state-of-the-art classification metrics for several distinct datasets based on sliding windows. Despite these promising results, authors in [30] had a high dependency on the event detection algorithm.

Nolasco et al. [5] proposed the DeepDFML architecture that integrated disaggregation, event detection, and multi-label classification. The DeepDFML had a shared DCNN stage, with three fully connected sub-networks, each one for one specific task. The authors proposed new metrics, as the work was pioneering in solving all three tasks at once. The authors needed external methods to generate more training data (data augmentation) despite the promising results – state-of-the-art for LIT dataset, proposed in [11].

Chen et al. [31] proposed two signatures for each appliance: one temporal and other spectral. The authors converted disaggregated 1D signals of voltage and current into images and defined those images as the signature of each load. Both temporal and Spectral signatures, 2D images, were inputs of two CNN structures, followed by a shared fully connected layer. The FScore validation results overcame VI and weighted recurrence graph methods, but the authors did not compare their proposal with other state-of-the-art CNN-based methods. Furthermore, the authors depended on the single load switching point detection (time-shifting covariant).

The authors proposed in [32] a feature selection method at low frequency (1Hz) based on 38 features of power theory. With a reduced set of features, the authors achieved up to 98% accuracy for the PLAID dataset. However, the authors in [32] did not discuss the effect of frequency variation on classification performance using the proposed features. In the work [33], authors proposed Neural Fourier Energy Disaggregator (NFFD). NFFD is based on the Fourier transform, a representation without a location in time. Although the results found by the authors are compatible with other state-of-the-art methods, the authors do not discuss the impact of the lack of temporal localization in detecting events. In [34], in the other hand, the authors proposed an event detection method for NILM signals based on harmonic content, which uses various frequency scales. This detection method is applicable for high-frequency data, and the false event detection results have shown promising results.

Authors in [35] proposed a low-frequency time-shifting invariant feature extractor that allows transfer learning to NILM. Although there was a 13% increase in performance compared to the best method in the literature, the authors do not assess the impact of predicting unknown loads on the metrics. One can find a comprehensive literature review of low-frequency NILM methods in [36] and of NILM applications with deep learning and neural networks in [37].

Bearing in mind the limitations of the methods in this section, we propose a new method based on the Scattering Transform to extract features from NILM signals that: (i) is more discriminatory than the methods based on electrical characteristics, (ii) it is less dependent of load switching detection (time-shifting invariant), (iii) it is a Convolutional Network with *untrained* coefficients and (iv) it is a Convolutional Network with 1D input, unlike CNN-based methods with 2D input images.

3. PROPOSED FRAMEWORK

Fig. 1 shows the proposed classification framework for NILM using Scattering Transform. The following steps constitute the proposed framework: (i) Pre-processing, (ii) Feature Extraction, (iii) Classification, and (iv) Evaluation of Results. In the following subsections, we initially detail the dataset used, and later we will discuss each of the steps of the proposed framework (Fig. 1).

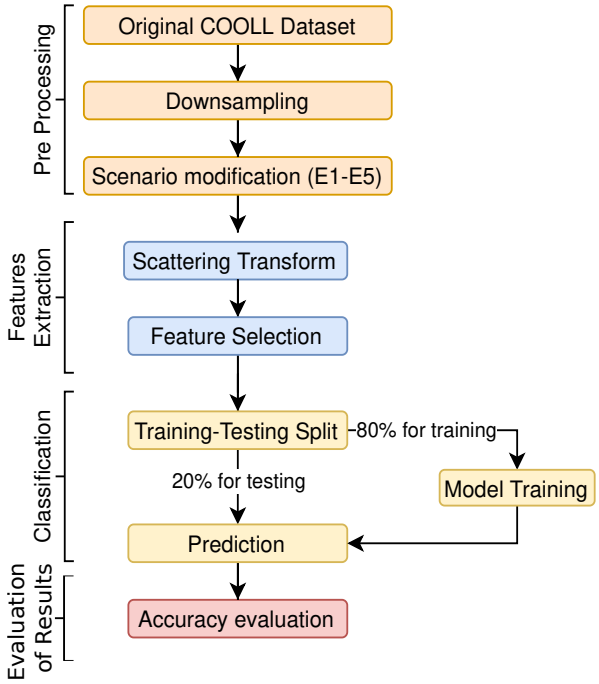


Figure 1: Proposed framework for NILM Classification.

3.1 COOLL Dataset

Considering that the features extracted from high-frequency data are more discriminative than low-frequency data [38], we chose the Controlled On/Off Loads Library dataset (COOLL), proposed in [21]. The COOLL dataset has submetered instantaneous current records from 42 different electrical appliances at a sampling frequency of 100kHz. Each electric load has 20 samples of 6s duration, at 100kHz, totaling 840 samples, as presented in Tab. 1.

Table 1: COOLL Dataset.

Class Number	Appliance	Category	Class Number	Appliance	Category
1	Drill 1	Drills	22	Paint Stripper 1	Paint Strippers
2	Drill 2		23	Planer 1	Planers
3	Drill 3		24	Router 1	Routers
4	Drill 4		25	Sander 1	Sanders
5	Drill 5		26	Sander 2	
6	Drill 6		27	Sander 3	
7	Fan 1	Fans	28	Saw 1	Saws
8	Fan 2		29	Saw 2	
9	Grinder 1	Grinders	30	Saw 3	
10	Grinder 2		31	Saw 4	
11	Hair Dryer 1	Hair Dryers	32	Saw 5	
12	Hair Dryer 2		33	Saw 6	
13	Hair Dryer 3		34	Saw 7	
14	Hair Dryer 4		35	Saw 8	
15	Hedge Trimmer 1	Hedge Trimmers	36	Vacuum Cleaner 1	Vacuum Cleaners
16	Hedge Trimmer 2		37	Vacuum Cleaner 2	
17	Hedge Trimmer 3		38	Vacuum Cleaner 3	
18	Lamp 1	Lamps	39	Vacuum Cleaner 4	
19	Lamp 2		40	Vacuum Cleaner 5	
20	Lamp 3		41	Vacuum Cleaner 6	
21	Lamp 4		42	Vacuum Cleaner 7	

3.2 Preprocessing

The filter banks that implement the ST are multi-scale operators, and their implementation uses downsampling by two. For each downsampling by two, one of every two samples in the time domain is discarded, and the sampling frequency drops by half. For this reason, we changed the sampling frequency of the original input signal, coming from the COOLL dataset in the downsampling stage. The signal resulting from this step has a sampling frequency that is a power of 2 (8 192Hz), which makes the filter banks of the Scattering Transform feasible. After downsampling, we changed the dataset based on five different Scenarios:

- **Scenario 1 (SC1):** Different number of cycles for the signal window, *with* turn-on events detection;

- **Scenario 2 (SC2):** Different number of cycles for the signal window, *without* turn-on events detection;
- **Scenario 3 (SC3):** Different number of examples per class;
- **Scenario 4 (SC4):** Different sampling frequency;
- **Scenario 5 (SC5):** Whole signal-length, at 8 192Hz of sampling frequency.

3.3 Feature Extraction

We apply the Scattering Transform to extract features for NILM signals. Therefore, we explain the mathematical definitions needed to clarify the Scattering Transform in this section.

Consider a set of features determined by the representations of two signals f and g. Let $\Phi(f)$ and $\Phi(g)$ representations of the signals f and g signals, respectively. Then, the Euclidean distance $d(f,g)$, defined by $d(f,g) = ||\Phi(f) - \Phi(g)||$ must be small for elements of the same class and large for elements of different classes [20]. The similarity measure between $\Phi(f)$ and $\Phi(g)$ depends on the inner product of the two representations. The central question of the classification is to define a good kernel[2], which allows for a reliable measurement of similarity [7].

Consider a discrete time signal $x[n]$, a translated signal $x_c[n] = x[n-c]$ and a deformed (time-warped) signal $x_d[n] = x[n-\tau[n]]$ from the same type of electrical appliance. There may be both translation and deformation of electrical signals of the same class in real cases. This occurs, for example, when the same load is switched on at different times in the same sampling window (translation) or when there is measurement noise (time-warping). The classifier should be invariant to translation and also to the small time-warping.

