sensors

MDPI

Article

Load Recognition in Home Energy Management Systems Based on Neighborhood Components Analysis and Regularized Extreme Learning Machine

Thales W. Cabral [1], Fernando B. Neto [2], Eduardo R. de Lima [3], Gustavo Fraidenraich [1] and Luís G. P. Meloni [1,*]

[1] Department of Communications, School of Electrical and Computer Engineering, University of Campinas, Campinas 13083-852, Brazil; t264377@dac.unicamp.br (T.W.C.); gfraiden@unicamp.br (G.F.)
[2] Companhia Paranaense de Energia, Curitiba 81200-240, Brazil; fernando.bauer@copel.com
[3] Department of Hardware Design, Instituto de Pesquisa Eldorado, Campinas 13083-898, Brazil; eduardo.lima@eldorado.org.br
* Correspondence: meloni@unicamp.br; Tel.: +55-19-3521-3830 (ext. 3818)

Abstract: Efficient energy management in residential environments is a constant challenge, in which Home Energy Management Systems (HEMS) play an essential role in optimizing consumption. Load recognition allows the identification of active appliances, providing robustness to the HEMS. The precise identification of household appliances is an area not completely explored. Gaps like improving classification performance through techniques dedicated to separability between classes and models that achieve enhanced reliability remain open. This work improves several aspects of load recognition in HEMS applications. In this research, we adopt Neighborhood Component Analysis (NCA) to extract relevant characteristics from the data, seeking the separability between classes. We also employ the Regularized Extreme Learning Machine (RELM) to identify household appliances. This pioneering approach achieves performance improvements, presenting higher accuracy and weighted F1-Score values—97.24% and 97.14%, respectively—surpassing state-of-the-art methods and enhanced reliability according to the Kappa index, i.e., 0.9388, outperforming competing classifiers. Such evidence highlights the promising potential of Machine Learning (ML) techniques, specifically NCA and RELM, to contribute to load recognition and energy management in residential environments.

Keywords: machine learning; household appliances; active power; appliance recognition

check for updates

Citation: Cabral, T.W.; Neto, F.B.; de Lima, E.R.; Fraidenraich, G.; Meloni, L.G.P. Load Recognition in Home Energy Management Systems Based on Neighborhood Components Analysis and Regularized Extreme Learning Machine. *Sensors* **2024**, *24*, 2274. https://doi.org/10.3390/s24072274

Academic Editor: Guido De Angelis

Received: 28 February 2024
Revised: 22 March 2024
Accepted: 31 March 2024
Published: 2 April 2024

1. Introduction

The rising demand for electrical energy presents a challenge to sustainable consumption, affecting various sectors. According to Kim et al. [1], considering diverse sources like biomass and natural gas, the residential sector contributes 27% to global consumption. Moreover, as per Rashid et al. [2] and Bang et al. [3], due to malfunctioning appliances and improper consumption habits, 30% of energy is wasted. One way to contribute to sustainable consumption and minimize such issues is to adopt Home Energy Management Systems (HEMS).

HEMS refers to technologies developed to manage the electricity consumption in households or commercial buildings. According to Motta et al. [4], a HEMS architecture consists of a controller and smart outlets to connect household appliances to the electrical grid. The electricity demand of a household may exhibit seasonal variations, influenced by the type of appliances in operation, such as heaters and air conditioning devices. However, HEMS is capable of monitoring such appliance activities, incorporating additional functionalities such as load disaggregation techniques presented by Lemes et al. [5], methods for anomaly detection in appliances as per Lemes et al. [6], and load recognition mechanisms as reported by Cabral et al. [7].

Load recognition means identifying appliances in operation and emerges as an essential building block for advancing home energy management systems. Load recognition also contributes to load disaggregation methods, allowing specific identification of devices after the disaggregation procedure. Furthermore, load recognition is fundamental to building robust appliance databases by analyzing electrical signals, making this process more precise and automatic. In the diversified domestic environment, where various appliances such as microwaves, dishwashers, air conditioning, freezers, and heaters operate, HEMS' capability to determine which appliance is operating is indispensable. This functionality takes on a relevant practical dimension when replacing appliances connected to the smart outlets. In this case, the HEMS automatically identifies the new appliance in operation with the load recognition system. The same practical importance can be observed for automated database construction.

