A New Convolutional Neural Network-Based System for NILM Applications

Fabrizio Ciancetta, *Member, IEEE*, Giovanni Bucci, *Member, IEEE*, Edoardo Fiorucci, *Senior Member, IEEE*, Simone Mari, *Student Member, IEEE*, and Andrea Fioravanti, *Member, IEEE*

***Abstract*— Electrical load planning and demand response programs are often based on the analysis of individual load-level measurements obtained from houses or buildings. The identification of individual appliances' power consumption is essential, since it allows improvements, which can reduce the appliances' power consumption. In this article, the problem of identifying the electrical loads connected to a house, starting from the total electric current measurement, is investigated. The proposed system is capable of extracting the energy demand of each individual device using a nonintrusive load monitoring (NILM) technique. An NILM algorithm based on a convolutional neural network is proposed. The proposed algorithm allows simultaneous detection and classification of events without having to perform double processing. As a result, the calculation times can be reduced. Another important advantage is that only the acquisition of current is required. The proposed measurement system is also described in this article. Measurements are conducted using a test system, which is capable of generating the electrical loads found on a typical house. The most important experimental results are also included and discussed in the article.**

***Index Terms*— Convolutional neural network (CNN), disaggregation algorithm, energy management, load identification, load signatures, machine learning (ML), nonintrusive load monitoring (NILM).**

I. Introduction

THE reduction in electrical energy consumption requires the acquisition of ever more detailed data on individual users' power consumption. Electrical load planning and demand response programs are often based on the analysis of individual load-level measurements obtained from houses or buildings. By performing this analysis, the least efficient or malfunctioning devices can be identified and appropriate actions for reducing power consumption can be implemented. This analysis requires the measurement of energy consumed by each individual electrical appliance over time intervals of a few days or weeks.

The most usually applied method is based on the measurement of the total energy demand by users and the identification of power consumption by each individual load. This method requires measurement of voltage and current, or often measurement of current alone. The processing of measurements is based on a nonintrusive load monitoring (NILM) algorithm, which applies the "load disaggregation" technique. Research on load disaggregation began by Hart [1] in the early 1990s. Over the years, significant improvements have been made with respect to event detection and feature extraction techniques. Various techniques have been proposed in the literature, often based on complex processing techniques.

The load characteristics can be analyzed by observing the trend over time or by performing analysis in the frequency domain. The analysis is conducted under stationary or transient conditions. The type of analysis influences the choice of the sampling frequency. A brief description of the different characteristics used in the literature for NILM [2] is described as follows.

1) *Active Power P*: This characteristic is generally integrated by measuring the duration and the appliance's frequency of use.
2) *P–Q Plan*: Step changes in the active and reactive power Q allow easy identification of the ON/OFFstatus of high-power equipment.
3) *Combination of the P–Q Plane With Extended Transient Characteristics*: It is suitable in identifying devices with relatively long transients and significant peaks of power.
4) *Characteristics Based on P, Q, I, and V at Low Frequencies*: These combinations exhibit good performance in identifying ON/OFFappliances.
5) *P–Q and Harmonic Planes*: The harmonic content or the spectrum of high-frequency sampled currents is usually combined with the P–Q characteristics.
6) *Short-Time Fourier Transform (STFT):* The spectral envelopes allow the identification of nonlinear and variable load devices.
7) *V–I trajectories:* It is suitable in identifying loads, starting from the signal shape.
8) *Nonactive current*: It is suitable in identifying some special equipment.
9) *Unconventional features*: The analysis of voltage–noise spectrum or electromagnetic interference voltage noise has been proposed by several authors [3].

An almost complete summary of the relevant techniques has been reported in [2]. Unfortunately, there is still no suitable technique for the unequivocal discrimination of electrical loads.

The type of analysis to be applied also depends on the type of installation, which must be defined first. The NILM system can be placed in the switch box inside a property or even

Manuscript received July 11, 2020; revised October 16, 2020; accepted October 21, 2020. Date of publication November 2, 2020; date of current version December 18, 2020. The Associate Editor coordinating the review process was Dr. Arunava Naha. *(Corresponding author: Edoardo Fiorucci.)*

The authors are with the Department of Industrial and Information Engineering and Economics, University of L'Aquila, 67100 L'Aquila, Italy (e-mail: edoardo.fiorucci@univaq.it).

Digital Object Identifier 10.1109/TIM.2020.3035193

at a long distance from it. In the first case, the NILM is a miniaturized system with an embedded microcontroller. This system is capable of acquiring and processing the signals locally. In the second case, the system is a computer, which processes the data transmitted to a cloud database by the local smart energy meter. Local systems are capable of acquiring voltage and current (even with sampling frequencies of some kilohertz) processing them in real-time and displaying the results or storing them on a remote server [4]. Remote systems can only use data (normally related only to the active power) available in the cloud with measurement frequencies ranging from 1 to 3 Hz due to limited data transmission and storage capabilities [5].

Event detection and load disaggregation can be conducted using "traditional" algorithms [1], [6] or, as more widely proposed, by applying machine learning (ML) techniques [7]–[14]. ML [15] is a science related to algorithm design and applications. It involves computer learning from the data provided so that they carry out certain tasks. The ML systems are highly automated and self-modifying. These systems continue to improve over time with minimal human intervention as they learn using more data.

ML techniques can be applied using different approaches. In the first category, direct acquisition of the data on each individual user is required. These data can be used for system training (supervised methods) [16]. The second category includes techniques, which reduce the requirements for data training [7], [8]. For example, Parson *et al.* [5] proposed the tuning of generic device models to specific device instances using the signatures extracted from the aggregate load. Tang *et al.* [9] formulated load disaggregation as an optimization problem and attempted to minimize the change in switching events based on the knowledge of the appliance power models and the infrequency of switching events.

The third category of load disaggregation approaches obviates the need for prior knowledge of the appliances using unsupervised learning methods. These methods process sequences of low-frequency active power measurements [10]. Recent algorithms are based on hidden Markov models, factorial hidden Markov models, and source separation via non-negative tensor factorization [11]–[13]. These techniques are based on learning the parameters of an appliance after collecting the aggregated data for an adequate period, ranging from a few hours to a few months. In these systems, the number of appliances determines the dimension of the state space to be estimated. Therefore, it is directly related to model complexity and computational overhead. Since only low-frequency real-power measurements are used as features in these frameworks, the scalability of the methods is limited [14].

Deep learning (DL) is a type of ML that trains a computer to perform human-like activities, such as speech recognition, image identification, or prediction-making. Instead of organizing data to perform predefined equations [17], DL sets basic parameters on the data and trains the computer to learn on its own, by recognizing patterns using some levels of processing. There is a substantial difference between ML and DL. If an ML system makes an inaccurate prediction, an engineer must intervene and make changes [15], [17]. By adopting a DL model, the algorithm itself is capable of determining whether a prediction is accurate or not. DL applications use a layered structure of algorithms called the artificial neural network (ANN) since they vaguely imitate the interconnected structure of the human brain to provide multi-level functionality [18], [19].