Let a Wavelet $\Psi_\lambda[n]$ be defined by $\Psi_\lambda[n] = \lambda\Psi[\lambda n t_s]$, where $\Psi[n]$ is the discrete time mother Wavelet, $\lambda = 2^{-jQ}$, Q is the number of Wavelets per octave, t_s is the sampling period, n is an integer that represent the n-th sample, and j is the scale factor. So the Wavelet transform of $x[n]$ is:

$$Wx[n] = \{x[n] * \Phi[n], x[n] * \Psi_\lambda[n]\}_\lambda. \tag{1}$$

The Wavelet Transform has the following advantages: (i) it is stable for small-time deformations; (ii) it is well located both in time and in frequency, but it has the disadvantage of being a time-shifting variant. This happens because the Wavelet transform is calculated using convolutions [4]. To solve that problem, Mallat et al. [7] used the coefficients module, followed by the average in time:

$$\{|x[n] * \Psi_\lambda| * \Phi[n]\}_\lambda, \tag{2}$$

being $\Phi[n]$ a low-pass filter that implements the average. The modulus and average operators guarantee the time-shifting invariance, but results in loss of information [7, 20].

Mallat et al. [7] proposed applying successive modulus and average operations to new layers of convolutions with Wavelets starting from $\{|x * \Psi_\lambda| * \Phi(t)\}_\lambda$. This gives rise to the Scattering Path, in discrete time, of sequence $x[n]$, given by:

$$S_m[n, \lambda_1, \lambda_2, \ldots, \lambda_m] = |[|(|x[n] * \Psi_{\lambda_1}|) * \Psi_{\lambda_2}|] \cdots * \Psi_{\lambda_m}| * \Phi[n], \tag{3}$$

depending on the order m and the frequency scales $\lambda_1, \ldots, \lambda_m$.

The total number of coefficients for the Scattering Transform of a discrete signal x is $Q^m log_m N_s$, being N_s the total number of samples of x.

We consider for the experiments $m = 2$, implemented by two filter banks. The first filter bank, corresponding to the first layer, has $Q = 8$. The second filter bank has $Q = 1$. The choice of the ST structure is based on [39].

Fig. 2 shows the structure of the ST graphically. The blue arrows in Fig. 2 represent the convolution operation with the low-pass filter Φ_T. The set of all coefficients obtained as a result of the average operation (the tip of the arrows in Fig 2) composes the Scattering Transform. The sequence x[n], which represents an electrical quantity sampled at discrete instants indexed by the integer n, passes through the first layer of convolutions with the Wavelets $\Psi_{q,i} = \Psi_{1,i}$. The subscript i is the frequency scale of the first filter bank, and q is the index for the layer of the Scattering Transform. The convolution module is taken at each tree node in the first layer. The output of each node in the first layer is used to calculate the second level of the Convolutional Network.

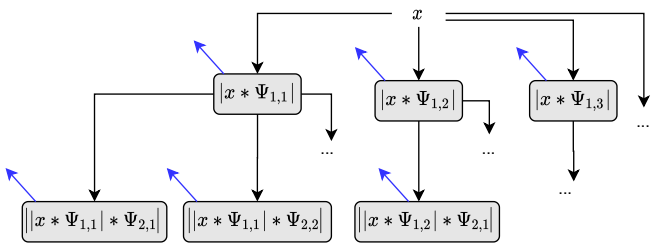


Figure 2: Scattering Transform Structure.

[2]It is an operator f that takes elements from set A to set B, which preserves the form (homeomorph). The kernel of this operator f is the inverse image of $0 \in B$.

At the second layer, the convolution of $|x * \Psi_{1,i}|$ with a second set of Wavelets, $\Psi_{2,k}$ results in a new set of coefficients. The index k represents the second frequency scale of the transform ($q = 2$), implemented by the second filter bank. Each second layer node comes from convolution-modulus with $\Psi_{2,k}$.

The cutoff frequency of Φ_T is half the bandwidth on the j-th sub-band. We propose to use the results of those convolutions to construct features used for classification.

3.4 Feature Selection Techniques

We propose, in this section, a set of methods for selecting features obtained with the Scattering Transform. We determine the features matrix employing six different strategies, briefly described below and detailed as follows:

- **Method A:** by taking the average of first order-path Scattering coefficients;
- **Method B:** by taking the energy of the first order-path Scattering coefficients;
- **Method C:** by taking all the first order-path Scattering coefficients;
- **Method D:** concatenating the averages of the first-order paths with the averages of the second-order paths;
- **Method E:** concatenating the energies of the first-order paths with the energies of the second-order paths;
- **Method F:** concatenating all first-order coefficients with all second-order coefficients.

3.4.1 Method A: Averages of first order-path Scattering coefficients

Let $S_{1,i} = |x * \Psi_{1,i}| * \Phi$ the first order-path coefficients of the i-th sub-band. Then, we compute the features by the averaging method ($A_{1,i}$), for the i-th sub-band, as

$$A_i = \frac{1}{N} \sum_{m=1}^{N} S_{1,i}[m], \tag{4}$$

where N is the number of coefficients at the i-th sub-band. We define the set of selected features for Method A as:

$$\mathcal{A} = \{A_i : i = 1, \cdots, J_1\}, \tag{5}$$

in which J_1 is the number of sub-bands at the first layer.

3.4.2 Method B: Energies of the first order-path Scattering coefficients

The Method B of feature selection consists on taking the energy of the first order-path Scattering coefficients. We compute the energy of the first order coefficients, B_i, as:

$$B_i = \sum_{m=1}^{N} S_{1,i}^2[m], \tag{6}$$

where N is the number of coefficients at the i-th sub-band. Then, we use the B_i to select features for each example. We define the set of selected features for Method B as:

$$\mathcal{B} = \{B_i : i = 1, \cdots, J_1\}, \tag{7}$$

in which J_1 is the number of sub-bands at the first layer.

3.4.3 Method C: All the first order-path Scattering coefficients

For Method C, we apply all the first order-path Scattering coefficients as features. Let $\mathcal{C}$ be all the set of all first-order Scattering coefficients, given by concatenating each first order sub-band Scattering coefficients, as follows:

$$\mathcal{C} = \{S_{1,i} : i = 1, \cdots, J_1\}, \tag{8}$$

in which J_1 is the number of sub-bands at the first layer.

3.4.4 Method D: Concatenation of the first order and second order-path averages of the Scattering coefficients.

Let $S_{2,i,k} = ||x * \Psi_{1,i}| * \Psi_{2,k}|*\Phi$ be the second order-path coefficients of the i first order sub-band and k second order sub-band. Then, for Method D, we compute $(D_{i,k})$, as:

$$D_{i,k} = \frac{1}{M} \sum_{m=1}^{M} S_{2,i,k}[m], \tag{9}$$

in which M is the number of coefficients at the i-th sub-band from first layer and k-th sub-band from second layer. Hence, we define the features vector for Method D as:

$$\mathcal{D} = \mathcal{A} \cup \{D_{i,k} : i = 1, \ldots, J_1;\ k = 1, \ldots, J_2\}, \tag{10}$$

in which J_2 is the total number of second-order Wavelets filters.

3.4.5 Method E: Concatenation of the first order and second order-path averages of the Scattering coefficients.

Let $S_{2,i,k} = ||x * \Psi_{1,i}| * \Psi_{2,k}|*\Phi$ be the second order-path coefficients of the i first order sub-band and k second order sub-band. Then, for Method E, we compute $(E_{i,k})$, as:

$$E_{i,k} = \sum_{m=1}^{M} S_{2,i,k}^2[m], \tag{11}$$

in which M is the number of coefficients at the i-th sub-band from first layer and k-th sub-band from second layer. Therefore, we define the features vector for Method E as:

$$\mathcal{E} = \mathcal{B} \cup \{E_{i,k} : i = 1, \ldots, J_1;\ k = 1, \ldots, J_2\}, \tag{12}$$

in which J_2 is the total number of second-order Wavelets filters.

