IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 30, 2020, accepted April 13, 2020, date of publication April 16, 2020, date of current version May 4, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2988366*

# Non-Intrusive Identification of Loads by Random Forest and Fireworks Optimization

**PAULO RICARDO ZAMBELLI TAVEIRA[1], CARLOS HENRIQUE VALÉRIO DE MORAES[1], AND GERMANO LAMBERT-TORRES[2], (Fellow, IEEE)**

[1]Institute of Systems Engineering and Information Technology, Itajuba Federal University, Itajuba 80305, Brazil
[2]Research and Development Department, Gnarus Institute, Itajuba 37500-052, Brazil

Corresponding author: Germano Lambert-Torres (germanoltorres@gmail.com)

**ABSTRACT** The control of expenses related to electricity has been showing significant growth, especially in residential environments. Monitoring of electrical loads that are turning on and off from home are often performed using smart plugs, providing to the consumers' information about operation intervals and power consumed by each device. Despite a practical solution to control and reduce electricity costs, it has a high cost due to the number of meters required. The high-cost problem can be worked around by using a non-intrusive load monitoring proposal (NILM), where voltage and current measurements are taken at the home entrance, in counterpart demand an extra processing step. In this extra step, various actions like computation of powers, identification of the occurrence of events, and identification of the status of which equipment (on/ off) must be made. The purpose of this work was the elaboration of a heuristic type event detector using floating analysis windows for locating stability zones on power signals after indicating a power change above a predetermined value. For that, tests of the best arrangement of event identifier data to identify which load has been added or removed from the monitored circuit are made. The proposed hybrid approach optimizes the processes using the Fireworks Algorithm (FWA) were used in the Random Forest classifier to improve classification performance. The proposed event identifier and classifier tests were performed on the dataset BLUED, which contains data collected at a North-American residence over one week. The event identifier results were compared with other publications that used different approaches, and the results of the classifications were compared to each other, using various data entry forms, and as an ideal classifier.

**INDEX TERMS** Non-intrusive load monitoring, BLUED, firework optimization algorithm, heuristic event detector, variable importance.

## I. INTRODUCTION

Each time more, the search for greater integration between energy suppliers, consumers, and government agencies (responsible for inspecting and regulating services and products related to electricity) has been done. The main idea behind this integration is to offer higher reliability safety and efficiency to the energy networks. For that, many actions have been taken, such as various equipment and techniques that have been applied to monitor energy consumption, strategies to minimize the possibility of black-outs, and estimate types of electrical charges that are connected to the electricity grid [1].

A great ally of this search for integration is the concept of Smart Grid, which is the interconnection of control and electricity systems, which provides data from the electrical system collected to be used in strategic planning and decision making of companies or responsible government agencies.

The data generated by identifying the loads coupled to the system can provide a consumption profile and habits of residents of the residences, such as times of higher consumption of electricity, which equipment is used simultaneously, what sequence of equipment occurs, among others [2] These profiles and habits can be used by energy suppliers to propose differentiated packages for each type of consumer, offering products more assertively. It can also be used as input data in a residential automation system and can predict user actions and act under equipment that has IoT technology.

To make this data available reliably and discriminated, it is generally necessary for several measuring instruments to be coupled to each equipment you want to monitor. This measurement strategy is known as intrusive measurement, and it

The associate editor coordinating the review of this manuscript and approving it for publication was Chang Choi.

is widely used due to the simplicity of its implementation. However, it ends by increasing the cost of data acquisition due to the number of meters to be equal to the number of monitored equipment.

Nonetheless, there is another measurement method called non-intrusive measurement (NIM), where few meters are coupled at the input of the system to be measured, reducing the number of equipment used and, consequently, the cost of measurement. However, this method does not make available from the data of each equipment separately, which makes the analyses more ''superficial'' when no other auxiliary technique is applied. Therefore, the data collected by this method require processing through computing techniques so that the maximum information can be extracted and used.

The data processing techniques used usually have the purpose of filtering, calculating, and segregating electrical characteristics of the equipment that were connected to the monitored system. Several methods are used, such as simple calculations of active and reactive power, Fast Fourier Transform (FFT), digital, and other filter applications [3]. The higher the accuracy of the techniques applied for data processing, the more assertive is the identification phase of the equipment and, consequently, the higher accuracy of the information extracted from the collected data.

This paper aims to classify residential loads by non-intrusive measurement. For that, large volumes of data must be treated, identifying events, and extracting useful information. This type of treatment can be classified into the Big Data area. The proposed approach applies two artificial intelligence (AI) techniques known as Random Forest [4], for the classification of residential loads, and Fireworks Optimization algorithm [5], to evaluate its performance.

The proposed methodology has the following main steps: (a) power calculations using the new IEEE 1459/2010 standard [6] to obtain training standards for the learning method; (b) load identification and its performance evaluation, both using the Random Forest technique; and (c) optimization of random forest technique parameters, through the Fireworks Optimization algorithm.

Several proposals have been elaborated over the years after the first work involving NIM in a residential load environment, in the early 90s [7]. Until today, this work is considered as one of the primary references in the area. Other seminal work was published a few years after applying the NIM concept in an industrial environment using neural networks to identify electrical charge patterns [8].

Observing the existing NIM approaches, notice that the sampling rate of electrical signals is one of the most relevant of NIM features. When they have a high sampling (between 1kHz and 100kHz), they allow the detection and analysis of the transition states of the equipment, but their implementations are more expensive [9]. However, projects with cheaper sampling equipment have lower sampling rates, usually less than 10Hz [10], and due to this fact (low cost), they can be more extensively applied [11].

The modes of extraction of characteristics of monitored equipment can be classified based on time or on frequency. When based on time, active power (P) and reactive (Q) are usually used. However, they are not the only ones. The construction of the PQ diagram allows a good perception of specific equipment depending on their resistance (R), inductance (L), and capacitance (C) [12]. And more, voltage and current [13], the trajectory of voltage and current (V−I) [14], and current peaks [15] can also be used.

When based on frequency, there is a need for a sampling rate of greater than or equal to 1kHz. When this principle is fulfilled, there are two fundamental applications for this purpose, the FFT [16], [17] and the Discrete Transformation Fourier (DTF). The DTF has low efficiency in a sampling frequency, not so high [18].

More recently, some works have drawn attention. Initially, a NIM approach develops to demand response programs for Demand-Side Management using a combinatorial PSO technique [19]. And then, the reference [20] shows a hybrid system, merging moving average, derivative analysis, and filtering analysis, and without any particular requirement information about the monitored appliances. The other work treats with a deep convolutional neural network [21]. This approach focuses on a residential consumer profile. And also, the reference [22] which shows a method using convolution neural networks and recurrent neural networks with a pinball quantile loss function to help in the training process. The results of this approach are compared with other deep neural networks.

Much of the work uses property databases, a fact that makes comparisons among approaches difficult. However, some works have been developed, generating datasets for tests, which are available for use, having distinct characteristics depending on the site where they were developed, sampling rate, and the number of monitored equipment, allowing comparisons among the methods. The most important are: REDD [23], BLUED [24], and UK-DALE [25].

## II. PRESENTATION OF THE USED AI TECHNIQUES

This section is divided into two main parts: one dedicated to searching techniques, including Random Forest (RF), and another about optimization techniques, including Fireworks Optimization (FWO).

### A. SEARCH AI TECHNIQUE

The search problem (or the problem to build a model using search techniques) is one oldest problem in AI. Many algorithms have been proposed to solve this type of problem, starting with brute-force and blind search methods (also named weak search methods), passing through heuristic search, and also involving neural networks [26], support vector machine [27], neighbors nearest [28], fuzzy C-means [29], among many others. Good overviews of the search problem can be found in [30], [31].

Another search method is RF, which is a machine learning method based on a collection of Decision Trees used for

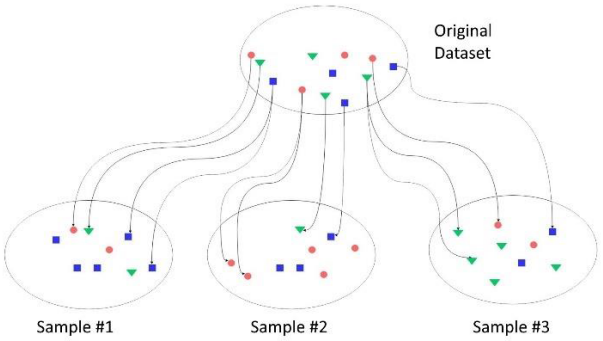


**FIGURE 1.** Exemplify of bootstrap.

both classification and regression [4]. The RF method has been extensively used in Big Data applications because the own technique can select and use the best input without any user action. Roughly speaking, in the method, many ''small'' decision-trees are build forming a forest. The answer of the final model for each set of inputs is the answer of the forest. RF has been used in several other applications such as remote monitoring [32], land classification [33], image classification [34], selection and classification of biological information [35], among others. Several different methodologies were tested, and RF has been chosen because it has a straightforward implementation, fast convergence, and to get good results.