In this work, a monitoring system installed inside a house is proposed. This system is capable of acquiring and processing the overall user current. The proposed solution is a DL-based NILM system, which adopts a particular type of ANN, namely, the convolutional neural network (CNN) [20]–[22]. This CNN is suitable for processing complex inputs such as multidimensional arrays. In the proposed application, the CNN processes the STFT of the total current. Although in most algorithms an event detection step is followed by device identification, in this work, event detection and classification of the related device are performed by the same and unique process.

The operational characteristics of the proposed system are verified by extensive measurements. The results obtained from field applications are also included and discussed.

II. Analyzed Load Signature Features

Although NILM has been investigated for over two decades, no systematic selection of the electrical characteristics that allow for unequivocal load discrimination has been presented yet. Therefore, the identification of the most significant set of electrical parameters that allow them to be distinguished remains one of the biggest challenges.

The load characteristics can be classified into stationary and transient-state characteristics based on the state of the measured waveform they represent. The load signature proposed in this article is based on the transient characteristics. More specifically, the transient characteristics are represented by the spectrogram of the derived rms current signal. By deriving the rms current, the steady states are filtered and all transient information is maintained. In this way, it is possible to classify an event, regardless of the load conditions present.

The load signature allows the action of a device to be identified when it comes into operation by measuring only the overall current of the monitored system. First, the current effective (rms) value is calculated by processing the acquired raw current with a sliding window technique, as follows:

$$I_{\mathrm{rms}\,(k)} = \sqrt{\frac{1}{N}\sum_{n=k}^{k+(N-1)} i_{(n)}^2} \tag{1}$$

where k is the kth measured current sample, N is the number of samples per cycle, $i_{(n)}$ is the sampled signal, and n is the summation index.

After this signal is derived, the impulsive signal $I'_{\mathrm{rms}\,(n)}$ can be obtained. The pulses of this signal represent the rms current transient states, as shown in Fig. 1. The position of the pulse in the derived signal identifies the moment in which a certain event occurred.

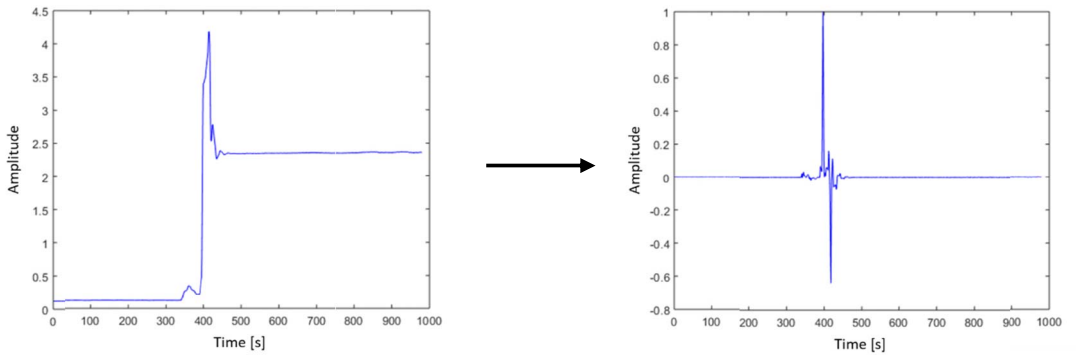


Fig. 1. Variation with time of the rms current I_{rms} (left) and its derivative I'_{rms}.

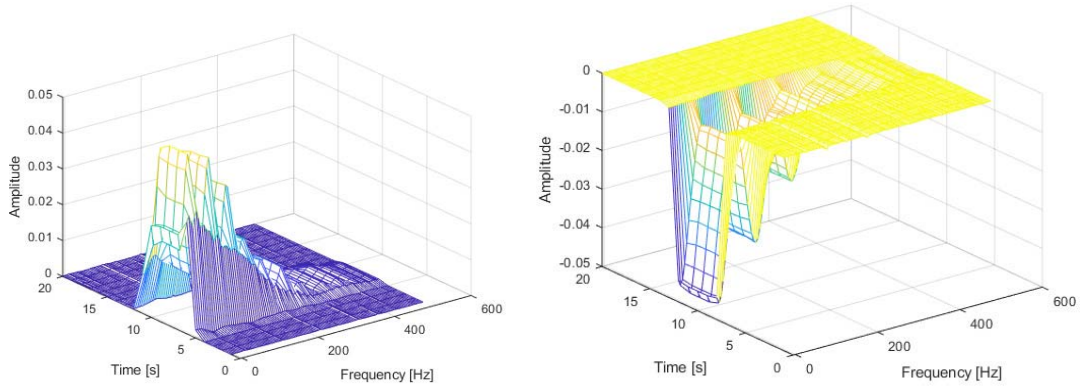


Fig. 2. Spectrograms obtained during switch ON (left) and OFF (right) of a microwave oven.

This impulsive signal is successively processed by the STFT through the following known transformation [23], [24]:

$$\text{STFT}_{(m,\omega)} = \sum_{n=-\infty}^{\infty} I'_{\text{rms}\,(n)}\, w_{(n-m)} e^{-j\omega n}. \tag{2}$$

Each specific event can be distinguished based on its spectral content and located in a precise time instant. In the above formula, w is the window function, and $I'_{\text{rms}\,(n)}$ is the sampled signal to be transformed (i.e., the derivative of the rms current value).

The current is processed cyclically at 1-s acquisition intervals following the described procedure. Each acquisition slot is processed (to calculate rms and the derivative) by adopting an overlap of 500 ms to ensure correct analysis. It is also processed for transient events, which can be fragmented into two successive slots. The STFT is implemented by processing ten-cycle (200 ms) windows with an overlap of 4/5 of the processing window.

To keep track of the type of event (switching ON or OFF), as the spectrograms of a device are often identical for both cases, the spectrogram described in (2) is multiplied by the sign of the cumulative sum, evaluated on the rms current signal, as follows:

$$S_N = \sum_{n=1}^{N} \left(I_{\text{rms}\,(n)} - I_{\text{rms}\,(n-1)}\right) \tag{3}$$

where $I_{\text{rms}\,(n)}$ is the rms value of the current described in (1), N is the number of samples, and S_N is the value of the cumulative sum. The final signal $S(i,\ j)$ can be obtained in the form of a 101×26 matrix, as follows:

$$\begin{aligned} S(i,\ j) &= \text{STFT}_{(m,\omega)} \cdot \text{sgn}(S_N) \\ &= \sum_{n=-\infty}^{\infty} I'_{\text{rms}\,(n)}\, w_{(n-m)} e^{-j\omega n} \\ &\quad \cdot \text{sgn}\left(\sum_{n=1}^{N} \left(I_{\text{rms}\,(n)} - I_{\text{rms}\,(n-1)}\right)\right). \end{aligned} \tag{4}$$

Two examples of the obtained spectrograms are shown in Fig. 2 for the case of a microwave oven.

The spectrograms obtained by processing the currents flowing through different loads are used as inputs to the neural network described in Section III. This network provides a response every 500 ms, indicating the presence or absence of events in the signal, and the type of device involved.

III. Deep Learning Systems

Based on the use of complex algorithms, DL provides systems, which are capable of analyzing massive amounts of data, recognizing patterns, and making predictions or decisions without being explicitly programmed to perform these tasks. In contrast, the DL systems operate by "learning and improving from experience."