3.4.6 Method F: All first and second order Scattering coefficients

For Method F, we compose the feature vector $\mathcal{F}$ by taking all the first and all the second order-path Scattering coefficients, as follows:

$$\mathcal{F} = \{S_{1,i} : i = 1, \ldots, J_1\} \cup \{S_{2,i,k} : i = 1, \ldots, J_1;\ k = 1, \ldots, J_2\}. \tag{13}$$

3.5 Classification

We perform the classification task considering the Scenarios presented in subsection 3.2 and the strategies of feature selection presented in subsection 3.4. The training and test sets are separated from the feature matrix and the label vector, i.e., 80% were used to train the classifier from the total number of instances and 20% for test.

We train the classification models using Ensemble Method (ENS), which present the best results for different NILM evaluations, as presented in [40]. This method comprises a set of classifiers whose individual decisions are combined in some way – normally averaging – to classify new examples. In classification methods, ensembles are often much more accurate than the individual classifiers that make them up [41]. Ensemble classification combines a set of trained weak learner models. It can predict ensemble responses for new data by aggregating predictions from its weak learners. This method can use different algorithms for sequential learning (weaker learning models), such as *AdaBoostM1, AdaBoostM2 Bag, GentleBoost, LogitBoost, LPBoost, LSBoost, RobustBoost, RUSBoost, Subspace*, and *TotalBoost*.

After training the classification model, we perform the prediction with the test subset (for each Scenario). Let n_c be the number of classes of the dataset. For COOLL, $n_c = 42$. The prediction, for each experiment, results in a $\mathbf{M}_{n_c \times n_c}$ confusion matrix:

$$\mathbf{M} = \begin{bmatrix} a_{1,1} & \ldots & a_{1,n_c} \\ \vdots & \ddots & \vdots \\ a_{n_c,1} & \ldots & a_{n_c,n_c} \end{bmatrix}, \tag{14}$$

whose rows represent the predicted classes and the columns represent the actual classes. From the confusion matrix we calculated two performance metrics: FScore and Accuracy. We chose these metrics because (i) they are metrics commonly used in other works in the literature (facilitates comparisons of results), and (ii) they are suitable metrics [42] for balanced datasets such as the one we use. The FScore, for each i-th class, is defined by

$$\text{FScore}_i = \frac{2 \times \text{Recall}_i \times \text{Precision}_i}{\text{Recall}_i + \text{Precision}_i}, \tag{15}$$

in which

$$\text{Precision}_i = \frac{a_{i,i}}{\sum_{k=1}^{n_c} a_{i,k}}, \tag{16}$$

Table 2: Scenarios and experimental cases details.

Scenario	Case	Number of 60Hz Cycles N_{before}	Number of 60Hz Cycles N_{after}	Total time per example	Examples per class	Samples per example	Total of examples	Fs [Hz]
SC1	1	5	5	166.67ms		1 365		
	2	10	10	333.33ms	20	2 731	840	8 192
	3	20	20	666.67ms		5 461		
SC2	1		5	83.33ms		683		
	2	0	10	166.67ms	20	1 365	840	8 192
	3		50	833.33ms		6 827		
	4		100	1.67s		13 653		
		Number of Cycles:						
SC3	1				10		420	
	2	240		4s	15	32 768	630	8 192
	3				20		840	
SC4	1	240		4s	20	16384	840	4 096
	2			4s		8 192		2 048
SC5	1	360		6s	20	49 152	840	8 192

and

$$\text{Recall}_i = \frac{a_{i,i}}{\sum_{k=1}^{n_c} a_{k,i}}. \tag{17}$$

The accuracy for each class is

$$\text{Accuracy}_i = \frac{\text{TP}_i + \text{TN}_i}{\text{TP}_i + \text{TN}_i + \text{FN}_i + \text{FP}_i}, \tag{18}$$

in which $\text{FN}_i = (\sum_{k=1}^{n_c} a_{k,i}) - a_{i,i}$, $\text{TN}_i = (\sum_{k=1}^{n_c} a_{k,k}) - a_{i,i}$, $\text{TP}_i = a_{i,i}$, and $\text{FP}_i = (\sum_{k=1}^{n_c} a_{i,k}) - a_{i,i}$.

From the FScore_i and Accuracy_i per class, we calculate the macro FScore and macro Accuracy by the expressions:

$$\text{FScore}_{\text{macro}} = \frac{1}{n_c} \sum_{i=1}^{n_c} \text{FScore}_i, \tag{19}$$

and

$$\text{Accuracy}_{\text{macro}} = \frac{1}{n_c} \sum_{i=1}^{n_c} \text{Accuracy}_i. \tag{20}$$

For simplification purposes, we refer to $\text{FScore}_{\text{macro}}$ and $\text{Accuracy}_{\text{macro}}$ as FScore and Accuracy, respectively, in the next sections.

4 EXPERIMENTAL ANALYSIS

We use the library *Wavelet Scattering*, from Matlab® R2021, to implement the Scattering Transform. We show in this section the results obtained from the five Scenarios derived from the COOLL dataset and described in subsection 3.2. We performed the experiments using both the proposed method (ST) and the Discrete Wavelet Transform (DWT) as a baseline comparison. The investigations follow the structure of Fig. 1. The choice of DWT as a baseline is because both ST and DWT are based on Wavelet filters [4]. We modify the input signal from the original dataset for all experiments to analyze the performance metrics for the classification process. The modifications are detailed in Tab. 2.

First, we present the setup of the Scattering Transform for the experiments. Then, we show the setup of the DWT baseline extraction method. At the end of this section, we present the classification Results and Discussions.

4.1 Scattering Transform (ST) Experimental Setup

For the feature extraction, we parameterized the ST as follows:

- **Sampling Frequency (Fs):** 8 192Hz for Scenarios SC1, SC2, SC3, and SC5. 4 096Hz and 2 048Hz for SC4;
- **Number of Layers (m):** 2 layers;
- **Number of Filter Banks:** 2;
- **Number of Wavelets per-octave, or Quality Factor (Q):** 8 for first layer, and 1 for second layer;
- **Type of Wavelet Filters:** Complex Morlet.

4.2 Discrete Wavelet Transform (DWT) Baseline Experimental Setup

For DWT, we use ten layers of detail signals. For each layer, we compute the energy of the Wavelet coefficients. Then, we use the ten energies of each detail layer as features for the baseline classification model. We add to these features the energy of the approximation coefficients [4], totaling 11 features for DWT baseline. The algorithm we use for DWT implementation is based on [43–45], implemented by function *wavedec*, on Matlab® r2021.

4.3 Feature Extraction and Classes Separability: Qualitative Analysis

We are interested in showing the class separability when using ST for feature extraction and comparing the results with the features obtained with the Wavelet baseline. We use the t-sne method, proposed in [46], to visualize the features, which were originally in the high-dimensional domain, in the Cartesian plane (dimension 2). The t-sne method consists of compute pairwise affinities $p_{j|j}$ with a so-called perplexity parameter $Perp$. For a given number of iterations, the t-sne method calculates the Low-dimensional affinities q_{ij}, given by:

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|\right)^{-1}}, \tag{21}$$

and the conditional probability $p_{i|j}$, given by:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n_{sig}}, \tag{22}$$

in which $p_{j|i}$, $p_{i|j}$ are conditional probabilities, n_{sig} is the total number of measures and the y_i are initially calculated by an approximation from the normal distribution. Let the low-dimension estimated vector be $\mathbf{y} = \begin{bmatrix} y_1, & y_2, & \ldots, & y_n \end{bmatrix}$. The cost function expression, based on the Kullback-Leibler divergence, is as follows:

$$C = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{23}$$

Then, the update of $\mathbf{y}$ is performed by the expression:

$$\mathbf{y}^{(t)} = \mathbf{y}^{(t-1)} + \eta \frac{\partial C}{\partial \mathbf{y}} + \alpha(t) \left(\mathbf{y}^{(t-1)} - \mathbf{y}^{(t-2)} \right), \tag{24}$$

where $\frac{\partial C}{\partial \mathbf{y}}$ is the gradient of the cost function, given by:

$$\frac{\partial C}{\partial \mathbf{y}} = 4 \sum (p_{ij} - q_{ij})(y_i - y_j) \left(1 + \|y_i - y_j\|^2\right)^{-1}. \tag{25}$$

We apply both ST and Wavelet baseline to extract features from each example, each Scenario, and each case presented in the table 2. Once this is done, we apply the t-sne method to reduce high-dimensional features to the (2D) plane. We use $Perp = 30$ for the t-sne experiments. As a result, we obtain a 2D representation for each case, Scenario, and Feature Selection Method. We choose the case of Scenario 5 to show in Fig. 3(a-d) because it represents the full dataset.