Currently, there is a trend towards using Machine Learning (ML) in state-of-the-art solutions for load recognition. Some works employ models considered computationally costly, such as Mian Qaisar and Alsharif [8] with the Artificial Neural Network (ANN), De Baets et al. [9], Faustine and Pereira [10], and Matindife et al. [11] with the Convolutional Neural Network (CNN), and Huang et al. [12] and Heo et al. [13] with the Long Short-Term Memory (LSTM). Other studies apply architectures with lower computational costs, such as Qaisar and Alsharif [14] with k-Nearest Neighbors (k-NN), Soe and Belleudy [15] with Classification and Regression Trees (CART), Qaisar and Alsharif [14] and Soe and Belleudy [15] with Support Vector Machine (SVM), and Zhiren et al. [16] with the standard Extreme Learning Machine (ELM). Furthermore, reducing the amount of information needed to identify the appliance in operation is a pertinent feature of the solutions. According to Cabral et al. [7] and Soe and Belleudy [15], it is feasible to achieve this using ML techniques for feature extraction, such as Linear Discriminant Analysis (LDA), Principal Component Analysis (PCA), and others. Conversely, we can limit the amount of information required regarding household appliances by solely using a single type of electrical signal, such as voltage, current, reactive power, or active power.

Designing an approach that guarantees high performance, reliability, and the short training time of ML models using only a single type of electrical signal represents a crucial challenge in the load recognition area. Several approaches are seeking to achieve these objectives. Presently, the modern methods for load recognition use images generated from electrical signals such as voltage, current, active power, and others. The advantages of this approach are substantiated by the works as Faustine and Pereira [10], Matindife et al. [11], Gao et al. [17], De Baets et al. [9], and Cabral et al. [7]. In this regard, the motivation of this study is to propose enhancements to the load recognition system in HEMS. To achieve this, the current study utilizes images generated exclusively from the active power data of household appliances. The present work contributes to advancements in the load recognition area, introducing novel applications not previously explored in the literature. Our analysis investigates the underlying factors, highlighting Neighborhood Component Analysis (NCA) as a promising alternative technique for feature extraction in load recognition. Besides reporting the quantitative improvements over existing methods, we also emphasize the qualitative benefits, such as the enhanced system reliability through employing the Regularized Extreme Learning Machine (RELM) classifier instead of the standard ELM. Furthermore, our work seeks to optimize the capabilities of ML models to the maximum through the combination of Grid Search (GS) with K-fold Cross-Validation (K-CV). The results of the proposed system reveal the highest accuracy values, weighted average F1-Score (F_1), and Kappa index ($\mathcal{K}$) when compared to the most modern methods in the literature. The innovations implemented also guarantee low training time. These results confirm the superiority of the innovations proposed in this manuscript. Furthermore, our solution is part of the research project under development named Open Middleware and Energy Management System for the Home of the Future. The project is a collaboration between the University of Campinas, the Eldorado Research Institute, and the Brazilian energy supplier Companhia Paranaense de Energia (COPEL).

Major Contributions

The principal contributions of our work are as follows:

1. Gaps like enhancing classification performance via approaches dedicated to separability between classes and models that reach improved reliability remain open. According to Manit and Youngkong [18], NCA provides enhanced class separability. In this study, we adopt NCA to extract relevant characteristics from the data, seeking the separability between classes to improve classification performance. We also apply the RELM to achieve higher reliability when identifying household appliances. It is relevant to mention that this work is the first to use NCA and RELM for load recognition. Besides the RELM-NCA pair, it is the first to verify the NCA-ELM pair for this task;
2. The values obtained for the performance metrics reveal the promising potential of ML techniques, specifically NCA and RELM, to contribute to load recognition. Using the 'Personalised Retrofit Decision Support Tools for UK Homes using Smart Home Technology' (REFIT), the proposed approach achieved 97.24% accuracy and 97.14% F1. As well as via the Reference Energy Disaggregation Dataset' (REDD), the method reached 96.53% accuracy and 96.48% F1. For both, the proposed approach has achieved performance improvements in load recognition. For the REFIT dataset, the difference between RELM and ELM reveals a 0.36% accuracy advantage for the RELM model. This advantage is the same for RELM compared to the SVM of the state-of-the-art system proposed in Cabral et al. [7]. For the REDD dataset, RELM's advantage over SVM is 0.21%. RELM's advantage over ELM for the same dataset is 2.71% accuracy. Furthermore, we can see a trend favoring the proposed method when we examine the accuracy of other state-of-the-art methods from the references. When comparing the best result of the proposed method, 97.24% accuracy, with the third-placed method in Qaisar and Alsharif [14], the difference is 1.84%;
3. Our method provides an ultra-low training time of 0.082 s with the REFIT database, less than the SVM of the technique reported in Cabral et. al [7], which has a time of 0.469 s. This result means that the proposed approach is approximately 5.72 times faster than the competitor, representing a time saving of 82.52% compared to the competitor. Concerning ELM, the proposed approach is 2.33 times faster and saves approximately 57.07% of the time. When checking the REDD database, the proposed method has a training time of 0.123 s, while the SVM has 0.167 s. In this case, the proposed method is 1.36 times faster than the SVM, saving approximately 26.35% of the time. For the REDD dataset, the proposed approach requires more time than ELM to complete its training process. However, only the proposed approach has the shortest achievable time compared to the other methods: 0.082 s.

The structure of the remainder of this paper is as follows: Section 2 provides a detailed background to lay the foundation for this study. Section 3 provides a detailed description of the proposed system, elucidating the implemented processing flow. This section includes detailed input data, the feature extraction technique employed, the criteria for selecting appropriate components, and the implementation of machine learning models. Section 4 presents the metrics utilized in this manuscript, accompanied by a justification for their choice. Section 5 examines the results obtained from the proposed system when employing two databases. Furthermore, this section discusses the outcomes, offering insights and interpretations of the findings. Section 6 concludes the paper, summarizing the contributions, implications of the proposed strategy, and potential future work.

2. Background

This section presents the related works in the literature and introduces the theoretical principles, such as the feature extraction technique and the architecture of the ML model.

2.1. Related Works

In the literature, there are a variety of strategies that perform load recognition. In Qaisar and Alsharif [14], the authors use active power. Active power performs work in an electrical system and refers to the energy consumed by the appliance to operate. However, the study uses different procedures to classify each device category. The paper uses devices from the Appliance Consumption Signature-Fribourg 2 (ACS-F2) database and the accuracy metric to evaluate the performance of the SVM and kNN models at the devices' classification stage. In Cabral et al. [7], the authors exclusively use the active power profile of the household appliances in the REDD and REFIT datasets. In this reference, the *k*-NN, Decision Tree (DT), Random Forest (RF), and SVM models can perform load recognition with a balance between short training time and high performance for the accuracy metrics, weighted average F1 score, and Kappa index.

Nonetheless, approaches do not necessarily only exploit the active power of a household appliance. The study reported by Mian Qaisar and Alsharif [8] uses active and reactive power. Reactive power in an alternating current system does not perform effective work, being stored and returned to the electrical system due to the presence of reactive elements, such as capacitors and inductors. In this case, the authors also apply accuracy to evaluate the performance of the ANN and *k*-NN models. In Matindife et al. [11], the researchers use a private dataset involving active power, current, and power factor. Here, by employing the Gramian Angular Difference Field (GADF) for feature extraction; the researchers convert the data into images. In the sequence, the CNN recognizes the appliances and the authors test the robustness of their proposal using the recall, precision, and accuracy metrics.