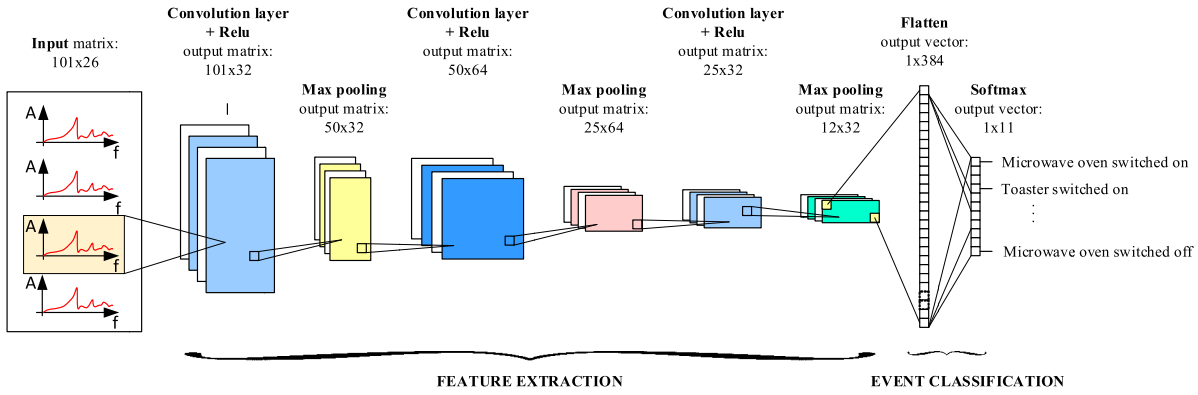


Fig. 3. Structure of the proposed CNN.

This learning ability does not imply that a DL system is capable of understanding what it is analyzing, learning from its experiences, and making decisions based on that understanding. Indeed, the real capabilities of such a system can be summarized considering that if particular behaviors have occurred in the past, it is possible to predict if they can happen again. It also means that if there are no past cases, then there are no predictions. Therefore, the analysis of previous cases is essential for achieving prediction results. Also, the number of cases is generally high.

Neural networks (NNs) are essential parts for the implementation of DL systems. NNs are capable of simulating a large number of densely interconnected brain cells in a simplified way using a computer. Originally, NNs were used for modeling biological neural systems without achieving significant results. Subsequently, other methodologies were used, achieving better performance. Today, NNs are generally software simulations implemented by conventional computer programming. They are called ANNs to differentiate from biological NNs [18], [19].

The basic computational elements of an ANN are the nodes (or neurons). These nodes are arranged in a series of layers, each of which is connected to the layers placed on either side. Some nodes receive various forms of information from outside sources (input nodes). Other nodes, which are placed on the opposite side of the network, indicate how the system responds to the supplied information (output nodes). Between the input and output nodes, there are one or more layers of hidden processing nodes.

Each node receives its input from other nodes or from an external source and calculates its output by applying a function, called activation function (AF), to the weighted sum of its inputs. A bias value is added to this sum. The application of the AF specifically introduces nonlinearities, aiming to emulate the way humans analyze real-world data [25]. The connection between one node and another is performed by a number (weight), which can be positive (one node excites another) or negative (one node inhibits another). The greater the weight, the greater the influence one node exerts on another. If the final sum is above a certain threshold, the node generates an output.

A. Proposed Convolutional Neural Network

In this work, a particular ANN type, namely, the CNN, is adopted [26] because of its capability of processing complex inputs such as multidimensional arrays. More specifically, CNNs are designed to exploit the intrinsic properties of some 2-D data structures, in which there is a correlation between spatially close elements (local connectivity). The CNNs are capable of reducing the number of operations required by converting the input into modules, which are easier to process. Thus, compared with ANNs, the number of parameters can be reduced. As a result, CNNs are widely used in the processing of audio and video signals [27]–[29].

In this work, a CNN suitable to process the current spectrograms was designed. The proposed system, which is shown in Fig. 3, includes different layers: an input level (for signal loading), three groups of convolution, Relu, and max pooling layers (for feature extraction from the input), and a group of flatten, fully connected, and softmax layers, which use data from the convolution layers to generate the output.

In the proposed CNN:

1) The first layer is the *input* layer, which holds the raw data related to the acquired current. These data are preprocessed through the STFT; each input is a 101×26 matrix (frequency $\times$ time).
2) The *convolution* layer, which is the core block of a CNN, detects the presence of specific features in the input spectrogram through the application of relevant filters. Instead of processing one matrix element at a time, the convolution layer collects restricted portions (square patches) of data and forwards them through a number of filters. Each of the applied filter seeks a different input parameter, such as a special behavior of the spectrum in a time instant or a particular evolution of a harmonic over time. A filter's spatial dimensions are smaller than the input signal. A filter is also a square matrix, equal in size to the patch, with a set of learnable parameters. Each convolutional layer applies a certain number of filters to its input. In this work, three convolutional layers with different numbers of filters (32 in the first convolution layer, 64 in the second, and 32 in the third) were applied. The filters were convolved (slided) across the width and

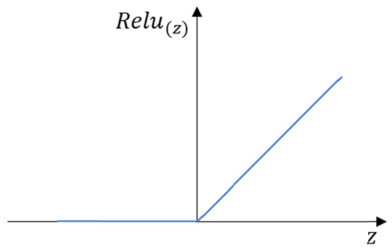


Fig. 4. RELU AF.

height of the input, and the dot products (between these two matrices) at any position were computed. The result is 2-D arrays (feature maps) that give the responses of the filters at every spatial position.

3) AFs can be either linear or nonlinear. Networks with nonlinear AFs are preferred, since they allow nodes to learn more complex data structures, even if they require more work in the initial configuration (training). In the proposed network, the AF used at each filter output is the *Relu* (z), an elementary rectified linear unit (RELU), whose piecewise linear characteristics are presented in Fig. 4 [25].
4) The activation maps are fed into a *pooling* (downsampling) layer, which processes one patch at a time, like a convolution. The pooling layer operates on each feature map independently and resizes it spatially using the *max* operation. The max pooling collects the largest value from one input patch, places it in a new matrix (next to the max values from other patches), and discards the rest of the information contained in the activation maps. This layer is inserted between successive convolution levels to progressively reduce (in this study by approximately 50%) the spatial dimensions of the representation (width, height), keeping the depth intact, to reduce the number of parameters and related calculations.
5) The *Flatten* layer transforms the entire pooled feature map matrix into a single vector (1×384).
6) The flattened feature vector is forwarded through a *fully connected* layer, which executes a multiclass classification using the following *softmax* AF:

$$\sigma_{(Z)\,j} = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, \quad \text{for } j = 1, 2, \ldots K. \tag{5}$$

This function maps K elements of the nonnormalized flattened feature vector z_k to a probability distribution over the predicted K output classes. The softmax applies an exponential function to each element z_k and then divides it by the sum of all these exponentials. Each K output value represents the probability that an input belongs to that particular class. In this study, five different loads were analyzed. Therefore, the problem's setting required the definition of ten different classes associated with each device's ON and OFFtransients and an additional class (number 5 in Table I), which is related to the "no event occurred" case. The definition of classes is shown in Table I.