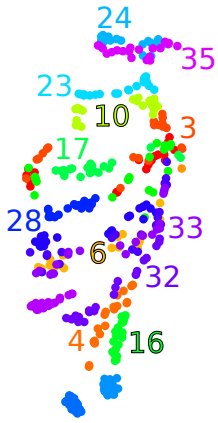
Fig. 3(a-g) shows the regions of interest (ROI), in which there is a more significant intersection between classes for each method of extracting and selecting features. Each method separates the classes differently; therefore, the subfigures do not show the same ROI, but the region with the highest class density for each selector. Note that the same sets of classes do not appear in each subfigure, because each subfigure represents a different ROI.

The numbers indicated in Fig. 3 correspond to the classes with the highest correlation in the feature space. Each of these class numbers corresponds to a specific appliance in the COOLL dataset, as shown in Tab. 1.

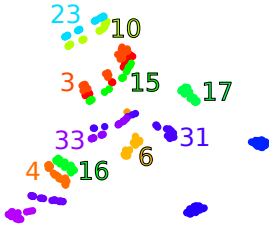
The appliances involved in the separability analysis of Fig. 3 correspond to class numbers 3 (Drill 3), 4 (Drill 4), 6 (Drill 6), 10 (Grinder 2), 15 (Hedge Trimmer 1), 16 (Hedge Trimmer 2), 17 (Hedge Trimmer 3), 23 (Planer 1), 24 (Router 1), 28 (Saw 1), 31 (Saw 4), 32 (Saw 5), 33 (Saw 6), and 35 (Saw 8) from Tab. 1. We observe that classes 3, 4, and 6 belong to the Drills category and classes 15, 16, and 17 to the Hedge Trimmer category. This behavior also occurs with the other Feature Selection Methods, as one can observe in Figs. 3b, 3c, 3d . Classes 28, 31, 32, 33, and 35 belong to the Saw category.

From Fig. 3a, one can observe that appliances 16 and 17 are located distant from each other, despite being from the same category (Hedge Trimmer). We obtain the same conclusion from 3 and 4 (Drills). In Fig. 3a, the Saws 28, 32, and 35 are close to each other, but there are intersections. The other Saw (35) is separated from 28 and 32. 24 (Router 1) and 35 (Saw 5) are appliances with completely different operation principles, but they are overlaped in Fig. 3a. 10 (Grinder 2) and 23 (Planer 1) are close in all sub-figures, unlike 16 (Hedge Trimmer 2) and 17 (Hedge Trimmer 3), which are far from each other. There is an evident intersection among 3 (Drill 3), 10 (Grinder 2), and 28 (Saw 1), which belong to different categories. The same occurs with 6 (Drill 6), 28 (Saw 1), and 33 (Saw 6).

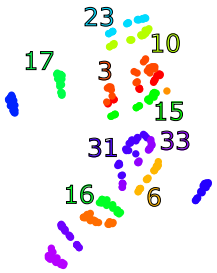
Figs. 3b, 3c and 3d have fewer overlapping regions than Fig. 3a. In Fig. 3b the pairs 23 (Planer 1) and 10 (Grinder 2), 3 (Drill 3) and 15 (Hedge Trimmer 1), 4 (Drill 4) and 16 (Hedge Trimmer 2) are close to each other, but there is no overlap. On the other hand, there is an intersection between 33 (Saw 6) and 31 (Saw 4).



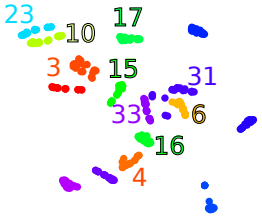
(a) Scenario 5, with Traditional Wavelet (baseline) for Feature Selection.



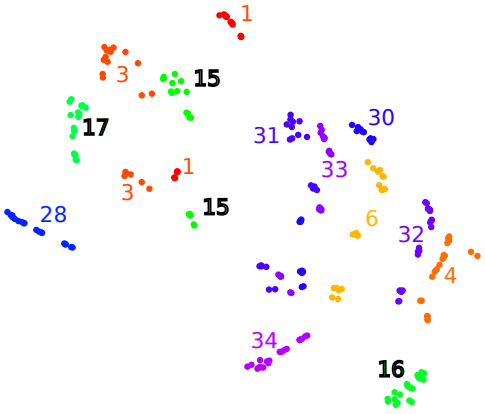
(b) Scenario 5, with Method A for Feature Selection.



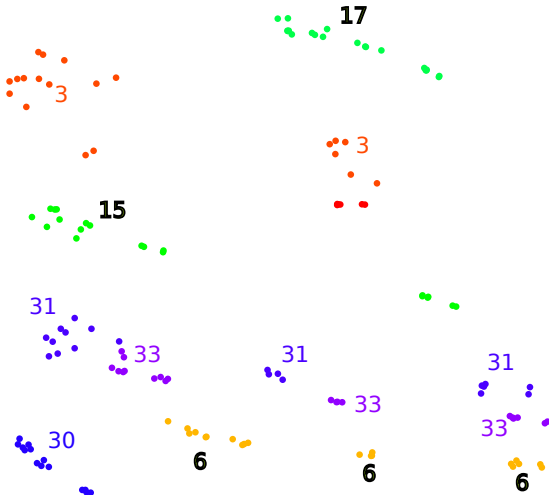
(c) Scenario 5, with Method B for Feature Selection.



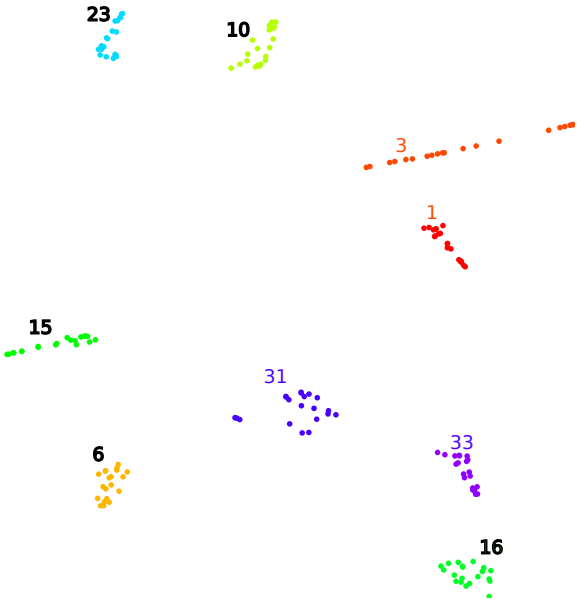
(d) Scenario 5, with Method C for Feature Selection.



(e) Scenario 5, with Method D for Feature Selection.



(f) Scenario 5, with Method E for Feature Selection.



(g) Scenario 5, with Method F for Feature Selection.

Figure 3: Analysis of the separability between classes, using the t-sne method. Each subfigure shows a region of interest of the full reduced-order feature map. We obtain the figures 3b-3g with Methods A-F for feature selection. The figure 3a was obtained with the traditional Wavelet transform. We noticed that the classes are considerably more separated in the Scattering Transform figures.