### B. RANDOM FOREST

The first proposition of random ''decision'' forests was in 95 [4], the main idea was creating trees in oblique hyperplanes. With this, each tree makes the forest gain in accuracy and avoid overtraining [36]. However, the idea to have a forest using a large set of ''small'' trees appears in [37], and the plan to use Bootstrap Aggregating (or Bagging) in [38].

The general purpose of the bagging procedure is to reduce statistical variance in learning methods. To explain that, let's take on a set of $n$ independent observations $Z_1$, $Z_2$,..., $Z_n$, each observation with a variance $\sigma^2$. The variance of the $Z'$ (mean of the observations) is given by $\sigma^2/n$. Thus, to reduce variance and increase the statistical accuracy of a learning method, it is necessary to use various training sets, build separate prediction models for each, and finally calculate the average of results [39].

Then, with the prediction models $f^1(x)$, $f^2(x)$,..., $f^B(x)$ using $B$ sets of training separately and performing their mean in order to obtain a model prediction of low statistical variance, which can be represented by equation (1).

$$f_{mean}(X) = \frac{1}{B}\sum_{b=1}^{B} f^b(x) \tag{1}$$

The lack of practicality caused by the need for access to multiple training sets can be circumvented using bootstrap, taking repeated samples from the same training set, as shown in Figure 1.

With this, it is possible to generate $B$ distinct sets for training, and thus train $f^{*b}(x)$ models. And in the end, calculate the average of their outputs, and its representation is represented by equation (2).

$$f_{bag}(X) = \frac{1}{B}\sum_{b=1}^{B} f^{*b}(x) \tag{2}$$

However, the use of Bagging alone can generate several similar trees if there are correlated attributes, and one of them can be dominant over the others for the classification of specific samples. It is not advantageous since several trees would select this dominant attribute to break nodes at the beginning of the tree. And more, the average of several highly correlated trees does not result in a reduction in total variation.

For solving that, each node break operation considers a random amount of $m$ attributes, chosen in the total set of $p$ attributes. Then, the break considers only one of these attributes, and a new amount of $m$ attributes is selected for each of the next break operations. This procedure allows uncorrelated attributes, since, on average, $(p-m)/p$ attributes are not be considered, and the resulting trees are less dependent and, consequently, more reliable. And more, from the uncorrelation of attributes, it is possible to extract the important factor of each attribute for classification.

The main parameters for building an RF are: (a) *trees* - the maximum number of trees; (b) *minimum split size* - minimum size for breaking each nodes; (c) *maximum tree depth* - maximum depth of trees; (d) *features of split* - number of attributes to consider per breaks; (e) *minimum information gain* - to occur the break; and (f) *sub-sample ratio* - the relationship between the number of repetition observations and the total amount of observations.

The complexity of an RF can be simplified for equation (3), where $m$ is the number of trees used in the forest, $n$ is the number of instances used, and $v$ is the number of attributes used per node, not taking into account the process of selecting attributes.

$$O(v * m * n \log(n)) \tag{3}$$

When limited to the height of the tree, equation (3) can be rewritten to equation (4), where $h$ is the limiting height of the tree.

$$O(v * m * n * h) \tag{4}$$

### C. OPTIMIZATION TECHNIQUES

Optimization methods have two distinct purposes: the selection of parameters to optimize the performance of a system, and test and evaluation of quantitative models by the interactive process. In the first purpose, the quality of the system optimization depends on the chosen parameters, which are traditionally made by the user (or operator). When these parameters are appropriately chosen, then they can cause better performances; however, when these selected parameters are not appropriated can cause worse performance. This fact can be measured through system objectives or evaluation

"fitness" functions. In real systems, usually, the interaction between these parameters cannot always be performed analytically, so the operator should seek appropriate techniques, aiming at system optimization, or at least an approximation of an optimal point.

In the second purpose, test, and evaluation of quantitative models by the interactive process, a model is built, and then data is collected and used to test this model through a fitness function. The result can be called the error function (or misfit function). When there is a discrepancy between the results of the model and the desired results, the model is modified. This process is repeated until the problem to be solved, a pre-established interaction limit is reached, or there is a no different response.

Usually, complex problems, which require optimization methods, have multiples objectives to accomplish, generating a class of techniques named Multi-Objective Optimization (MOP). In these problems, rarely a single solution exists, but rather a set of solutions that create a combination of optimal solutions called Pareto border [40].

Among the various existing optimization methods, there is a type of optimization known as Swarm Intelligence [41], which has presented big success to find optimal global solutions in multimodal spaces. These methods consist of using a population of individuals, where each individual is the representation of a combination of state variables for a given problem, which interact with each other and with the environment. Each individual represents a possible solution of the problem, and they follow an objective function of given by the problem, trying to converge to an optimal point in the environment in which they are, thus generating an optimal solution.

Some examples of algorithms of this type of optimization are: (a) stochastic diffusion search [42], ant colony optimization [43], and swarm particle optimization [44]. Most intelligent swarm algorithms are bio-inspired; that is, the basic idea of the optimization process they use was copied from the observation of the behavior of animal colonies, insects, and other living beings in nature [45].

Another newer algorithm is the Fireworks Optimization (FWO), inspired by the explosion of fireworks, proposed in [5]. This method is not a bio-inspired algorithm; however, it has many similar characteristics to Swarm Intelligence, like a representation of individuals, interactions among individuals and with the environment, among others. FWO method is used in this work to optimize the parameters of the RF training process, as explained in this next section.

### D. FIREWORKS OPTIMIZATION

The FWO algorithm starts from the explosion of fireworks in random locations when a portion of sparks fills the space around each site of the explosions. The explosion process performs a search around a specific point. The algorithm uses $N$ vectors of $x_i^G$ parameters, with $D$-dimensions, as the basic population of each generation. The parameter $i$ ranges from 1 to $N$, and the parameter $G$ is the indicator of generations.

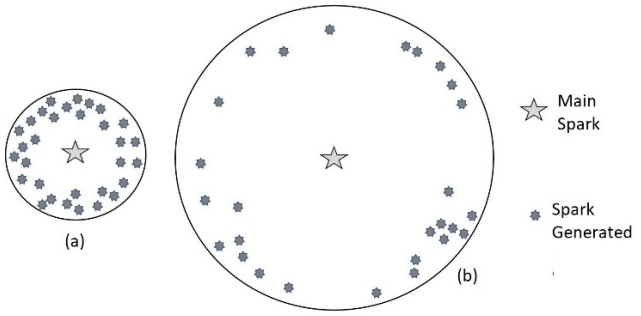


**FIGURE 2.** Exemplify of FWO algorithm: (a) good firework, and (b) bad firework.

Each individual in the population goes through the explosion process, thus generating sparks around them. And the two best sparks are selected for the next generation. Below, the main phases of the FWO algorithm are more thoroughly addressed: explosion, mapping, Gaussian sparks and selection.

#### 1) EXPLOSION

In a pyrotechnic show, two possible behaviors of the fireworks can be observed. When well made, many sparks are generated around the initial explosion, and when poorly made, only a few sparks are generated well dispersed from the blast point. The same logic is used in the FWO algorithm when the explosion is carried out in a promising area, that is, close to an optimal point, several sparks close to this point occur, generating a good firework (Figure 2(a)). However, if the explosion occurred far away from an optimal point, the process of scattering the sparks is larger, generating a bad firework (Figure 2(b)) [46].