TABLE I
CLASS DEFINITION

Class	Event
0	Microwave oven switched off
1	Oven switched off
2	Induction hob switched off
3	Toaster switched off
4	Light switched off
5	*No events detected*
6	Light switched on
7	Toaster switched on
8	Induction hob switched on
9	Oven switched on
10	Microwave oven switched on

Each of the 11 values in this *output* layer corresponds to a class score: the result is the class with the greatest probability. The number 10 indicates the number of events selected and represents the specific application, which was selected as a reference. It is possible to identify a larger number of events by changing the architecture of the CNN.

B. CNN Configuration

As a general remark, it can be considered that the correct functioning of the proposed NILM system can be ensured by optimally designing the architecture of the CNN network (number and type of layers and nodes). It is also necessary to adequately define the AFs and set the network by assigning the appropriate values for the filter weights. After designing the network, as described in Section III-A, the weights of each filter can be defined using the procedure described in the following.

When the CNN operation is started for the first time, the filters are configured by assigning default values to the individual weights. Thus, the results obtained cannot be optimal. The filter weights' adjustment is accomplished through the training process, which consists of two distinct phases [30].

1) *Forward Propagation*. A reference input signal is fed into the input layer. The nodes in the hidden layers apply the defined mathematical operations to these numerical values. The result is sent to the output layer, which generates the final result (classification).
2) *Backward Propagation*. By comparing the generated output with the expected one, the error value can be calculated. This calculation is based on which new (better) filter parameter values are defined.

This cycle is repeated for a new signal, which is obtained from a *reference* set of input data, until the error falls below a predetermined threshold. Obviously, the performance strongly depends on the set of reference examples selected for training. Therefore, these examples must be representative of the real type of the electrical load present. The more data available to the system, the more accurate the load identification will be.

Beyond its unusual name, the so-called training phase is, in fact, an optimization process capable of finding the best

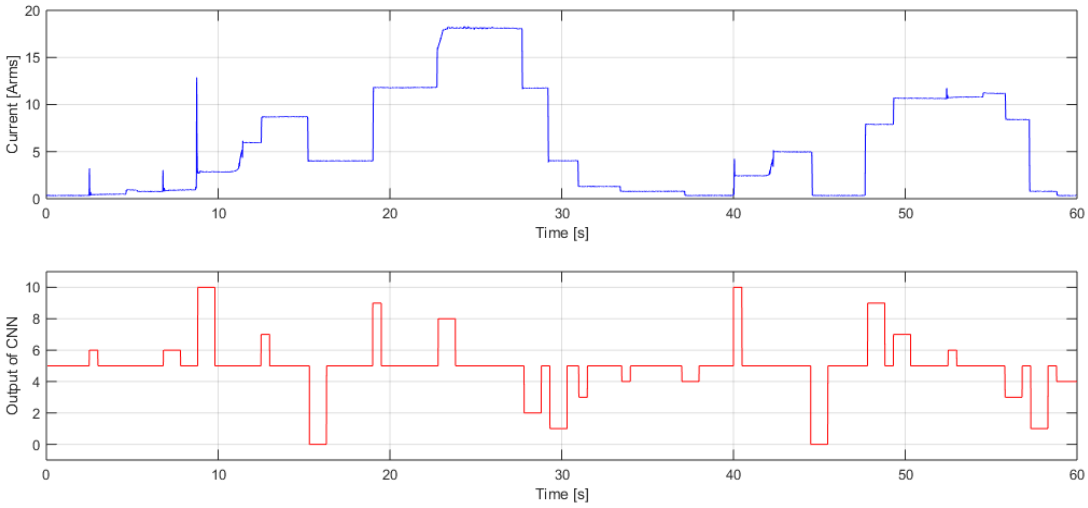


Fig. 5. Sequence of events: variation in the rms current (above) and detected events (below).

solution among all possible ones. During this process, an input data set is mapped to an output data set. The optimization process is based on a function, which represents the error occurring in the network. If the value of this function (which is called cost or loss function) is low (low loss), this means a better system performance. In this study, the function selected was the *logarithmic loss* [31].

At the end of this phase, the network is configured by assigning optimal values for the filter weights based on the analyzed reference data, that is, the values that allow the best event detection and the best load identification. The value of the achieved loss function during configuration was approximately 0.008.

After this phase, tests were conducted to verify the behavior of the system for a set of *test* signals, other than the reference set. The verification of the correct network configuration was completed by checking whether the system provided the correct answers for inputs other than those considered as a reference. In this work, the reference set adopted consists of 11 500 signals and the test set of 2876 signals. The time required for the network training was approximately 7 min.

IV. Experimental Results

As part of the development phase, the proposed algorithm was implemented and tested to evaluate its performance with real data.

A. Proposed System Setup

The measurement system includes an Agilent U2542A data-acquisition module with a 16-bit resolution. The sampling frequency was set to 10 kHz. The current signal was acquired using a TA SCT-013 current transducer. The CNN network was implemented on a desktop computer (based on the Windows 10 × 64 operating system) using the open-source Python 3.7 from Anaconda [32]. Python is the programming language mostly used in artificial intelligence (AI) applications due to the availability of numerous libraries for continuous data acquisition and processing.

To systematically evaluate the performance of NILM techniques, it is essential to use a set of reference data. The main tests were conducted on signals directly acquired from a real system because of the flexibility regarding both the sampling frequency and the generation of multiple events. Other tests were conducted on signals belonging to a public data set.

The proposed measurement system was installed on a test system, which was designed to generate electrical loads created by domestic users, as part of the "nonintrusive infrastructure for monitoring loads in residential users" research project. The plant, which is located in the Electrical Engineering Laboratory of the University of L'Aquila (I), allows the generation of electrical loads in a single or simultaneous way. These loads correspond to the loads generated by the most common household appliances and are integrated in a structure similar to that of a residential building to reproduce the real problems of conditioning and measurement of the signals.

B. Results Obtained With the Acquired Signals

Before conducting experimental measurements, the current measurement channel was calibrated using the Fluke 6100A power-supply standard. More specifically, a reference current was generated and applied to the TA SCT-013 current transducer. The current was acquired through the Agilent U2542A data-acquisition system, and the data were processed to calculate the rms value. In this way, the entire signal acquisition and processing path were tested. The system was calibrated with ten different current values, ranging from 2 to 20 A. The maximum uncertainty obtained was approximately ±1.3%.

Subsequently, the performance of the NILM system was assessed by conducting acquisitions, during which various loads were turned ON and OFFfor a total of over 519 events.