One can notice that the appliances 23 (Planer 1) and 10 (Grinder 2) are better separated when using Method B in Fig. 3c than when using other methods in Figs. 3a, 3b and 3d. There is an intersection between appliances 31 (Saw 4) and 33 (Saw 6) with Method B, but the separability is better than Methods A and the Traditional Wavelet.

The separation of 3 (Drill 3) and 15 (Hedge Trimmer 1) is more prominent with Method C in Fig. 3d than Methods A, B, and the Traditional Wavelets. Furthermore, 31 (Saw 4) and 33 (Saw 6) are not intersected, and the separation between the appliances 3 and 15 is much more evident than the other cases shown in Figs. 3a, 3b, 3c and 3d.

We noticed the separability of methods D, E and F (shown in figures 3e-3g) is more significant than methods A, B, and C and well higher than the Traditional Wavelet (baseline). This observation indicates that the greater number of features chosen, corresponding to methods D to F, contributes positively to increasing separability.

In summary, the conclusions of the separability analysis are:

- The intersection regions between appliances are smaller in ST selection methods;
- The tested ST-based Feature Selection Methods showed similar visual separability characteristics;
- Separability between appliances of the same category was met with ST methods.

4.4 Results and Discussions

We follow the structure of Fig. 1 and trained five classification models for each experiment. We use different training-test sets for each one of those classification models (five-fold cross-validation). Then, we perform the prediction five times for the Ensemble classifier, one for each different training-test set. Finally, we compute FScore and Accuracy for each trained model and each class. With these metrics, we calculate the macro FScore and Macro Accuracy.

Initially, we evaluate the influence of the feature selector type on classification metrics. For that, we perform the experiments according to Fig.1 of each Scenario and each case of Tab.2 using the six Feature Selection Methods presented in the subsection 3.4. For each Feature Selection Method (A to F), we average the macro FScores for all cases and Scenarios. Hence, we show these Global FScores (all cases average metric) in Fig. 4.

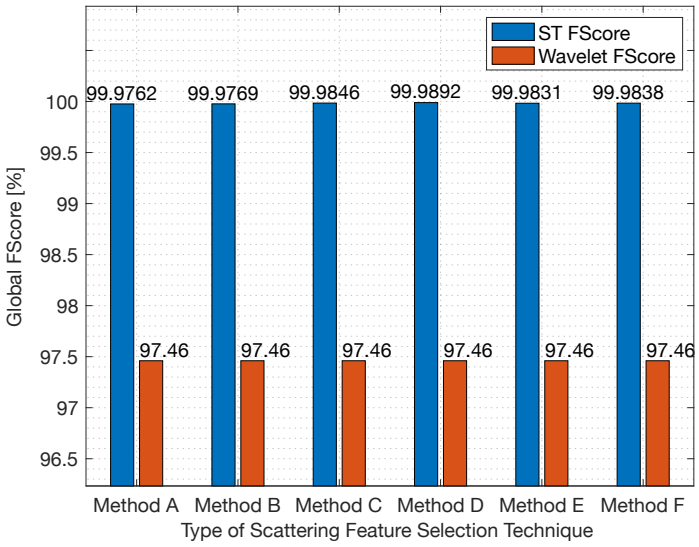


Figure 4: Average of the macro FScores for each method and Feature Extractor. One can note that ST overcame the Wavelet Baseline for all proposed Feature Selection Methods. There is no significant variation among the obtained global FScores when we modify the selection Method.

The results in Fig. 4 show that global FScore values (the average of all $FScore_{macro}$ from all cases and all Scenarios) do not depend on the method. In other words, there is no significant variation of Global FScore among all blue bars.

We show in Tab. 3, a comparison of the classification results obtained with each Feature Selection Method for the five analyzed Scenarios. We obtain the value of each bar in this figure by averaging all $FScore_{macro}$, for each case, in the relative Scenario.

We can observe in Tab. 3 that there is less than 0.06% variation of FScore in the classification, regardless of the Scenario and the feature extraction method. Such a low difference in the classification metric reinforces the argument that the performance of the ST for classification in the COOLL dataset is independent of the evaluated Feature Selection Methods.

We show in Fig. 5, the global FScore results for each Feature Selection Method, considering Scenarios 1 and 2. The objective of these experiments is to verify the influence of event detection on classification results. We obtain the blue bars in Fig. 5 from the average of the 5-folds of case 1, Scenario 1 (in which case there are five cycles before the turn-on annotation and five cycles after the turn-on annotation, totaling ten cycles). We obtain the red bars of Fig. 5 by averaging the 5-folds of case 2 of Scenario 2.

Table 3: Comparison of the average FScore for each Scenario considering different Feature Selection Methods. We obtain each of the bars in this figure by averaging the $FScore_{macro}$ of all cases in each Scenario. We notice that even the maximum variation of FScore is relatively small (0.06%). This indicates that the global classification result is independent of the Feature Selection Method.

Scn.	**Method A** FScore	**Method B** FScore	**Method C** FScore	**Method D** FScore	**Method E** FScore	**Method F** FScore
SC1	99.98 %	99.98 %	99.98 %	99.99 %	99.98 %	99.98 %
SC2	99.95 %	99.96 %	99.98 %	99.98 %	99.76 %	99.98 %
SC3	100.00 %	99.99 %	99.99 %	100.00 %	99.99 %	99.99 %
SC4	99.99 %	99.99 %	99.99 %	99.99 %	99.98 %	99.99 %
SC5	99.98 %	99.98 %	99.98 %	100.00 %	100.00 %	99.98 %

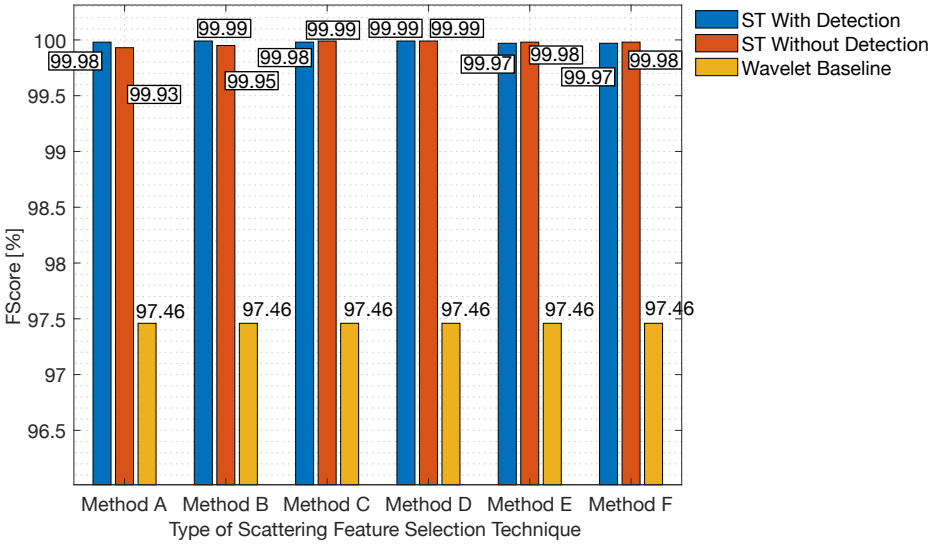


Figure 5: Comparison of FScore between ST with event detection (Scenario 1, case 1, represented with blue bars) and ST without event detection (Scenario 2, case 2, represented with red bars) for different Feature Selection Methods. The existence or not of event detection does not significantly interfere in the FScore obtained with the ST since the variation between the blue and red bars in the graph is less than 0.05%. In the figure, we also represent, in yellow, the mean value of the FScore obtained with the DWT (Wavelet Baseline).

One can observe from Fig. 5 that the proposed ST framework overcomes the DWT baseline for all Methods of features selection analyzed. The minimum increase of FScore of ST over the baseline is 2.46%, when comparing ST without detection with the baseline applying Method A for feature selection. Besides that, the maximum variation between ST With and ST Without Detection is 0.07%, which indicates that the classification results were practically independent concerning the turn-on instant of time.