Assuming an FWO algorithm implemented for minimization of a problem as expressed by equation (5).

$$\min f(x), \quad x_{min} \leq x \leq x_{max} \tag{5}$$

where $f(x)$ is the fitness function to be minimized, $x = x_1, x_2, \ldots, x_d$ are locations in space, and $x_{min}$ and $x_{max}$ are the limits of the search space. When an inner spark explodes, this spark is extinguished, and other sparks appear around where it was. The explosion strategy is part of this idea to produce new individuals by an explosion. Two parameters need to be determined: the number of sparks and the sparks amplitude. The first parameter is defined by equation (6).

$$S_i = S' * \frac{Y_{max} - f(x_i) + \varepsilon}{\sum_{i=1}^{N} (Y_{max} - f(x_i) + \varepsilon)} \tag{6}$$

where $S_i$ represents the number of sparks generated by an individual in the population, $S'$ is the maximum number of sparks, and $i$ ranges from 1 to $N$. To control the total number of sparks, $S'$ is maintained constant. Assuming the purpose of optimization is to minimize a function, the variable $Y_{max}$ represents the worst value of the fitness function in the current generation, and $f(x_i)$ represents the fitness function value

of individual $x_i$. The last parameter $\varepsilon$ serves exclusively to prevent the denominator from becoming zero.

Usually, the number of not very satisfactory results can be high if $S_i$ is too large [5]. So, to avoid this, it is necessary to create limits for $S_i$ according to equation (7).

$$S_i(x) = \begin{cases} (\alpha * S_i), & if\ S_i < \alpha * S_i \\ (\beta * S'), & if\ S_i > \beta * S_i \\ S'_i, & \text{otherwise} \end{cases} \tag{7}$$

where $\alpha$ and $\beta$ are constant and $\alpha < \beta < 1$. Typical values with good experimental results are $\alpha = 0.04$ and $\beta = 0.8$.

The second parameter in this strategy is sparks amplitude, given by equation (8).

$$A_i = A' * \frac{f(x_i) - Y_{min} + \varepsilon}{\sum_{i=1}^{N} (f(x_i) - Y_{min} + \varepsilon)} \tag{8}$$

where $A_i$ represents amplitude generated by the explosion of an individual $x_i$, $A'$ is the maximum amplitude, and $i$ ranges from 1 to $N$. With the aiming to control the amplitude, $A'$ is maintained constant. The best value of the $Y_{min}$ fitness function in the current generation is used to calculate amplitude. The last parameter $\varepsilon$ serves exclusively to prevent the denominator from becoming zero. When an individual is very close to the boundaries of the problem, the generated sparks can be placed outside the search space. In this case, the mapping limits are used to keep the spark within the search space.

In addition to the explosion strategy, another strategy to generate spark is used, named Gaussian sparks strategy, which is discussed below in this paper.

#### 2) MAPPING

The mapping strategy ensures that all individuals created remain in the search space. If individuals appear outside the search space, they are mapped into the search space through equation (9).

$$x_i = x_{min} + |x_i| \,\%(x_{max} - x_{min}) \tag{9}$$

where $x_i$ represents the position of a spark that appears outside the search space, $x_{min}$ and $x_{max}$ are the maximum and minimum values, respectively, of the search space that a spark can take. The % symbol represents the rest of the integer division.

#### 3) GAUSSIAN SPARKS

In order to maintain diversity among individuals in a population, the Gaussian spark strategy is used to generate sparks with Gaussian distribution. Assuming that an individual's current position is given by $x_k^j$, the Gaussian explosion is given by equation (10).

$$x_k^{j+1} = x_k^j + g \tag{10}$$

where $g$ is a random number in Gaussian distribution, as shown in equation (11), meaning, the parameter $g$ follows a Gaussian distribution with average value and standard deviation equal to 1.

$$g = Gaussiana(1, 1) \tag{11}$$

After a normal explosion and a Gaussian explosion, it is necessary to choose individuals for the next generation. Usually, a selection based on distance is used.

#### 4) SELECTION

In the selection of individuals for the next generation, the best individual, who had better value in the fitness function, is always selected first. Then the next $(N-1)$ individuals are selected based on the distance to the others. Individuals more distant from others are more likely to be selected compared to those closest to others.

The general distance between two positions in space can be calculated according to equation (12).

$$R(x_i) = \sum_{j \in K} d(x_i, x_j) = \sum_{j \in K} \|x_i - x_j\| \tag{12}$$

where $x_i$ and $x_j$ $(i \neq j)$ are positions of the individuals $i$ and $j$, and $K$ is the set of all current positions. For the calculation of distances, any method can be used, such as Euclidean distance and Manhattan distance. The Euclidean distance was used, as can be observed by equation (13).

$$d(x_i, x_j) = |f(x_i) - f(x_j)| \tag{13}$$

where $f(x_i)$ is the fitness equation for position $x_i$, and $d(x_i,x_j)$ represents the distance between two positions. And finally, a roulette method is used to calculate the possibility of selecting a location, as shown in equation (14).

$$p(x_i) = \frac{R(x_i)}{\sum_{j \in K} R(x_i)} \tag{14}$$

Thus, individuals with great distance from others have a higher chance of being selected, ensuring the diversity of the population.

The complexity of the FWO algorithm can be approximated by equation (15) [5], where $G$ is the number of generations, $n$ is the number of fireworks, $m$ is the number of sparks produced by the normal explosion strategy, and $m'$ is the number of sparks produced by the gaussian sparks:

$$O(G * (n + m + m')) \tag{15}$$

## III. PROPOSED METHODOLOGY

The proposed methodology involves many steps: (a) the selection of the dataset, (b) computing the electrical powers, (c) identification of events using the statistical tool developed and optimized, (d) selection of intervals of events occurred in the dataset to extract variations in electrical characteristics, (e) selection of events to be used in the training of the classifier, (f) training of the RF classifier using the FWO algorithm, and (g) construction of the confusion matrix. Figure 3 illustrates the schematic of the proposed methodology, divided into three stages.

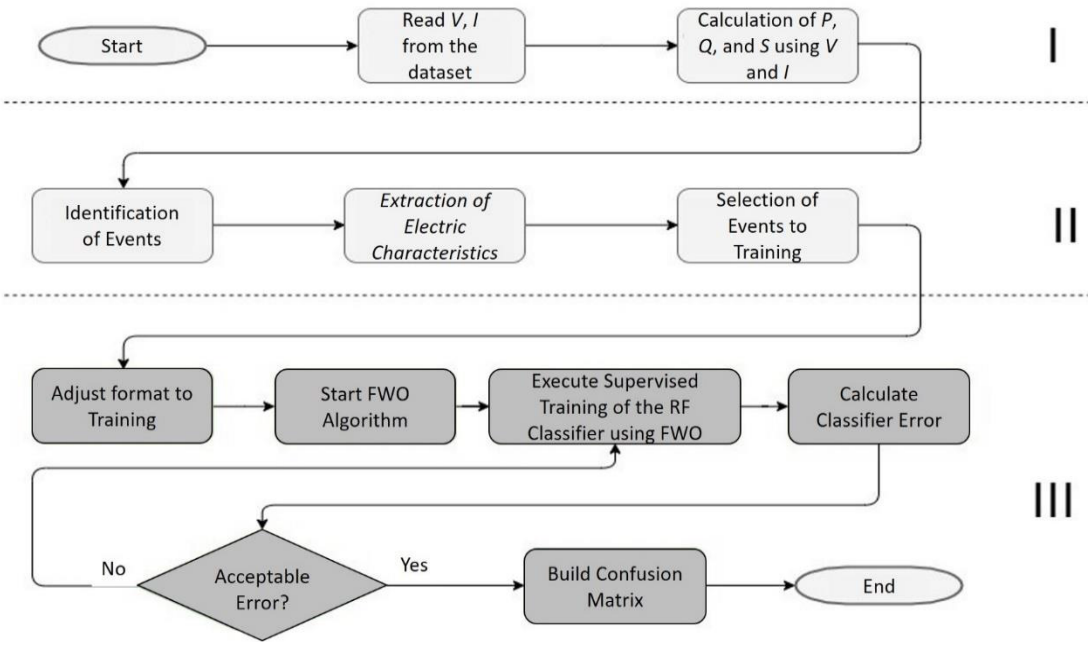


**FIGURE 3.** Flowchart of the proposed method.

### A. STAGE I – PRE-PROCESSING OF DATA

In this stage, the preprocessing of data from the dataset occurs. For example, if the BLUED dataset [24] has been used, each sample on the dataset contains information about the instant measurement ($t$), current in phase A ($I_A$), current in phase B ($I_B$), and voltage in phase A ($V_A$). The voltage and current data are used to calculate the powers, using the IEEE 1459/2010 standard [6].