Regarding the NILM systems, no standard and consolidated techniques can be found in the literature to evaluate the performance of event detectors. Since the purpose of an NILM

TABLE II
ERRORS MEASURED DURING PROCESSING OF THE ACQUIRED SIGNALS

Number of processed windows	2883
Number of total events	519
Number of true power-on events ON_T	272
Number of true power-off events OFF_T	247
Number of windows with no events NE_T	2364
Number of power-on events identified ON_i	281
Number of power-off events identified OFF_i	247
Number of no-events identified as no-events NE_i	2355
Error in the identification of power-on events $E_{ident,on}$ (%)	3.31%
Error in the identification of power-off events $E_{ident,off}$ (%)	0.00%
Error in the identification of no events $E_{ident,ne}$ (%)	0.38%
Number of correctly classified power-on events ON_c	267
Number of correctly classified power-off events OFF_c	244
Error in the classification of power-on events $E_{class,on}$ (%)	1.84%
Error in the classification of power-off events $E_{class,off}$ (%)	1.21%

system is to disaggregate consumption for each of the devices in question, their performances were analyzed to verify the achievement of these objectives, which in summary are correct identification and classification of the events.

The first thing verified was the ability to correctly identify the ONand OFF events, which were also performed in rapid succession, and to correctly classify the device that produced a particular event. The proposed system was found to be capable of correctly identifying the insertion of loads, even by performing maneuvers at very short time intervals, up to approximately 500 ms. An example of the acquired signal representing the current variation for a 1-min window is presented in Fig. 5. This signal was extracted from the overall acquisition process. The effective value of the measured current and the relative system responses are also shown in this figure. It can be observed that the system is capable of detecting all the events.

Since the electrical load signature proposed in this work is based on the transitory characteristics, the system defined is not capable of classifying two different devices when the events associated with them are exactly superimposed. Tests were conducted to verify the system's ability to detect temporally close events. Fig. 6 shows that a toaster's switch on occurs approximately 400 ms after a microwave oven's switch on and the consequent system response. It is possible to observe how the system perfectly distinguishes the two events and classifies them correctly (events 10 and 7). A third event in the interval between 6 and 7 s can be observed in Fig. 6. This transient is normally produced by the microwave oven under examination approximately 3 s after its start. The double transient is very common in many household appliances. In an

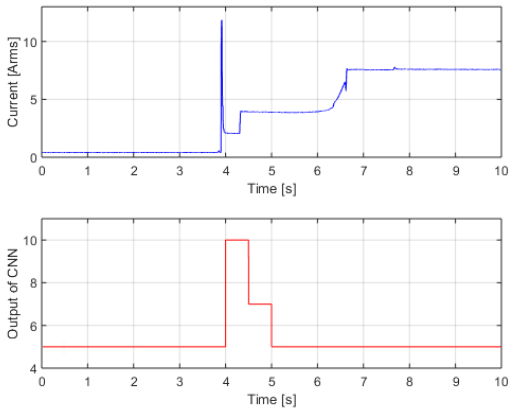


Fig. 6. Switch ON of a toaster 400 ms after the switch on of a microwave oven (above) and detected events (below).

oven, for example, the electronic section is activated first followed by the heating section. The proposed system was configured to filter this transient, classifying it as a nonevent (5), being linked to the first correctly recorded insertion. It was verified that the system is capable of identifying nearby events up to 300 ms.

The percentage of events, which were correctly detected, was compared with the total number of real events by adopting the following definitions.

1) Number of true power-ON events: ON_T
2) Number of true power-OFF events: OFF_T
3) Number of windows with no events: NE_T
4) Number of power-ON events identified: ON_i
5) Number of power-OFF events identified: OFF_i

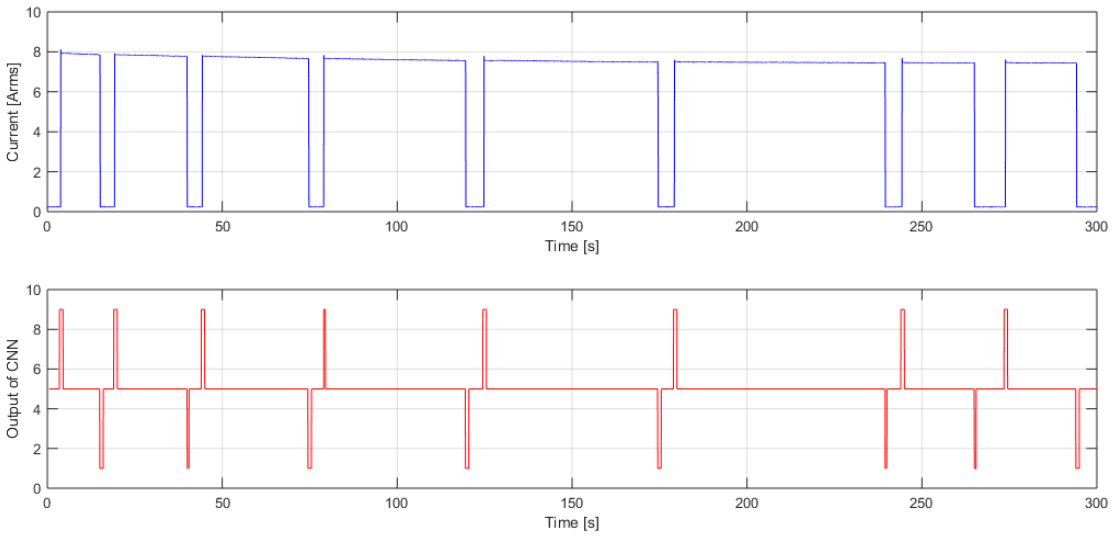
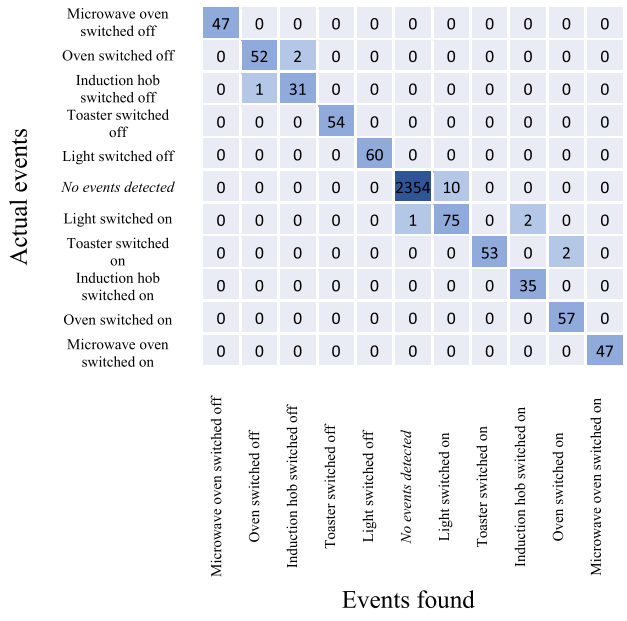


Fig. 7. Oven switch ON and OFF.