So far, observe that:

- FScore does not significantly vary when using the different proposed ST Feature Selection Methods;
- The FScores with the ST surpass the results of the Wavelet baseline for all the cases tested;
- The position in time (time-stamp) at which the load is turned on does not significantly interfere with the classification results.

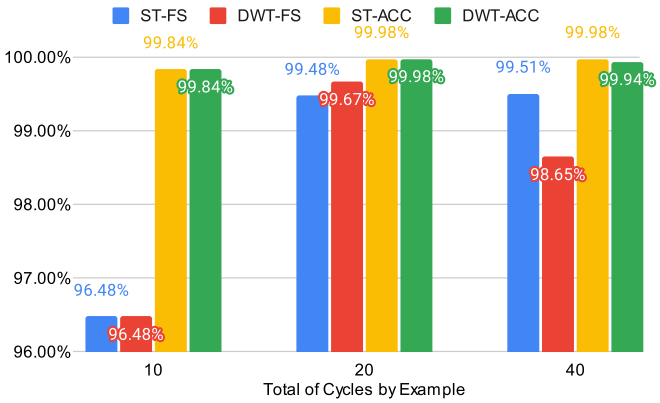
From subsection 3.4, one can observe that Method A and Method B produces smaller features vectors. This implies simpler training stages, smaller overall complexity, and faster processing time. Considering that we verified in Figs. 4, 5 and Tab. 3 that the classification performance does not vary substantially with the Feature Selection Method, we follow the experiments considering Method B for detailed analysis in subsection 4.4.1.

4.4.1 Detailed Experiments with Method B

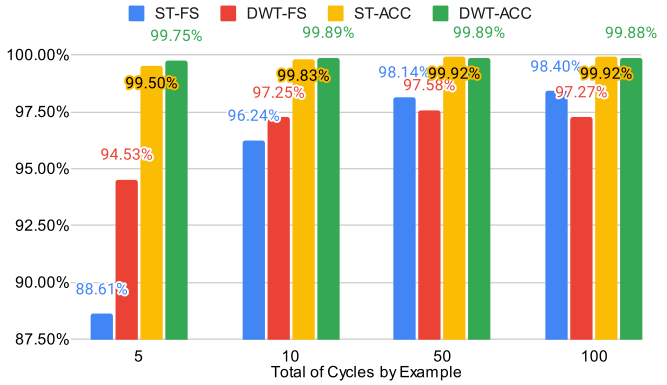
Tab. 4 shows detailed results obtained with Method B. We obtain these results considering the 5-fold classification models, with different examples for training and validation. We take the average of these five folds for each case and Scenario. We also calculate accuracy for comparison purposes.

Table 4: Macro Accuracies and FScores for all Scenarios and cases with Method B.

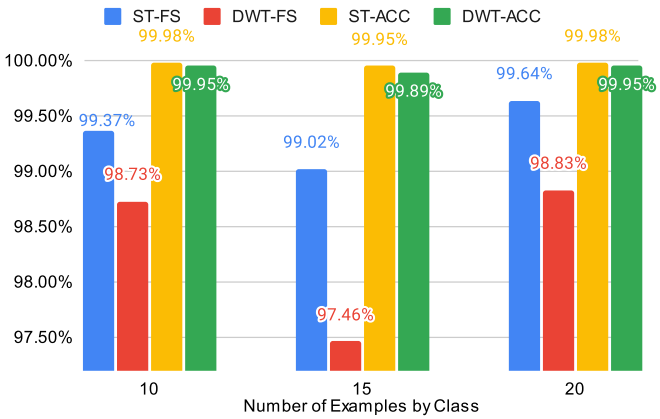
			Accuracy		FScore	
Scn.	**Case**	**Description**	**ST**	**DWT**	**ST**	**DWT**
SC1	1	5 cycles	99.84%	99.84%	96.48%	96.48%
	2	10 cycles	99.98%	99.98%	99.48%	99.67%
	3	20 cycles	**99.98%**	99.94%	**99.51%**	98.65%
SC2	1	5 cycles	99.50%	99.75%	88.61%	94.53%
	2	10 cycles	99.83%	99.89%	96.24%	97.25%
	3	50 cycles	**99.92%**	99.89%	**98.14%**	97.58%
	4	100 cycles	**99.92%**	99.88%	**98.40%**	97.27%
SC3	1	10 examples	**99.98%**	99.95%	99.37%	98.73%
	2	15 examples	**99.95%**	99.89%	**99.02%**	97.46%
	3	20 examples	**99.98%**	99.95%	99.64%	98.83%
SC4	1	2048Hz	**99.97%**	99.76%	**99.35%**	94.67%
	2	4096Hz	**99.98%**	99.83%	**99.67%**	96.12%
SC5	1	6s samples	**99.95%**	99.91%	**98.86%**	98.02%



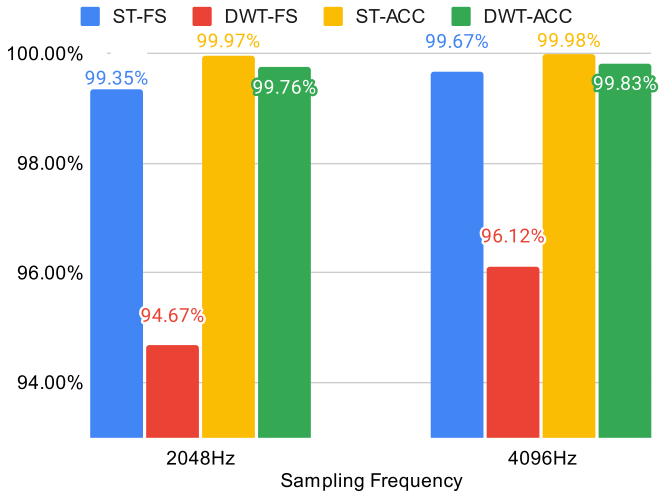
(a) Scenario 1 (SC1) Results: FScore (FS) and accuracy (ACC) with different number of cycles of the time window surrounding the *turn-on* event.



(b) Scenario 2 (SC2) Results: FScore (FS) and accuracy (ACC) with different number of cycles per example *after* the *turn-on* event.



(c) Scenario 3 (SC3) Results: FScore (FS) and accuracy (ACC) Examples per Class.



(d) Scenario 4 (SC4) Results: FScore (FS) and accuracy (ACC) with different sampling frequency (Fs).

Figure 6: Macro Accuracy (ACC) and Macro FScore (FS) results for different experimental Scenarios using Scattering Transform with Method B (ST) and Discrete Wavelet Transform (DWT) Feature Extractors.

Fig. 6a shows that with the smaller cycle window (10 cycles), both DWT and ST have the same accuracy (99.84%) and FScore (96.48%). For 20 cycles window, both methods present the same accuracy (99.98%), but DWT shows slightly better FScore than ST (99.66% vs. 99.48%). On the other hand, with 40 cycles, the ST method shows marginally better accuracy than

DWT (99.98% vs. 99.94%). In this case, ST presents a significantly better global FScore than DWT (99.51% vs. 98.55%).

Similar to SC1 results, in Scenario 2 (Fig. 6b), DWT presents slightly better FScores and Accuracy than ST for five cycles: FScore 94.53% vs. 88.61% and Accuracy 99.75% vs. 99.50%. For 10 cycles per sample: FScore 97.25% vs. 96.24% and Accuracy 99.89% vs. 99.83%. For the cases with 50 and 100 cycles per sample, ST outperforms DWT in both FScore and Accuracy. FScore 98.14% vs. 97.58% and Accuracy 99.92% vs. 99.89% for 50 cycles, and FScore 98.40% vs. 97.27% and Accuracy 99.92% vs. 99.88% for 100 cycles per sample.

Fig. 6c shows that ST has better performance than DWT for all cases of Scenario 3. The macro FScore is 0.65% better with ten examples per class, 1.6% with 15 examples, and 0.82% with 20 samples per class. The improvements in accuracy are smaller than the improvements in FScore: 0.03% in 10 examples per class, 0.06% in 15 examples, and 0.03% in 20 examples.