#### 1) CALCULATION OF POWERS OF THE DATASET

For the calculations of the powers associated with phase B, it is necessary to calculate the voltage B ($V_B$), which is made using voltage A ($V_A$) as reference.

Data samples can be used in two different ways for power calculations: continuous or per cycle. In continuous form, $n$ samples are used to calculate the power in a period $t$, and in the calculation of power in the period $t + 1$ is removed the older sample and added a new one for the new calculation, as can be observed in equation (16).

$$\begin{aligned} p(t) &= [v(0) * i(0)] + [v(1) * i(1)] + \cdots \\ &\quad + [v(n) * i(n)] \\ p(t+1) &= [v(1) * i(1)] + \cdots + [v(n) * i(n)] \\ &\quad + [v(n+1) * i(n+1)] \end{aligned} \tag{16}$$

In the form by cycles, the calculation of power in the period $t$ uses $n$ samples, and in the calculation of power in the period $t+1$ are removed the $n$ samples and added new $n$ samples, as can be observed in equation (17).

$$\begin{aligned} p(t) &= [v(0) * i(0)] + [v(1) * i(1)] + \cdots \\ &\quad + [v(n) * i(n)] \\ p(t+1) &= [v(n+1) * i(n+1)] + [v(n+2) * i(n+2)] \\ &\quad + \cdots + [v(2n) * i(2n)] \end{aligned} \tag{17}$$

Both forms of calculation for any power at any time it is necessary at least one complete cycle of samples, so $n$ needs to be at least 200 (i.e., with acquisition frequency of 12 kHz).

For the calculations of the active and reactive powers, filter applications are required to eliminate frequencies other than the fundamental (60 Hz for this dataset). The Notch filter [47] is used to remove the non-desired frequencies from the original data, resulting in data containing only the fundamental frequency.

#### 2) MOVING AVERAGE FILTER

Applying a filter to the calculated powers is necessary to eliminate rapid variations in short time intervals, such as normal variations in signal and noise, which can make it difficult in the event identification process, presented below. The most suitable filter for this application is the moving average filter that smoothes the curve of the calculated powers, which can be expressed by equation (18).

$$P(i) = \frac{1}{N} \sum_{n=0}^{N} P(i+n) \tag{18}$$

The action of the filter can be observed in Figure 4, which presents: (a) the original signal and the result of different settings for the mobile filter (Figure 4 (b), (c), and (d)).

The filter parameter $N$ is optimized through the FWO to improve the performance of the identifier, aiming at minimizing the identifier error. The goal of optimization is to minimize the classification error, focused on calculating the accuracy of the system.

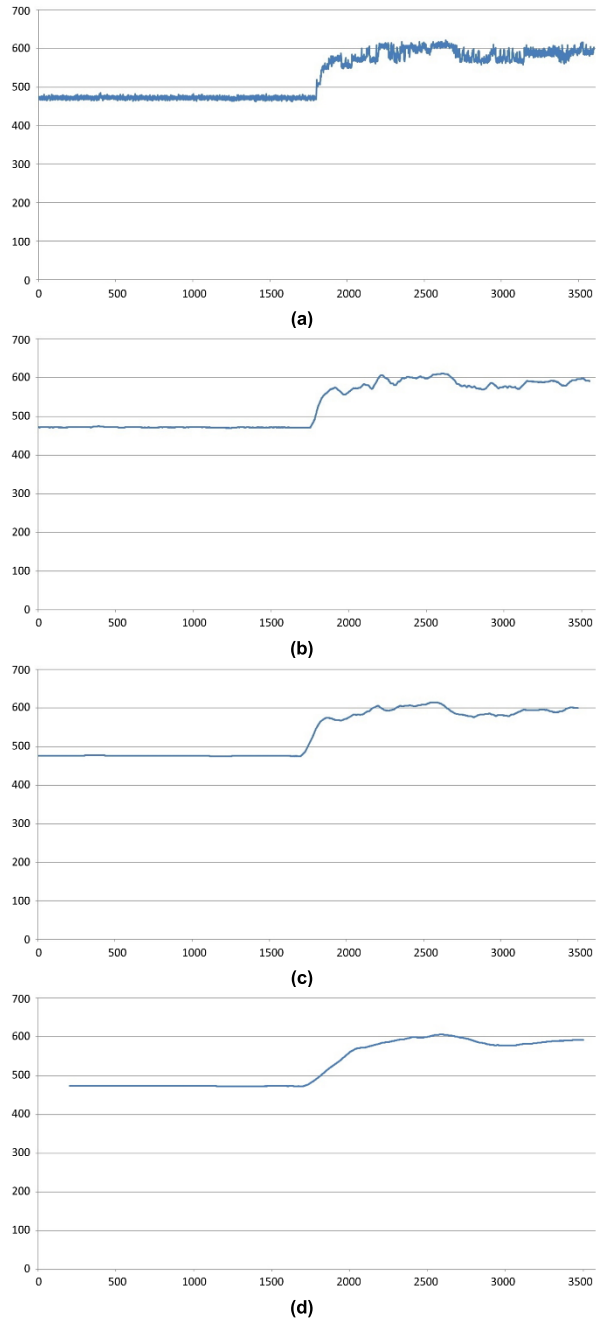


**FIGURE 4.** Application of the mobile average filter: (a) original signal, (b) filter with $N = 50$, (c) filter with $N = 100$, and (d) filter with $N = 300$.

### B. STAGE II – IDENTIFICATION OF THE OCCURRENCE OF EVENTS

For the function of identifying events developed, data analysis windows of a time series and statistical calculations are used but without the probabilistic voting system [48].

The principles used in the approach developed are based on the use of floating windows (moving from a fixed point), moving medium filters, and analysis of the variance of measurements between windows to detect the occurrence of events seeking stability states (before and after their performance). These principles are described below.

#### 1) STATISTICAL CALCULATIONS

For each dataset analyzed, the mean set needs to be calculated according to equation (19).

$$P' = \frac{1}{n}\sum_{1}^{n} P_i \tag{19}$$

where $P'$ is the arithmetic mean of the powers in the analyzed time interval, $P_i$ is i-th power of the set, and $n$ is the number of samples in the analyzed time interval.

Using the mean set analyzed, it is possible to calculate the variance of the values using equation (20).

$$\sigma^2 = \frac{1}{n}\sum_{1}^{n} (P' - P_i)^2 \tag{20}$$

where $\sigma^2$ is the variance of the measurements in the time interval analyzed.

#### 2) FLOATING WINDOWS

The proposal for identification by floating windows is initially based on monitoring the powers used to identify the loads. A power buffer is filled with $n$ sequential power samples (amount $n$ enough to detect an event). A pivot is used to reference a specific buffer point (for all experiments the center of the buffer was used), and the pre- and post-events windows of previously defined sizes ($w_{pre}$ and $w_{post}$) are allocated adjacent to the pivot initially. When a minimum power variation greater than a predefined limit ($P_{thr}$) is detected, as equation (21) shows, it is initially considered that an event occurred at that instant. Confirmation of this event is done by commuting the pre and post windows in order to distance themselves from the pivot to a maximum predefined distance ($dist_{max}$).

$$\left|P_{post} - P_{pre}\right| > P_{thr} \tag{21}$$

For each new position assumed by the windows, their means and variances are calculated. The point where each window (pre and post-event) has less variance is used to confirm whether it was possible to detect an event for the instant under analysis. Comparisons are made with the boundary variances ($\sigma^2_{limit}\,[i]$), sequentially, according to the equation (22).

$$\sigma^2_{limite}\,[0] < \sigma^2_{limite}\,[1] < \sigma^2_{limite}\,[2] < \sigma^2_{limite}\,[3] \tag{22}$$

To avoid the same event being indicated again as a new occurrence of the same event when pivot movement occurs. The pivot to the next position is defined as a pivot jump. So its new position would be such that the previous window ($w_{pre}$) was in the position of the post window ($w_{post}$) of the stability point of the detected event. This procedure can be observed in the example of Figure 5.