TABLE III
SCORES ACHIEVED WITH THE ACQUIRED SIGNALS

TP	*TN*	*FP*	*FN*	*Precision*	*recall*	*FPR*	*FPP*	*F1-score*
518	2354	10	1	0.981	0.998	0.004	0.019	0.989

TABLE IV
ERROR MATRIX



6) Number of no events identified as no events: NE_i
7) Number of correctly classified power-ON events: ON_c
8) Number of correctly classified power-OFF events: OFF_c
9) Error in the identification of power-ON events

$$E_{\text{ident,ON}}\% = \frac{\text{ON}_i - \text{ON}_T}{\text{ON}_T}100. \tag{6}$$

10) Error in the identification of power-OFF events

$$E_{\text{ident,OFF}}\% = \frac{\text{OFF}_i - \text{OFF}_T}{\text{OFF}_T}100. \tag{7}$$

11) Error in the identification of no events

$$E_{\text{ident},ne}\% = \frac{NE_i - NE_T}{NE_T}100. \tag{8}$$

12) Error in the classification of power-ON events

$$E_{\text{class,ON}}\% = \frac{\text{ON}_c - \text{ON}_T}{\text{ON}_T}100. \tag{9}$$

13) Error in the classification of power-OFF events

$$E_{\text{class,OFF}}\% = \frac{\text{OFF}_c - \text{OFF}_T}{\text{OFF}_T}100. \tag{10}$$

The absolute values relating to these parameters are presented in Table II.

From the above table, it can be observed that the system is capable of perfectly identifying all the OFFevents. An error of approximately 3% was recorded regarding the ONevents. This error value is essentially linked to the multiple oscillations of the current signal produced by some devices during the activation phase. Regarding the classifier performance, errors of 1.84% were recorded during the ONphases and 1.21% during the OFFphases, respectively.

For a better evaluation of the system performance, precision, recall, and F1-score metrics were also used [33]. These parameters were obtained using the number of true positive (TP),

TABLE V
ERRORS MEASURED DURING A SUCCESSION OF ON AND OFF SWITCHING

Event	Actual time	Measured time	Actual time interval [s]	Measured time interval [s]	Error [%]
1	0:03.8	0:04.0	11.2	11.5	2.68
2	0:15.0	0:15.5			
3	0:19.2	0:19.5	20.8	21.0	0.96
4	0:40.0	0:40.5			
5	0:44.2	0:44.5	30.5	30.5	0.00
6	1:14.7	1:15.0			
7	1:19.0	1:19.5	40.6	40.5	−0.25
8	1:59.6	2:00.0			
9	2:04.8	2:05.0	49.8	50.0	0.40
10	2:54.6	2:55.0			
11	2:59.2	3:00.0	60.3	60.0	−0.50
12	3:59.5	4:00.0			
13	4:04.3	4:04.5	20.7	21.0	1.45
14	4:25.0	4:25.5			
15	4:33.8	4:34.0	20.4	20.5	0.49
16	4:54.2	4:54.5			

false positive (FP), true negative (TN), and false negative (FN) as follows:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

$$\text{F1-score} = \frac{2x \text{ precision } x \text{ recall}}{\text{precision} + \text{recall}} \tag{13}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{14}$$

$$\text{FPP} = \frac{\text{FP}}{\text{TP} + \text{FN}}. \tag{15}$$

The calculated results are presented in Table III.

The system was capable of correctly classifying 511 of 519 overall events. The obtained basic accuracy during classification [33], which is defined as

$$\text{Acc}\% = \frac{\text{Correct matches}}{\text{Total possible matches}} 100 \tag{16}$$

is 98% with the acquired signals.

The errors obtained during the classification phase can be tabulated using the error matrix [34]. This is shown in Table IV. Each column of the matrix represents instances in a predicted class, whereas each row represents instances in an actual class.

The second type of tests is related to the measured duration between the ONand off events, being linked to the energy consumed. The results shown in Fig. 7 show a rapid succession of switch ON(event 9) and OFF(event 1) for the oven. The obtained results, which are tabulated in Table V, show a maximum relative error of 2.68%. However, it should be noted that this value is related to the reduced activation time (11.2 s), which was produced during the test. With longer intervals, for example 49.8 s, the relative error is reduced to 0.40%.

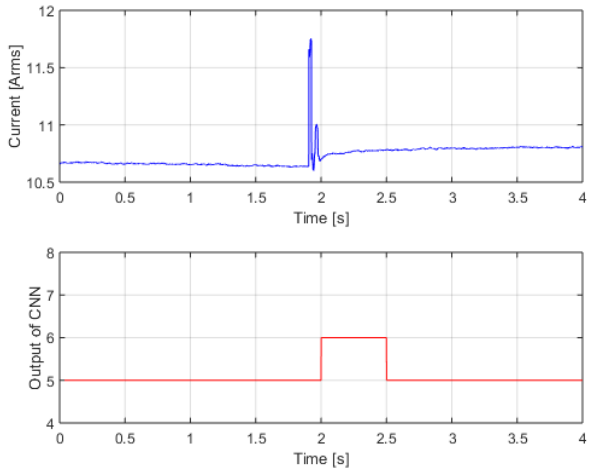


Fig. 8. Lamp switch ON.

The absolute error essentially depends on the duration of the window used for processing the STFT. The reduction of this window results in a reduction in this error. However, the processing time increases. The value of the window duration selected allows a good compromise, considering that the activation times of the loads are normally much greater than those selected in the test.

The system's performance was further evaluated by investigating its behavior with respect to distinguishing currents flowing in small loads in the presence of significant current values. To this purpose, Fig. 8 shows an acquisition in which a lamp is switched on (0.54 A) while more than 10 Arms are already flowing in the system (this represents 5% of the total load). The system proved to be efficient even under these operating conditions by identifying the event and correctly classifying the device.

TABLE VI
ERRORS MEASURED DURING THE PROCESSING OF THE BLUED DATA SET

Number of processed windows	11918
Number of total events	7901
Number of true power-on events ON_T	3887
Number of true power-off events OFF_T	4014
Number of windows with no events NE_T	4017
Number of power-on events identified ON_i	3890
Number of power-off events identified OFF_i	4006
Number of no-events identified as no-events NE_i	4022
Error in the identification of power-on events $E_{ident,on}$ (%)	0.08%
Error in the identification of power-off events $E_{ident,off}$ (%)	0.20%
Error in the identification of no events $E_{ident,ne}$ (%)	0.12%
Number of correctly classified power-on events ON_c	3569
Number of correctly classified power-off events OFF_c	3377
Error in the classification of power-on events $E_{class,on}$ (%)	8.18%
Error in the classification of power-off events $E_{class,off}$ (%)	15.87%

TABLE VII
SCORES ACHIEVED WITH THE BLUED DATA SET

TP	*TN*	*FP*	*FN*	*precision*	*Recall*	*FPR*	*FPP*	*F1-score*
7890	4011	6	11	0.998	0.998	0.001	0.001	0.998

C. Results Obtained With the BLUED Data Set

To make comparisons between the obtained results, tests were also conducted using the Building-Level fUlly-labeled data set for Electricity Disaggregation (BLUED), which is a residential electricity-usage public data set. This data set includes voltage and current measurements for a single-family house in the United States, sampled at 12 kHz for an entire week [35].

The current signal, which was extracted from the data set, was processed as described previously. Specifically, the spectrograms of 59 587 windows of 1-s duration were processed by the CNN, and 59 587 output values were obtained. This data set was randomly divided into two parts: 80% for network training and 20% (11 918 windows) for its performance evaluation. In the windows analyzed, 3887 switch-ON events and 4014 switch-OFF events were observed, whereas 4017 windows were free of events.

For the elaboration of the BLUED data set, the CNN was modified by extending the classification of Table I to 69 different classes to detect the ON and OFFstates of 34 different types of devices. The obtained results are presented in Table VI.