ST presents better FScore and Accuracy than DWT for the 2 cases of Scenario 4, as shown in Fig. 6d. This indicates that ST deals better with sub-sampling conditions than DWT for COOLL dataset (considering Ensemble classifier). Fig. 6d also shows that both ST and DWT increase their performance metrics (FScore and Accuracy) when the sampling frequency rises. From 99.35% to 99.67% and from 99.97% to 99.98% for ST (FScore and Accuracy, respectively), and from 94.67% to 96.12% of FScore and from 99.76% to 99.83% of Accuracy for DWT.

4.5 Discussion of The Results

We propose six feature selection methods from the Scattering transform. Using the t-sne visualization method, we showed that the proposed selection methods better separated the classes of the COOLL dataset, in relation to the DWT baseline. We chose the DWT as a baseline because it is the fundamental building block that defines the Scattering Transform. Furthermore, the separability results showed that the greater the number of features (D, E, and F methods), the more distant the classes are in the 2D plane (including correlated classes).

We modified the COOLL dataset to create five scenarios and evaluate the FScore and Accuracy metrics from them. We compared the proposed methods with the Wavelet baseline for each Scenario. Each Scenario was determined given the recurring difficulties encountered in Machine Learning problems: Unavailability of datasets with many training examples, low sampling frequencies, and shorter periods for each sample. The classification results comparing the A-F selection methods against each other showed a performance variation of less than 0.06%. This performance invariance led us to choose method B to perform more detailed tests since this method produces a reduced set of features.

The detailed results showed that the proposed method B outperformed the DWT for cases with fewer examples per class for training (confirming the theory presented in [7]) and for Scenarios with subsampling. On the other hand, ST did not outperform DWT for the cases in which we used the smallest time windows for each training sample.

5 COMPARISONS WITH STATE-OF-THE-ART APPROACHES

Other state-of-the-art works presented feature extraction strategies applied to COOLL dataset. For comparison purposes, we selected state-of-the-art methods that used similar frameworks to the one we proposed, with the same dataset, and performed the same classification task. Tab. 5 shows the comparison of the classification results metrics between those approaches and the proposed method. For all values of FScore and Accuracy in Tab. 5, the best Scenario for each method is considered.

Table 5: Comparison between literature approaches and the proposed method.

Reference	Method	FScore [%]	Accuracy [%]
[47]	Traditional Wavelets	-	92.00%
[25]	Hybrid V-I Trajectory	-	99.44%
[48]	Prony	-	98.00%
Proposed	Scattering Transform	**99.51%**	**99.98%**

As shown in Tab. 5, the proposed method presents, in terms of accuracy, an improvement of 8.67% in relation to [47], 0.54% to [25], and 2.02% in relation to [48].

6 CONCLUSIONS AND FUTURE WORKS

The classification results of NILM signal feature extraction and selection methods based on Deep CNN depend on the amount of data available for training. We propose a classification framework for NILM using the Scattering Transform in the feature extraction stage. In this framework, the weights of the Convolutional Network are not trained but are analytically calculated using Wavelets. We included variations in the dataset properties in the framework to test the performance of our proposal against reductions in the amount of training data. We vary the signal length, the number of examples per class, and sampling frequency. The main contributions of our work are: (i) apply the Scattering Transform to improve NILM state-of-the-art classification results; (ii) validate these improvements under dataset properties variations; (iii) visual low-dimensional separability analysis of ST-based feature extraction techniques; and (iv) proposal and evaluation of six different feature selectors based on ST applied to NILM signals.

In section 4.3, we presented a reduced-order representation for the features extracted from all examples of the COOLL dataset, both with the proposed method and with the baseline method (traditional Wavelet). These 2D representations, obtained with the t-sne [46] method, showed that the intersection regions between appliances are smaller in the proposed method than in the baseline for all tested feature selectors. This verification indicates that the separability of the proposed method is better than the baseline for all tested selectors. In addition, all the proposed Feature Selection Methods showed similar visual separability characteristics, which indicates robustness concerning the number of features chosen. Finally, t-sne showed that proposed ST-based Feature Selection Methods separated the appliances of the same category (different brands, for example), which is desirable in the NILM classification.

We tested the six proposed Feature Selection Methods with the five Scenarios in the table 2, using the structure of Fig. 1. The average of all macro FScores from each case and Scenario for each proposed Feature Selection Method did not vary significantly depending on the different proposed Feature Selection Methods. This result corroborates the separability inspection of section 4.3. The results presented in Tab. 3 showed that the FScore of proposed feature selectors surpasses the results of the Wavelet baseline for all Scenarios. Furthermore, the results from Tab. 3 showed that the appliance turn-on event does not significantly interfere with the classification results. This last verification is critical considering that many methods in the literature have such dependence.

We perform detailed experiments with Method B, applied to five different Scenarios and several cases per Scenario. From SC1, the proposed method outperformed the DWT baseline for cases with more cycles per example (20 and 40 for SC1; 50 and 100 for SC2). The experiments with Scenario SC3 showed that the proposed method outperformed DWT when reducing the number of examples per class for all evaluated cases. This result corroborates the theory proposed by [7], which established that Scattering Transform networks, as they are untrained convolutional networks given by time-shifting invariant representations, extract more discriminative information with smaller datasets. We also conclude, with SC4 results, that Scattering Transform presented better classification metrics for downsampling conditions in the dataset. Finally, the proposed method resulted in better macro accuracy when compared to state-of-the-art methods of feature extraction for NILM in the literature, as shown in Tab. 5.

The need for high-frequency data is a significant limitation of our proposed framework. Currently, high-frequency data is not directly found in traditional smart meters, limiting our proposal's embedded implementation. The implementation and analysis of our framework with low-frequency data is a proposal for future work. Also, we intend to implement the proposed framework in aggregated data and perform a comparison between Scattering Transform and Convolutional Neural Networks (CNN) in NILM disaggregation and classification tasks.

REFERENCES

[1] G. W. Hart. "Nonintrusive Appliance Load Monitoring". *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[2] M. H. Bollen and I. Gu. *Event Classification*, pp. 677–733. 2006.

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

[4] C. S. Burrus, R. A. Gopinath and H. Guo. *Introduction to Wavelets and Wavelet Transforms A Primer*.

[5] 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 Journal*, vol. 22, no. 1, pp. 501–509, 2022.

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

[7] S. Mallat. "Group Invariant Scattering". *Communications on Pure and Applied Mathematics*, vol. 65, no. 10, pp. 1331–1398, 2012.

[8] E. L. de Aguiar, A. E. Lazzaretti, B. M. Mulinari and D. R. Pipa. "Scattering Transform for Classification in Non-Intrusive Load Monitoring". *Energies*, vol. 14, no. 20, 2021.

[9] E. L. Aguiar, A. E. Lazzaretti and D. R. Pipa. "Performance of Scattering Transform Feature Extraction for Electrical Load Classification". pp. 1–8, 2021.

[10] R. Medico, L. De Baets, J. Gao, S. Giri, E. Kara, T. Dhaene, C. Develder, M. Bergés and D. Deschrijver. "A voltage and current measurement dataset for plug load appliance identification in households". *Scientific Data*, vol. 7, no. 1, pp. 1–10, 2020.

[11] D. P. B. Renaux, F. Pottker, H. C. Ancelmo, A. E. Lazzaretti, C. R. E. Lima, R. R. Linhares, E. Oroski, L. da Silva Nolasco, L. T. Lima, B. M. Mulinari, J. R. L. da Silva, J. S. Omori and R. B. dos Santos. "A dataset for non-intrusive load monitoring: Design and implementation". *Energies*, 2020.

[12] J. Powers, B. Margossian and B. Smith. "Using a Rule-Based Algorithm to Disaggregate End-Use Load Profiles from Premise-Level Data". *IEEE Computer Applications in Power*, vol. 4, no. 2, pp. 42–47, 1991.