Figure 6 (a) exemplifies a sequence of powers calculated in time. In Figure 6(b), it is possible to visualize the beginning of the search process for stable regions, where the pivot is indicated by the purple line, and the pre- and post-windows are indicated by the space between the green lines, initially

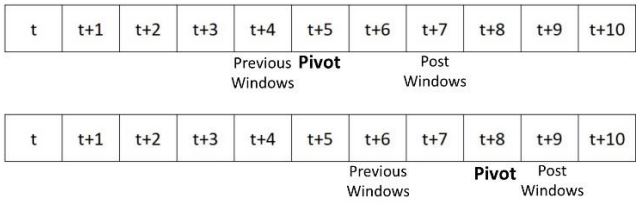


**FIGURE 5.** Pivot jumping process.

adjacent to each other. In Figure 6(c), it is possible to note the end of the process, where the windows moved away from the pivot until it reaches stable regions (low variances) at the terms.

The $\sigma^2_{thrpre}$ and $\sigma^2_{thrpost}$ parameters are optimized using the FWO algorithm, focusing on minimizing event identification error, and the parameters $w_{pre}$, $w_{post}$, and $dist_{max}$ are chosen experimentally.

The complexity of the algorithm is $O(N^2)$ because two main searches happen. The first search is in the vector with the samples instantly with a power variation greater than the $P_{thr}$. In the second first, within the first search, occurs for the zones of anterior and posterior stability.

#### 3) EXTRACTION OF EVENTS CHARACTERISTICS

During the process of identifying the occurrence of events, the windows used around the pivot contain the powers calculated for these moments. The difference between these powers provided the electrical characteristics added or removed with the activation or removal of any equipment to the monitored system. The classification process can be simplified if only absolute values of the electrical characteristics are used, reducing the order of the problem.

#### 4) SELECTION OF EVENTS FOR TRAINING

This part of the algorithm depends on the characteristic of each data set, such as the number of samples, types of samples, and so on. In the case of the use of the BLUED dataset [24], it has more than 35 monitored and demarcated circuits (with labels), but among them, 28 of these circuits represent a single load.

Labels greater than 204 indicate circuits that have more than one equipment connected to them or unknown equipment. Therefore, they cannot and are not used in the training phase, as they would introduce errors in training in the classification algorithm since the electrical characteristics and their labels would not reflect a specific load.

Other events that are not selected for training are those of events that occur simultaneously, since electrical characteristics may be masked due to overlapping events.

### C. STAGE III – TRAINING AND CLASSIFICATION

This stage consists of selecting the format for training, the training itself, and optimization of the RF classifier, and classifier performance analysis.

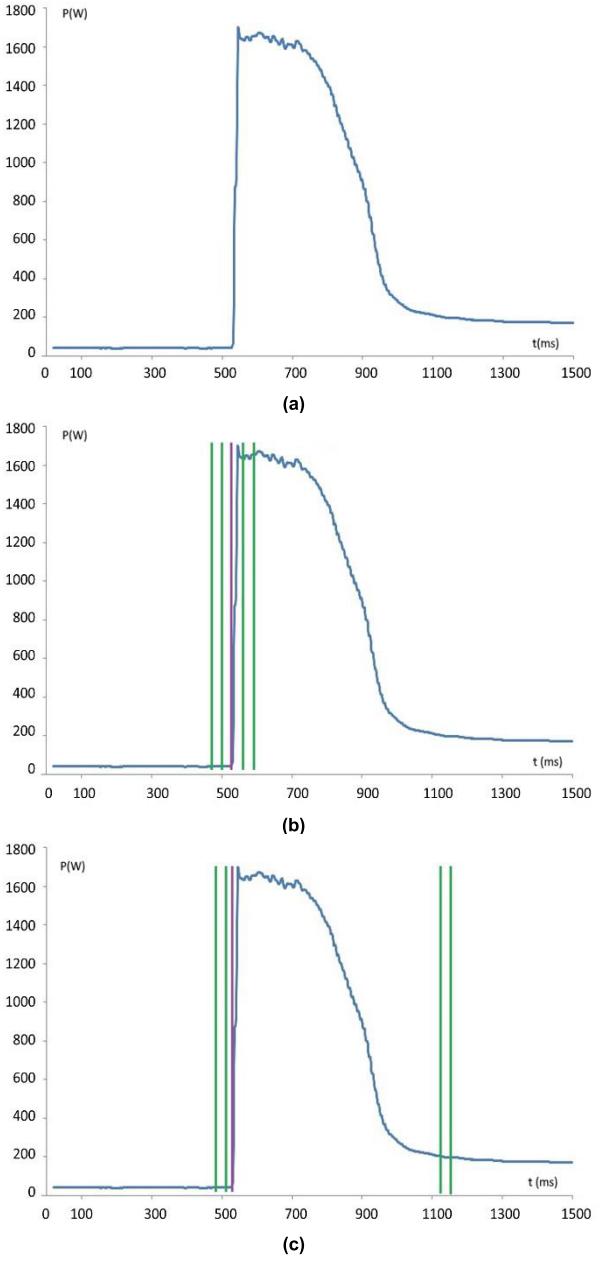


**FIGURE 6.** Floating windows: (a) power in time, (b) initial previous and post windows, and (c) final previous and post windows.

#### 1) FORMAT SELECTION FOR TRAINING

The organization of the data impacts the performance of the classifier. Four different ways to data organize used in the proposed method are a variation of electrical characteristics (VCE), simple waveform (SWF), offset waveform (OWF), and wave variation rate (WVR).

The variation of electrical characteristics is based on the use of the variation of each of the electrical characteristics before and after the event. For the purpose of decreasing the order of complexity of the problem, the original signals are disregarded, only the variations are used. Variations of characteristics are arranged, as shown in Figure 7. An example of this process can be observed in Figure 8.

| $\Delta I$ | $\Delta V$ | $\Delta I_1$ | $\Delta V_1$ | $\Delta P$ | $\Delta Q_1$ | $\Delta S$ | $\Delta S_1$ | $\Delta S_n$ | Class |
|---|---|---|---|---|---|---|---|---|---|

**FIGURE 7.** Data layout in the training table.

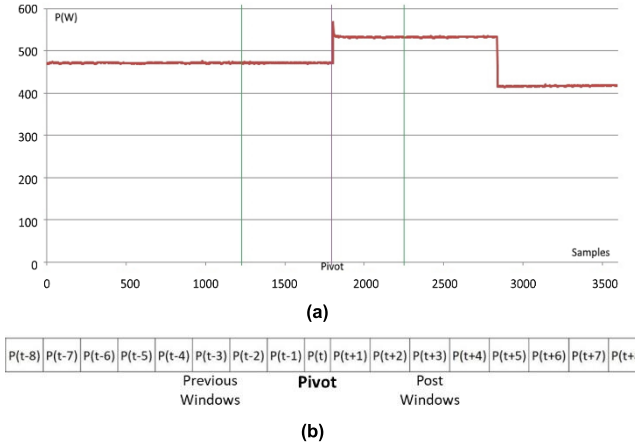


**FIGURE 8.** Example of data arrangement using the variation of the characteristics: (a) Graphic with power in time, showing the pivot, and (b) pre- and post-windows.

| $I(i\rightarrow -n)$ | $V(i\rightarrow -n)$ | $I_1(i\rightarrow -n)$ | $V_1(i\rightarrow -n)$ | $P(i\rightarrow -n)$ | $Q_1(i\rightarrow -n)$ | $S(i\rightarrow -n)$ | $S_1(i\rightarrow -n)$ | $S_n(i\rightarrow -n)$ |
|---|---|---|---|---|---|---|---|---|

**FIGURE 9.** Data layout in a table for training with a simple waveform.

For the other modes of data layouts, a time interval is selected around the pivot. The data are arranged sequentially, as demonstrated in Figure 9, where $i$ is the beginning, and $n$ is the end of the analyzed interval.

In the simple waveform, no treatment is required in the data used. But, in the offset waveform, the wave is shifted vertically based on this displacement at the point where the pivot is, moving it to the zero points, as can be seen in Figure 10.

At the wave variation rate, a time interval is selected around the pivot, and the variation rate is calculated in this range, as can be observed in Figure 11.

### 2) CLASSIFIER OPTIMIZATION USING FWO

Classifier optimization is performed offline, optimizing parameters of direct performance influence, which are: trees, maximum tree depth, feature per split, and sub-sample ratio.

The optimization algorithm used is the previously presented FWO algorithm. The objective of the optimization is to minimize the classification error, focusing on calculating the accuracy of the system.