The scores achieved with the BLUED data set are tabulated in Table VII.

The obtained basic accuracy (16) with the BLUED data set is 87.9%. The data processing value of the data set is lower than that obtained by processing the signals acquired. This is due to the greater number of devices the system must identify.

V. CONCLUSION AND FINAL REMARKS

A typical NILM system involves three main processes: signal acquisition (current and/or voltage), event detection, and their classification. NILM approaches based on NNs have already been proposed in the literature. For example, in [36], a framework for using two NILMs for fault detection and isolation on a ship has been proposed. This system identifies the changes in a load by detecting transients. After this phase, two fully connected NNs identify the load from the extracted signature features. In [38], a technique based on the detection of events has been proposed, and it is based on four steps: 1) zero-cross detection; 2) current detection of similarity; 3) threshold evaluation; and 4) current event acquisition. Subsequently, the current is converted into an image and processed by a CNN.

In contrast to other NILM systems, which perform load classification based on the analysis of quantities also related to voltage (e.g., analysis in the P–Q or V–I plane [38]), the proposed system has the advantage of only measuring the overall current in a house. As a result, the complexity of the processing system is reduced. Another advantage is that

the measuring system can be implemented as a galvanically isolated system at low cost using a clamp current transducer.

In this work, the detection of an event and the classification of the related device were conducted by the same and unique process. The proposed system was implemented in Python's open-source development environment, thus reducing the system cost. The online system configuration (training) required approximately 7 min. The processing times measured were of the order of 105 ms for processing 1 s of acquired data (10K samples).

The proposed NILM algorithm allows the system to recognize a device, regardless of whether it operates singularly or in combination with other loads. The first results obtained after a large number of measurements appeared to be satisfactory, with error rates of approximately 3% for event detection and less than 2% for event classification.

The results obtained by processing the data available on the public BLUED data set appeared very encouraging. The value obtained for the F1-score was 99.8%, which is higher than that obtained with other systems using the same data set such as those proposed in [39] (91.5%) and [40] (93.2%).

REFERENCES

[1] G. W. Hart, "Nonintrusive appliance load monitoring," *Proc. IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec. 1992, doi: 10.1109/5.192069.

[2] 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, doi: 10.1016/j.enbuild.2017.06.042.

[3] A. Fioravanti, A. Prudenzi, G. Bucci, E. Fiorucci, F. Ciancetta, and S. Mari, "Non intrusive electrical load identification through an online SFRA based approach," in *Proc. Int. Symp. Power Electron., Electr. Drives, Autom. Motion (SPEEDAM)*, Sorrento Italy, Jun. 2020, pp. 694–698.

[4] M. Zeifman, C. Akers, and K. Roth, "Nonintrusive monitoring of miscellaneous and electronic loads," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2015, pp. 305–308.

[5] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," in *Proc. Nat. Conf. Artif. Intell.*, Toronto, ON, Canada, Jul. 2012, pp. 1–8. [Online]. Available: http://eprints.soton.ac.uk/id/eprint/336812

[6] G. Bucci, F. Ciancetta, E. Fiorucci, and S. Mari, "Load identification system for residential applications based on the NILM technique," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (IMTC)*, Dubrovnik, Croatia, May 2020, pp. 1–6.

[7] S. Barker, S. Kalra, D. Irwin, and P. Shenoy, "PowerPlay: Creating virtual power meters through online load tracking," in *Proc. 1st ACM Conf. Embedded Syst. Energy-Efficient Buildings*, 2014, pp. 60–69.

[8] R. Dong, L. J. Ratliff, H. Ohlsson, and S. S. Sastry, "Energy disaggregation via adaptive filtering," in *Proc. 51st Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Monticello, IL, USA, Oct. 2013, pp. 173–180, doi: 10.1109/Allerton.2013.6736521.

[9] G. Tang, K. Wu, J. Lei, and J. Tang, "A simple model-driven approach to energy disaggregation," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Venice, Spain, 2014, pp. 566–571, doi: 10.1109/SmartGridComm.2014.7007707.

[10] S. Pattem, "Unsupervised disaggregation for non-intrusive load monitoring," in *Proc. 11th Int. Conf. Mach. Learn. Appl.*, Boca Raton, FL, USA, Dec. 2012, pp. 515–520, doi: 10.1109/ICMLA.2012.249.

[11] M. Figueiredo, B. Ribeiro, and A. de Almeida, "Electrical signal source separation via nonnegative tensor factorization using on site measurements in a smart home," *IEEE Trans. Instrum. Meas.*, vol. 63, no. 2, pp. 364–373, Feb. 2014, doi: 10.1109/TIM.2013.2278596.

[12] D. Egarter, V. P. Bhuvana, and W. Elmenreich, "PALDi: Online load disaggregation via particle filtering," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 2, pp. 467–477, Feb. 2015, doi: 10.1109/TIM.2014.2344373.

[13] J. Z. Kolter and T. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation," in *Proc. 15th Int. Conf. Artif. Intell. Statist. (AISTATS)*, La Palma, Canary, 2019, pp. 1472–1482.

[14] R. Jia, Y. Gao, and C. J. Spanos, "A fully unsupervised non-intrusive load monitoring framework," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Miami, FL, USA, Nov. 2015, pp. 872–878, doi: 10.1109/SmartGridComm.2015.7436411.

[15] A. Ibrahim, A. Dalbah, A. Abualsaud, U. Tariq, and A. El-Hag, "Application of machine learning to evaluate insulator surface erosion," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 2, pp. 314–316, Feb. 2020, doi: 10.1109/TIM.2019.2956300.

[16] L. Mauch and B. Yang, "A new approach for supervised power disaggregation by using a deep recurrent LSTM network," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2015, pp. 63–67.

[17] M. Sewak, S. K. Sahay, and H. Rathore, "Comparison of deep learning and the classical machine learning algorithm for the malware detection," in *Proc. 19th IEEE/ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distributed Comput. (SNPD)*, Busan, Jun. 2018, pp. 293–296, doi: 10.1109/SNPD.2018.8441123.

[18] Y. Zhang, Y. Huang, W. Li, and M. Lin, "Application of artificial neural network and DS algorithm to calibration transfer of rice protein powder," in *Proc. 6th Int. Conf. Instrum. Meas., Comput., Commun. Control (IMCCC)*, Harbin, China, Jul. 2016, pp. 822–825, doi: 10.1109/IMCCC.2016.118.

[19] D. Jung, M. D. Nguyen, M. Park, J. Kim, and K. Mun, "Multiple classification of gait using time-frequency representations and deep convolutional neural networks," in *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 4, pp. 997–1005, Feb. 2020, doi: 10.1109/TNSRE.2020.2977049.

[20] P. Meier, K. Rohrmann, M. Sandner, M. Streitenberger, and M. Prochaska, "Convolutional neural networks for robust angular measurement with xMR sensor arrays," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, May 2019, pp. 1–6, doi: 10.1109/I2MTC.2019.8827051.