[13] A. S. Bouhouras, P. A. Gkaidatzis, E. Panagiotou, N. Poulakis and G. C. Christoforidis. "A NILM algorithm with enhanced disaggregation scheme under harmonic current vectors". *Energy and Buildings*, vol. 183, pp. 392–407, 2019.

[14] M. Dong, P. C. M. Meira, W. Xu and C. Y. Chung. "Non-Intrusive Signature Extraction for Major Residential Loads". *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1421–1430, 2013.

[15] Y. Liu, X. Wang, L. Zhao and Y. Liu. "Admittance-based load signature construction for non-intrusive appliance load monitoring". *Energy and Buildings*, vol. 171, pp. 209–219, 2018.

[16] E. Sejdić, I. Djurović and J. Jiang. "Time–frequency feature representation using energy concentration: An overview of recent advances". *Digital Signal Processing*, vol. 19, no. 1, pp. 153–183, 2009.

[17] Y. Su, K. Lian and H. Chang. "Feature Selection of Non-intrusive Load Monitoring System Using STFT and Wavelet Transform". In *2011 IEEE 8th International Conference on e-Business Engineering*, pp. 293–298, 2011.

[18] M. Gray and W. G. Morsi. "Application of wavelet-based classification in non-intrusive load monitoring". In *2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 41–45, 2015.

[19] L. Guo, S. Wang, H. Chen and Q. Shi. "A Load Identification Method Based on Active Deep Learning and Discrete Wavelet Transform". *IEEE Access*, vol. 8, pp. 113932–113942, 2020.

[20] J. Bruna. "Invariant Scattering Convolution Networks". vol. 35, no. 8, pp. 1872–1886, 2013.

[21] T. Picon, M. N. Meziane, P. Ravier, G. Lamarque, C. Novello, J.-C. L. Bunetel and Y. Raingeaud. "COOLL: Controlled On/Off Loads Library, a Public Dataset of High-Sampled Electrical Signals for Appliance Identification", 2016.

[22] H. Liu. *Non-intrusive load monitoring: Theory, technologies and applications*. Science Press and Springer Nature Singapore Pte Ltd., Gateway East, Singapore 189721, first edition, 2019.

[23] T. Hassan, F. Javed and N. Arshad. "An Empirical Investigation of V-I Trajectory Based Load Signatures for Non-Intrusive Load Monitoring". *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 870–878, 2014.

[24] 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.

[25] B. M. Mulinari, D. P. de Campos, C. H. da Costa, H. C. Ancelmo, A. E. Lazzaretti, E. Oroski, C. R. E. Lima, D. P. B. Renaux, F. Pottker and R. R. Linhares. "A New Set of Steady-State and Transient Features for Power Signature Analysis Based on V-I Trajectory". In *2019 IEEE PES Innovative Smart Grid Technologies Conference - Latin America (ISGT Latin America)*, pp. 1–6, 2019.

[26] J. Kelly and W. Knottenbelt. "Neural NILM: Deep neural networks applied to energy disaggregation". *BuildSys 2015 - Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built*, pp. 55–64, 2015.

[27] 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". *International Journal of Electrical Power and Energy Systems*, vol. 129, no. January, pp. 106837, 2021.

[28] A. Moradzadeh, B. Mohammadi-Ivatloo, M. Abapour, A. Anvari-Moghaddam, S. Gholami Farkoush and S. B. Rhee. "A practical solution based on convolutional neural network for non-intrusive load monitoring". *Journal of Ambient Intelligence and Humanized Computing*, vol. 8, 2021.

[29] J. Kolter and M. Johnson. "REDD: A Public Data Set for Energy Disaggregation Research". *Artif. Intell.*, vol. 25, 01 2011.

[30] Y. Himeur, A. Alsalemi, F. Bensaali and A. Amira. "An intelligent nonintrusive load monitoring scheme based on 2D phase encoding of power signals". *International Journal of Intelligent Systems*, vol. 36, no. 1, pp. 72–93, 2021.

[31] 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 Transactions on Smart Grid*, vol. 13, no. 1, pp. 762–772, 2022.

[32] W. A. Souza, A. M. Alonso, T. B. Bosco, F. D. Garcia, F. A. Gonçalves and F. P. Marafão. "Selection of features from power theories to compose NILM datasets". *Advanced Engineering Informatics*, vol. 52, no. October 2021, pp. 101556, 2022.

[33] C. Nalmpantis, N. V. Gkalinikis and D. Vrakas. "Neural Fourier Energy Disaggregation". *Sensors*, vol. 22, no. 2, 2022.

[34] F. Zhang, L. Qu, W. Dong, H. Zou, Q. Guo and Y. Kong. "A Novel NILM Event Detection Algorithm Based on Different Frequency Scales". vol. 71, 2022.

[35] P. A. Schirmer and I. Mporas. "Device and Time Invariant Features for Transferable Non-Intrusive Load Monitoring". *IEEE Open Access Journal of Power and Energy*, vol. 9, no. January, pp. 121–130, 2022.

[36] P. Huber, A. Calatroni, A. Rumsch and A. Paice. "Review on deep neural networks applied to low-frequency nilm". *Energies*, vol. 14, no. 9, 2021.

[37] G. F. Angelis, C. Timplalexis, S. Krinidis, D. Ioannidis and D. Tzovaras. "NILM applications: Literature review of learning approaches, recent developments and challenges". *Energy and Buildings*, vol. 261, pp. 111951, 2022.

[38] A. Ruano, A. Hernandez, J. Ureña, M. Ruano and J. Garcia. "NILM Techniques for Intelligent Home Energy Management and Ambient Assisted Living: A Review". *Energies*, vol. 12, no. 11, pp. 1–29, 2019.

[39] J. Andén, V. Lostanlen and S. Mallat. "Joint Time-Frequency Scattering". *IEEE Transactions on Signal Processing*, vol. 67, no. 14, pp. 3704–3718, 2019.

[40] A. Lazzaretti, D. Renaux, C. Lima, B. Mulinari, H. Ancelmo, E. Oroski, F. Pöttker, R. Linhares, L. Nolasco, L. Lima, J. Omori and R. Santos. "A Multi-Agent NILM Architecture for Event Detection and Load Classification". *Energies*, vol. 13, no. 17, pp. 1–37, 2020.

[41] V. S. Cherkassky and F. Mulier. *Learning from Data: Concepts, Theory, and Methods*. John Wiley & Sons, Inc., New York, NY, USA, first edition, 1998.

[42] M. Grandini, E. Bagli and G. Visani. "Metrics for Multi-Class Classification: an Overview". pp. 1–17, 2020.

[43] Y. Meyer. *Wavelets and Operators*, volume 1 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1993.

[44] S. G. Mallat. "A theory for multiresolution signal decomposition: The wavelet representation". *Fundamental Papers in Wavelet Theory*, vol. II, no. 7, pp. 494–513, 1989.

[45] I. Daubechies. "The Wavelet Transform, Time-Frequency Localization and Signal Analysis". *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 961–1005, 1990.

[46] L. van der Maaten and G. Hinton. "Visualizing Data using t-SNE". *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[47] D. P. B. Renaux, C. R. E. Lima, F. Pöttker, E. Oroski, A. E. Lazzaretti, R. R. Linhares, A. R. Almeida, A. O. Coelho and M. C. Hercules. "Non-Intrusive Load Monitoring: an Architecture and its evaluation for Power Electronics loads". In *2018 IEEE International Power Electronics and Application Conference and Exposition (PEAC)*, pp. 1–6, 2018.

[48] H. C. Ancelmo, F. L. Grando, B. M. Mulinari, C. H. da Costa, A. E. Lazzaretti, E. Oroski, D. P. B. Renaux, F. Pottker, C. R. E. Lima and R. R. Linhares. "A Transient and Steady-State Power Signature Feature Extraction Using Different Prony's Methods". In *2019 20th International Conference on Intelligent System Application to Power Systems (ISAP)*, pp. 1–6, 2019.