The performances of the classifier optimizations are exposed through the confusion matrix. The matrix demonstrates the results by grouping the classifier outputs according to the following concepts [49]: (a) True Positive ($TP$) - event classified and corresponds to reality; (b) True Negative ($TN$) - unclassified event corresponds to non-existence with the real; (c)· False Positive ($FP$) - event classified, but erroneously with reality; and (d) False Negative ($FN$) - an event not

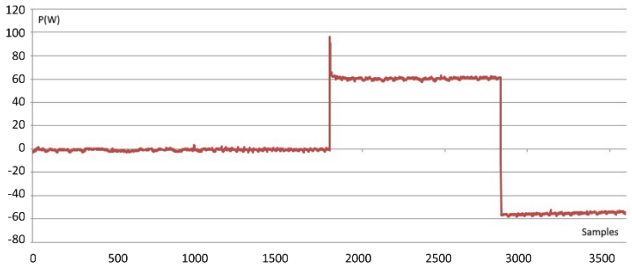


**FIGURE 10.** Example of data provision using offset waveform.

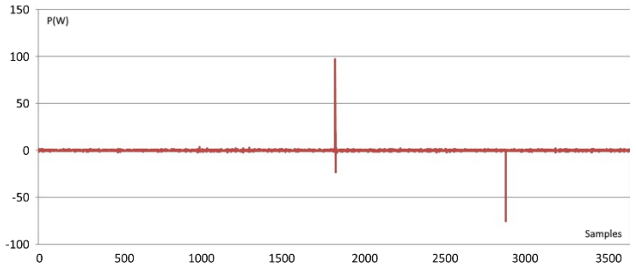


**FIGURE 11.** Example of data arrangement using the rate of variation of curves.

classified, but existing in reality. With these concepts, it is possible to calculate metrics to indicate the performance of the classifier, which are:

1. Accuracy – which calculates the proportion of correct classifications, without separating whether it is a true positive or true negative, in relation to all data, as shown in equation (23).

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{23}$$

2. Recall- which calculates the proportion of true positives in relation to all positives, as shown in equation (24). It also assumes the percentage name of true positives ($TPP$). It demonstrates the system's ability to classify the occurrence of a given event correctly.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{24}$$

3. Specificity – which calculates the proportion of true negatives in relation to all negatives, as shown in equation (25). It demonstrates the system's ability to sort the absence of events correctly.

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{25}$$

4. Precision – which calculates the proportion of correct positive in relation to the total number of positive events, as shown in equation (26).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{26}$$

5. Efficiency – which calculates the arithmetic mean of Sensitivity and Specificity, as shown in equation (27).

$$\text{Efficiency} = \frac{\text{Recall} + \text{Specificity}}{2} \tag{27}$$

6. False Positive Percentage ($FPP$) – which calculates the percentage of false true stemming from the total amount of negative events, as shown in equation (28).

$$FPP = \frac{FP}{TN + FP} \tag{28}$$

7. F-Score - also known as F1-Score, which calculates the harmonic mean of precision and sensitivity, as shown in equation (29).

$$\text{F} - \text{Score} = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{29}$$

The system error is also calculated with the Euclidean distance, from the optimal point til the point formed by ($FPP$, $TPP$), as can be observed in equation (30) [20]. The optimal point is given by (0, 1) because there are 0% false positives and 100% true positives. The lower the result, the closer to the optimal point; consequently, the better the identifier performance.

$$\text{Error} = \|(0, 1) - (FPP, TPP)\| \tag{30}$$

Or alternatively, the equation (31) can calculate the performance.

$$\psi = 1 - \text{Error} = 1 - \|(0, 1) - (FPP, TPP)\| \tag{31}$$

## IV. RESULTS OF TESTS

The evaluation of the proposed methodology and performance comparison is carried out in the following ways: 1. Optimization of parameters for identifying event occurrence using a portion of the BLUED dataset [24]; 2. Validation of optimized parameters in the full dataset; 3. Analyze the possible sources of error of the identification process; 4. Evaluation of which data format provides the best results for classification; 5. Optimization of the classification method through the RF algorithm; 6. Evaluation of the performance of the method with optimized parameters; and 7. Analyze the possible sources of error in the classification process.

The results of the main objectives proposed, identification of events, and classification of loads, are presented in the following subsections.

### A. IDENTIFICATION PROCESS

Because the dataset is very extensive, the optimization process has been performed in a segment of it, randomly chosen.

The parameters of the best individual in optimization are: $N_A = 50$; $\sigma^2_{thrpreA} = 0.01$; $\sigma^2_{thrpostA} = 0.0033$; $N_B = 200$; $\sigma^2_{thrpreB} = 0.01$; $\sigma^2_{thrpostB} = 0.0033$. As defined by equation (22) the limits of variance ($\sigma^2_{limit}$) is based on optimized value, assuming the following values: $\sigma^2_{limit}$ [?] $= \sigma^2_{thr}$ ; $\sigma^2_{limit}$ [1] $= 3 * \sigma^2_{limit}$ [?]; $\sigma^2_{limit}$ [2] $= 6 * \sigma^2_{limit}$ [?]; and $\sigma^2_{limit}$ [3] $= 12 * \sigma^2_{limit}$ [?].

**TABLE 1. Confusion matrix for the identification process.**

| Phases | A | | B | | A+B | |
|---|---|---|---|---|---|---|
| | Det | N-Det | Det | N-Det | Det | N-Det |
| Ev | 830 | 62 | 1251 | 312 | 2081 | 374 |
| N-Ev | 82 | 19268 | 763 | 18597 | 845 | 37865 |

Note: Ev- Event, N-Ev - Non-Event, Det- Detected, and N-Det -Non-Detected.

**TABLE 2. Performance indexes of the identification process.**

| Phases | A | B | A+B |
|---|---|---|---|
| Accuracy | 99.29% | 98.22% | 98.01% |
| Recall | 93.05% | 62.12% | 71.12% |
| Specificity | 99.58% | 99.46% | 99.36% |
| Efficiency | 96.31% | 80.79% | 85.24% |

**TABLE 3. Performance index comparison of the identification processes.**

| Method | Proposed | GLR | Hybrid |
|---|---|---|---|
| Phase A | 88% | 78% | 94% |
| Phase B | 47% | 49% | - |

**TABLE 4. Performances of the event detection.**

| Methods | **VCE** | SWF | **OWF** | WVR |
|---|---|---|---|---|
| Recall | 89.91% | 77.18% | 87.84% | 60.92% |
| Specificity | 99.04% | 96.94% | 98.63% | 92.94% |
| Precision | 89.60% | 75.18% | 86.55% | 58.10% |
| F-Score | 89.65% | 75.36% | 86.73% | 53.73% |
| Efficiency | 94.47% | 87.06% | 93.24% | 76.93% |

Table 1 represents the confusion matrix of the identification process for the phases A, B, and the two phases together (A + B). The total event occurrence (Event) in both phases is 2455 ($TP + FN$). The total event non-occurrence (Non-Event) is approximated from the total monitoring time, taking into account the event with a longer transition time. The number of non-events is 37865. Table 2 shows the performance indexes.

Performance comparison of the proposed method is made with the GLR identifier [48] and with the hybrid identifier [20], using $TPP$ and $FPP$ values, and the performance index (Eq. (31)) Table 3 shows these results. The proposed identifier performs higher in the identification in phase A and slightly lower performance in phase B when compared to the GLR identifier. However, when compared to the hybrid identifier presents lower performance relative to phase A.

### B. EVENT DETECTION

The computation of all methods are made, and Table 4 summarizes the main results. Two methods with better performance are VCE and OWF, respectively, when analyzed from

**TABLE 5.** Random forest final characteristics.

| Methods | Limits | VCE+FWO | OWF+FWO | VCE+PSO |
|---|---|---|---|---|
| Trees | 3000 | **557** | 996 | 704 |
| MaxTreeDep | 100 | **23** | 32 | 25 |
| FeatPerSplit | - | **3** | 22 | 4 |
| S-sampleRat | 100% | 79% | 100% | 99% |

Note MaxTreeDep - Maximum Trees Depth, FeatPerSplit - Feature Per Split, and S-sampleRat - Sub Sample Ratio.

**TABLE 6.** Performances of the load identification process.

| Methods | VCE+FWO | OWF+FWO |
|---|---|---|
| Accuracy | 98.37% | 88.53% |
| Recall | 89.57% | 88.53% |
| Specificity | 98.96% | 98.69% |
| Precision | 89.33% | 87.19% |
| F-Score | 89.28% | 87.37% |
| Efficiency | 94.26% | 93.61% |

**TABLE 7.** Performance index comparison of the identification processes.

| | *TPP* | *FPP* | $\psi$ |
|---|---|---|---|
| VCE+FWO | 89.57 | 1.04 | 98,06 |
| OWF+FWO | 88.53 | 1.31 | 97.75 |

the final values of indexes. These two approaches are used to continue the load classification process.