[21] R. C. Gonzalez, "Deep convolutional neural networks [Lecture Notes]," *IEEE Signal Process. Mag.*, vol. 35, no. 6, pp. 79–87, Nov. 2018, doi: 10.1109/MSP.2018.2842646.

[22] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton, "Sequence-to-point learning with neural networks for non-intrusive load monitoring," in *Proc. AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, Feb. 2018, pp. 2604–2611.

[23] W. Yuegang, J. Shao, and X. Hongtao, "Non-stationary signals processing based on STFT," in *Proc. 8th Int. Conf. Electron. Meas. Instrum.*, Xi'an, China, Aug. 2007, pp. 3–301, doi: 10.1109/ICEMI.2007.4350914.

[24] S. Zhang, D. Yu, and S. Sheng, "A discrete STFT processor for real-time spectrum analysis," in *Proc. IEEE Asia Pacific Conf. Circuits Syst.*, Singapore, Dec. 2006, pp. 1943–1946, doi: 10.1109/APCCAS.2006.342241.

[25] D. Stursa and P. Dolezel, "Comparison of ReLU and linear saturated activation functions in neural network for universal approximation," in *Proc. 22nd Int. Conf. Process Control (PC19)*, Strbske Pleso, Slovakia, Jun. 2019, pp. 146–151, doi: 10.1109/PC.2019.8815057.

[26] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Eng. Technol. (ICET)*, Antalya, Turkey, Aug. 2017, pp. 1–6, doi: 10.1109/ICEngTechnol.2017.8308186.

[27] K. Singh, A. Seth, H. S. Sandhu, and K. Samdani, "A comprehensive review of convolutional neural network based image enhancement techniques," in *Proc. IEEE Int. Conf. Syst., Comput., Autom. Netw. (ICSCAN)*, Pondicherry, India, Mar. 2019, pp. 1–6, doi: 10.1109/ICSCAN.2019.8878706.

[28] O. Sheremet, K. Sheremet, O. Sadovoi, and Y. Sokhina, "Convolutional neural networks for image denoising in infocommunication systems," in *Proc. Int. Sci.-Practical Conf. Problems Inf. Commun. Sci. Technol.*, Kharkiv, Ukraine, Oct. 2018, pp. 429–432, doi: 10.1109/INFOCOMMST.2018.8632109.

[29] T. Katona and B. Antal, "Automated analysis of radiology images using convolutional neural networks," in *Proc. 11th Int. Symp. Image Signal Process. Anal. (ISPA)*, Dubrovnik, Croatia, Sep. 2019, pp. 89–92, doi: 10.1109/ISPA.2019.8868764.

[30] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Dec. 1998.

[31] J. Brownlee. *Loss and Loss Functions for Training Deep Learning Neural Networks*0 Accessed: Jan. 28, 2019. [Online]. Available: https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/

[32] Anaconda Inc. (2020). *Anaconda Software Distribution*. Accessed: Feb. 13, 2020. [Online]. Available: https://www.anaconda.com/distribution/

[33] S. Makonin and F. Popowich, "Nonintrusive load monitoring (NILM) performance evaluation," *Energy Efficiency*, vol. 8, no. 4, pp. 809–814, Jul. 2015, doi: 10.1007/s12053-014-9306-2.
[34] J. L. Garcia-Balboa, M. V. Alba-Fernandez, F. J. Ariza-Lopez, and J. Rodriguez-Avi, "Homogeneity test for confusion matrices: A method and an example," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Valencia, Spain, Jul. 2018, pp. 1203–1205, doi: 10.1109/IGARSS.2018.8517924.
[35] K. Anderson, A. Ocneanu, D. Benitez, D. Carlson, A. Rowe, and M. Bergés, "BLUED: A fully labeled public dataset for event-based non-intrusive load monitoring research," in *Proc. 2nd KDD Workshop Data Mining Appl. Sustainability (SustKDD)*, Beijing, China, Aug. 2012, pp. 1–5.
[36] P. A. Lindahl, D. H. Green, G. Bredariol, A. Aboulian, J. S. Donnal, and S. B. Leeb, "Shipboard fault detection through nonintrusive load monitoring: A case study," *IEEE Sensors J.*, vol. 18, no. 21, pp. 8986–8995, Nov. 2018, doi: 10.1109/JSEN.2018.2869115.
[37] 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, doi: 10.1109/TCE.2020.2977964.
[38] T. Hassan, F. Javed, and N. Arshad, "An empirical investigation of V-I trajectory based load signatures for non-intrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 5, no. 2, pp. 870–878, Mar. 2014.
[39] M. Aiad and P. H. Lee, "Energy disaggregation of overlapping home appliances consumptions using a cluster splitting approach," *Sustain. Cities Soc.*, vol. 43, pp. 487–494, Nov. 2018, doi: 10.1016/j.scs.2018.08.020.
[40] A. K. Jain, S. S. Ahmed, P. Sundaramoorthy, R. Thiruvengadam, and V. Vijayaraghavan, "Current peak based device classification in NILM on a low-cost embedded platform using extra-trees," in *Proc. IEEE MIT Undergraduate Res. Technol. Conf. (URTC)*, Cambridge, MA, USA, Nov. 2017, pp. 1–4, doi: 10.1109/URTC.2017.8284200.

Fabrizio Ciancetta (Member, IEEE) received the Ph.D. degree in electrical and information engineering from the University of 'Aquila, L'Aquila, Italy, in 2009.

He is currently an Assistant Professor of electrical measurements at the University of L'Aquila. His research and technical interests cover several aspects related to distributed measurement systems, signal processing, and multilevel inverter; in these fields, he has authored about 70 journal and conference articles.

Giovanni Bucci (Member, IEEE) received the master's degree in electrical engineering from the University of L'Aquila, L'Aquila, Italy, in 1985.

He worked with Selenia Spazio, Rome, Italy, as a Designer of automatic test equipment until 1989. Since 2006, he has been a Full Professor of electrical measurements at the University of L'Aquila. His current research interests include smart sensors, multiprocessor-based measuring systems, digital measurement stations operating in real-time, and power measurements. He has authored more than 180 scientific articles in these fields.

Edoardo Fiorucci (Senior Member, IEEE) received the Ph.D. degree in electrical and information engineering from the University of L'Aquila, L'Aquila, Italy, in 2004.

He is currently an Associate Professor of electrical measurements with the University of L'Aquila. His research interests include power measurements, testing of electrical machines and power systems, pc-based instrumentation, measurement systems for power quality evaluation, and smart web sensors for distributed measurement applications. He has authored more than 100 scientific articles in these fields.

Dr. Fiorucci is currently an Associate Editor of the IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT.

Simone Mari (Student Member, IEEE) received the bachelor's and master's degrees in electrical engineering from the University of L'Aquila, L'Aquila, Italy, in 2017 and 2019, respectively. He is currently pursuing the Ph.D. degree in electrical measurements with the University of L'Aquila.

Andrea Fioravanti (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the University of L'Aquila, L'Aquila, Italy, in 2013 and 2016, respectively, where he is currently pursuing the Ph.D. degree in electrical engineering with a focus on resilience of electrical power systems.

In 2017, he was a Research Fellow with the Department of Industrial and Information Engineering and Economics, University of L'Aquila.