### C. CLASSIFICATION OF LOADS

For the stage of optimization and classification of loads, VCE, and OWF approaches are used. Table 5 shows some information about the ranges of RF optimization parameters, the final results of the RF using VCE and OWF data and being optimized by FWO, and the result comparison using particle swarm optimization (PSO) [45]. While Table 6 presents the performance indexes for each method, and Table 7 shows the results of the load identification process of each approach. The comparison among the methods shows that the performances are slightly the same; however, the Random Forest (RF) produced by the variation of electrical characteristics (VCE) data preparation, and training with Fireworks Optimization (FWO) algorithm, provides much more single forest (highlights in Table 5), producing a more simple model to be used. Because the number of trees is smaller than the other approaches, and more, these trees are smaller in deep and in the feature per split, producing simpler trees. It means the model created by the VCE + FWO has less and simpler trees with a better representation index ($\psi$). This index, computed by the equation (31), represents the quality of the performance of the proposed approach to classifying the loads correctly. The value of 98.06% indicates the percentage of correct classifications provided by the proposed approach.

## V. CONCLUSION

This paper presents a non-intrusive method for load identification based on Random Forest (RF) and Firework Optimization (FWO). The main idea behind the proposed method is to use the flexibility of the RF to build models without any necessity of input attribute selection and the facility to handle a large set of data. The time consumption in the training process is small when compared with other models treating a large amount of data.

The RF parameters are optimized by an FWO algorithm, which setting its parameters for a better performance according to a cost fitness function. Also, FWO has been chosen because the number of adjusting parameters is small, and its values are practically known.

The proposed method is divided into three main integrated stages: pre-processing data, identification of the event occurrence, and training and classification. The first stage prepares the data from the dataset to be used for an event identifier. The second stage detects the occurrence of events (changes of the loads), while in the third stage, the RF is training an RF using the FWO algorithm to classify these events.

The proposed method is applied in a public dataset named BLUED, which contains voltage and current data from a specific electric network. This dataset has been used to provide a common base to generate comparisons among methods. Among the event identification approaches of the proposed method, two has a better performance in this dataset: the variation of electrical characteristics (VCE), and the offset waveform (OWF). The performance of these two identifiers is compared with the other two approaches GLR and hybrid, available in the literature, which used the same dataset, with better results.

The test continues using VCE and OWF outputs to train RF with the FWO algorithm. The final performance of these two approaches is slightly the same; however, the produced RF produced by VCE data is much more sample and simpler, with a smaller number of trees and reduced depth of the trees. Finally, this best-proposed approach is compared with an RF training with a particle swarm optimization (PSO) algorithm, using VCE data. The comparison between FWO and PSO to training RF also produces slightly the same results of classification; however, the RF with FWO produces a much simpler tree model, in a number of trees and in its depth, showing the quality of the proposed method.

It is important to notice that the other identifier proposed approaches, in special the wave variation rate (WVR), can also be useful. It depends on the characteristics of the dataset used.

## ACKNOWLEDGMENT

The authors would like to thank the Coordination for the Improvement of Higher Education Personnel (CAPES) for supporting this project.

## REFERENCES


[1] Z. Ma, F. Liu, C. Shen, Z. Wang, and S. Mei, "Fast searching strategy for critical cascading paths toward blackouts," *IEEE Access*, vol. 6, pp. 36874–36886, 2018.

[2] L. C. Siebert, A. R. Aoki, T. S. P. Fernandes, and G. Lambert-Torres, "Customer targeting optimization system for price-based demand response programs," *Int. Trans. Electr. Energy Syst.*, vol. 29, no. 2, p. e2709, Feb. 2019.

[3] S. C. Ferreira, R. B. Gonzatti, R. R. Pereira, C. H. da Silva, L. E. B. da Silva, and G. Lambert-Torres, "Finite control set model predictive control for dynamic reactive power compensation with hybrid active power filters," *IEEE Trans. Ind. Electron.*, vol. 65, no. 3, pp. 2608–2617, Mar. 2018.

[4] T. K. Ho, "Random Decision Forests," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, Montreal, QC, Canada, 1995, pp. 278–282.

[5] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," in *Proc. Int. Conf. Swarm Intell.*, Sanya, China, 2010, pp. 355–364.

[6] *Definitions for the Measurement of Electric Power Quantities under Sinusoidal, Nonsinusoidal, Balanced, or Unbalanced Conditions*, Standard 1459-2010, 2010.

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

[8] J. G. Roos, I. E. Lane, E. C. Botha, and G. P. Hancke, "Using neural networks for non-intrusive monitoring of industrial electrical loads," in *Proc. 94. Adv. Technol. IEEE Instrum. Meas. Technolgy Conf.*, Jul. 1991, pp. 1115–1118.

[9] V. Namboodiri, V. Aravinthan, S. N. Mohapatra, B. Karimi, and W. Jewell, "Toward a secure wireless-based home area network for metering in smart grids," *IEEE Syst. J.*, vol. 8, no. 2, pp. 509–520, Jun. 2014.

[10] K. Balasubramanian and A. Cellatoglu, "Improvements in home automation strategies for designing apparatus for efficient smart home," *IEEE Trans. Consum. Electron.*, vol. 54, no. 4, pp. 1681–1687, Nov. 2008.

[11] A. C. Jose and R. Malekian, "Improving smart home security: Integrating logical sensing into smart home," *IEEE Sensors J.*, vol. 17, no. 13, pp. 4269–4286, Jul. 2017.

[12] J. Liang, S. K. K. Ng, G. Kendall, and J. W. M. Cheng, "Load signature study—Part I: Basic concept, structure, and methodology," *IEEE Trans. Power Del.*, vol. 25, no. 2, pp. 551–560, Apr. 2010.

[13] T. Kato, H. S. Cho, D. Lee, T. Toyomura, and T. Yamazaki, "Appliance recognition from electric current signals for information-energy integrated network in home environments," in *Proc. 7th Int. Conf. Smart Homes Health Telematics*, Tours, France, 2009, pp. 150–157.

[14] H. Lam, G. Fung, and W. Lee, "A novel method to construct taxonomy electrical appliances based on load signaturesof," *IEEE Trans. Consum. Electron.*, vol. 53, no. 2, pp. 653–660, 2007.

[15] M. Ito, R. Uda, S. Ichimura, K. Tago, T. Hoshi, and Y. Matsushita, "A method of appliance detection based on features of power waveform," in *Proc. Can. Conf. Electr. Comput. Eng. Toward Caring Humane Technol.*, 1990, pp. 291–294.

[16] A. Reinhardt, D. Burkhardt, M. Zaheer, and R. Steinmetz, "Electric appliance classification based on distributed high resolution current sensing," in *Proc. 37th Annu. IEEE Conf. Local Comput. Netw. Workshops*, Oct. 2012, pp. 999–1005.

[17] P. Meehan, S. Phelan, C. McArdle, and S. Daniels, "Temporal and frequency analysis of power signatures for common household appliances," in *Proc. Symp. ICT Energy Efficiency Workshop Inf. Theory Secur. (CIICT)*, 2012, pp. 22–27.

[18] A. A. Zaidi, F. Kupzog, T. Zia, and P. Palensky, "Load recognition for automated demand response in microgrids," in *Proc. 36th Conf. IEEE Ind. Electr.*, Glendale, AZ, USa, Aug. 2010, pp. 2442–2447.

[19] Y.-H. Lin and M.-S. Tsai, "Non-intrusive load monitoring based on swarm intelligence," in *Proc. 8th Int. Congr. Adv. Appl. Informat. (IIAI-AAI)*, Jul. 2019, pp. 546–551.

[20] M. Lu and Z. Li, "A hybrid event detection approach for non-intrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 528–540, Jan. 2020.

[21] 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. Consumer Electron.*, early access, Mar. 3, 2020, doi: 10.1109/TCE.2020.2977964.

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

[23] J. Z. Kolter and M. J. Johnson, "Redd: A public data set for energy disaggregation research," in *Proc. Workshop Data Mining Appl. Sustainability*, San Diego, CA, USA, 2011, pp. 59–62.

[24] K. Anderson, A. F. Ocneanu, D. Benítez, 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 Works. Data Mining Appl. Sustainability*, Beijing, China, 2012, pp. 1–5.

[25] J. Kelly and W. Knottenbelt, "The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes," *Sci. Data*, vol. 2, no. 1, Dec. 2015, Art. no. 150007.

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

[27] M. Figueiredo, A. de Almeida, and B. Ribeiro, "Home electrical signal disaggregation for non-intrusive load monitoring (NILM) systems," *Neurocomputing*, vol. 96, pp. 66–73, Nov. 2012.

[28] M. Figueiredo, A. D. Almeida, and B. Ribeiro, "An experimental study on electrical signature identification of non-intrusive load monitoring (nilm) systems," in *Proc. Int. Conf. Adapt. Natural Comp. Alg.*, Ljubljana, Slovenia, 2011, pp. 31–40.

[29] Y.-H. Lin, M.-S. Tsai, and C.-S. Chen, "Applications of fuzzy classification with fuzzy c-means clustering and optimization strategies for load identification in NILM systems," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jun. 2011, pp. 859–866.

[30] J. Pearl and R. E. Korf, "Search Techniques," *Ann. Rev. Comput. Sci.*, vol. 2, pp. 451–467, Jul. 1987.

[31] V. Kumar, "Algorithms for Constraint-Satisfaction Problems: A Survey," *AI Mag.*, vol. 13, pp. 33–44, Apr. 1992.

[32] M. Belgiu and L. Dragut, "Random forest in remote sensing: A review of applications and future directions," *ISPRS J. Photogramm. Remote Sens.*, vol. 114, pp. 24–31, Apr. 2016.

[33] V. F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. P. Rigol-Sanchez, "An assessment of the effectiveness of a random forest classifier for land-cover classification," *ISPRS J. Photogramm. Remote Sens.*, vol. 67, pp. 93–104, Jan. 2012.

[34] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Rio de Janeiro, Brazil, Aug. 2007, pp. 1–8.

[35] Y. Qi, "Random forest for bioinformatics," in *Ensemble Machine Learning*. C. Zhang and Y. Ma, eds. Boston, MA, USA: Springer, 2012.

[36] T. Kam Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, 1998.

[37] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural Comput.*, vol. 9, no. 7, pp. 1545–1588, Oct. 1997.

[38] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, Aug. 2001.

[39] G. James, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*. New York, NY, USA: Springer, 2013.

[40] *Single-Objective vs. Multiobjective Optimisation for Integrated Decision Support*, SAVIC, Lleida, Spain, 2002.

[41] G. Beni, "The concept of cellular robotic system," in *Proc. IEEE Int. Symp. Intell. Control*, Arlington, VA, USA, 1988, pp. 57–62.

[42] J. M. Bishop, "Stochastic searching networks," in *Proc. 1st IEE Int. Conf. Art. Neur. Netw.*, London, U.K., 1989, pp. 329–331.

[43] M. Dorigo and G. Di Caro, "Ant colony optimization: A new meta-heuristic," in *Proc. Congr. Evol. Comput.*, Washington, DC, USA, 1999, pp. 1470–1477.

[44] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neur. Netw.*, Perth, WA, Australia, 1995, pp. 1942–1948.

[45] A. A. A. Esmin, G. Lambert-Torres, and A. C. Zambroni de Souza, "A hybrid particle swarm optimization applied to loss power minimization," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 859–866, May 2005.

[46] Y. Tan, C. Yu, S. Zheng, and K. Ding, "Introduction to fireworks algorithm," *Int. J. Swarm Intell. Res.*, vol. 4, no. 4, pp. 39–70, Oct. 2013.

[47] S. C. Ferreira, R. B. Gonzatti, C. H. Silva, L. E. B. da Silva, R. R. Pereira, and G. Lambert-Torres, "Adaptive real-time power measurement based on IEEE standard 1459-2010," *Electric Power Compon. Syst.*, vol. 43, no. 11, pp. 1307–1317, Jul. 2015.

[48] K. D. Anderson, M. E. Berges, A. Ocneanu, D. Benitez, and J. M. F. Moura, "Event detection for non intrusive load monitoring," in *Proc. 38th Annu. Conf. IEEE Ind. Electron. Soc.*, Oct. 2012, pp. 3312–3317.

[49] R. Kohavi and F. Provost, "Glossary of terms," *Mach. Learn.*, vol. 30, pp. 271–274, Jun. 1998.

**PAULO RICARDO ZAMBELLI TAVEIRA** received the B.S. degree in automation and control engineering from Itajuba Federal University, Brazil, in 2011, where he is currently pursuing the M.Sc. degree in computer science. He worked as a Field Engineer in Kuwait and other countries. He has experience in the instrumentation area with an emphasis on collecting, storing, and processing geological data. His research interests include data management and non-intrusive data monitoring.

**CARLOS HENRIQUE VALÉRIO DE MORAES** received the B.S., M.Sc., and Ph.D. degrees in electrical engineering from Itajuba Federal University (UNIFEI), Itajuba, Brazil, in 2001, 2002, and 2006, respectively. He has been an Associate Professor with UNIFEI, where he is currently the Information Technology Director. He teaches the course in control engineering and automation in the areas of programming, programming in embedded systems, digital electronics, power electronics, logic circuits, linear circuits, microcontrollers, and artificial intelligence. His research interests include intelligent control, computer vision, intelligent embedded systems, signal processing, autonomous navigation, and robotics.

**GERMANO LAMBERT-TORRES** (Fellow, IEEE) received the B.S. and M.Sc. degrees in electrical engineering from the Federal University of Itajuba (UNIFEI), Itajuba, Brazil, the B.S. degree in economics from the South Minas Gerais Economic and Social Sciences Faculty, Itajuba, the B.S. degree in mathematics from the Itajuba Faculty of Sciences and Languages, Itajuba, and the Ph.D. degree in electrical engineering from the École Polytechnique de Montreal, Montreal, QC, Canada, in 1990.

From 1983 to 2012, he was a Professor with the Electrical Engineering Department, UNIFEI. From 2000 to 2004, he was the Prorector of Research and Graduate Studies at UNIFEI. He also serves a member of two high committees in Brazil, one for Education: a member of National Final Exam for undergraduate electrical engineers, from 1998 to 2003, and other for Research: Member of National Council for Research (CA-EE CNPq), from 2004 to 2007. From 1995 to 1996, he was a Visiting Professor with the University of Waterloo, Waterloo, ON, Canada. He is currently the Director of Research and Development at PS Solutions, Brazil, and the Chair of the Scientific-Technical Council with the Gnarus Institute, Brazil. He has taught numerous IEEE and IFAC tutorials in the U.S., Europe, and Asia. He has completed more than 80 M.Sc. and Ph.D. thesis supervisions and published more than 600 journal and technical conference papers. He is also the author/editor or coauthor of nine books, more than 30 book chapters, and 80 transactions articles on intelligent systems and nonclassical logic.

Dr. Lambert-Torres is a member of the International Conference on Intelligent Systems Applications to Power Systems (ISAP) international board. He is also a member of the IEEE Fellow Committee in the Industry Application Society (IEEE IAS) and the Education Society (IEEE EdSoc). He is also a member of the IEEE Medal in Power Engineering Committee. He also serves as a Consultant for many power industries in Brazil and South America, with more than 150 Research and Development developed projects. He is also a member of several committees in Brazilian Governmental and Regulatory Agencies, such as The Ministry of Education (MEC), The Ministry of Science and Technology (MCT), The National Electric Energy Agency (ANEEL), and The National Petroleum Agency (ANP). He serves on several committees related to intelligent systems, including the IEEE and the International Council on Large Electric Systems (CIGRÉ). He is also a Fellow of the IEEE, Class 2014, in the Engineer/Scientist category, with the following citation: for contributions to the application of intelligent systems to power systems. He was a recipient of several awards, including the Technical Committee Working Group Recognition Award on New Technologies and Practical Applications from the IEEE Power and Energy Society (PES), in 2006, the Outstanding Leadership as Member of the ISAP Board of Directors from the International Council of ISAP, in 2007, and the Technical Committee Working Group Recognition Award on Multiagent Systems from the IEEE-PES, in 2008. He has served as the General Chair for ISAP, in 1999 and 2009, and the Vice-General Chair for ISAP, in 2001, and the Congress on Logic Applied to Technology, in 2003 and 2007.