On the other hand, alternative studies utilize different types of data, as demonstrated in Borin et al. [19], which employs instant current measurements. The study applies Vector Projection Classification (VPC) in pattern recognition of loads. In this reference, the authors assess the performance of the proposed approach through the percentage of identified devices. Some methods utilize a combination of these other variables, such as voltage and current, for instance. In Zhiren et al. [16], the study uses a private dataset. The authors evaluate the proposed solution through accuracy metric, where the models tested are ELM, Adaboost-ELM, and SVM. In Faustine and Pereira [10], the scientists employ the Plug Load Appliance Identification Dataset (PLAID) dataset and the F_1 example-based (F_1-eb) and F_1 macro-average (F_1-macro) metrics in their analysis. The methodology proposed in this reference focuses on the Fryze power theory, which decomposes the current characteristics into components. As a result, the current becomes an image-like representation and a CNN recognizes the loads.

Unlike the utilization of voltage and current profiles, certain studies consider alternative attributes. Heo et al. [13] use Amplitude–Phase–Frequency (APF). The researchers employ accuracy and the F_1-Score as metrics to evaluate the overall performance of the proposed system and the following databases: Building-Level fully labeled Electricity Disaggregation (BLUED), PLAID, and a private database. As reported in the study, the use of HT-LSTM improves the recognition of devices with differences in the transient time and transient form of the load signal. Furthermore, the proposed scheme includes an event detection stage. Event detection is not always present in the strategies published in the literature but it is a tool that allows the system to identify when the appliance has turned on and off. The references Cabral et al. [7], Anderson et al. [20], Norford and Leeb [21], and Le and Kim [22] contain event detection strategies. It is relevant to mention that event detection is not the focus of our proposed work. Nevertheless, we use Wavelet transform to detect the ON/OFF status of the appliances according to references Lemes et al. [6] and Cabral et al. [7]. The selection of the Wavelet transform is justified due to its ability to detect appliance activity simply through the analysis of the level 1 detail coefficient. According to Lemes et al. [6], level 1 already contains enough information to detect ON/OFF activity. Hence, detecting the ON/OFF activity of the appliance can be performed without needing to decompose the signal into higher levels.

Another significant factor is the volume of data involved in the proposed approaches; more attributes to consider mean a more computationally complex and invasive system. In Huang et al. [12], the authors consider the steady-state power, the amplitude of the fundamental wave of the transient current, the transient voltage, and the harmonics of the transient current. In this case, the work adopts the REDD dataset and F_1-Score in the tests. The methodology combines LSTM layers with Back Propagation (BP) layers, resulting in the following architecture: the Long Short-Time Memory Back Propagation (LSTM-BP) network. The method described by Soe and Belleudy [15] uses characteristics from the active power of the equipment present in the Appliance Consumption Signature-Fribourg 1 (ACS-F1). Such features are the maximum power, average power, standard deviation, number of signal processing, operating states, and activity number of the appliances. The article evaluates the performance of the SVM, *k*-NN, CART, LDA, Logistic Regression (LR), and Naive Bayes (NB) models in terms of accuracy. It is worth emphasizing that as the diversity of electrical signals and parameters demanded from home appliances increases, the load recognition method becomes more intrusive and computationally expensive. For this reason, creating a strategy that has an optimal balance between high performance, reliability, and short training time based on a single type of electrical signal represents a key challenge in the load recognition field.

Finally, it is essential to mention some shortcomings in the methods proposed in the literature. Few studies use only one type of electrical signal in their approaches. The greater the number of electrical signals involved, the more invasive and computationally costly the method becomes. For example, the works of Mian Qaisar and Alsharif [8], Qaisar and Alsharif [14], and Zhiren et al. [16] use many parameters excessively. On the other hand, the majority of existing studies do not include a stage for detecting the ON/OFF status of the equipment, for example, the works of Mian Qaisar and Alsharif [8], Soe and Belleudy [15], and Zhiren et al. [16]. This condition limits the practical use of these methods. Most studies in the literature do not consider applying procedures to optimize their approaches, such as the hyperparameter search, for example, Faustine and Pereira [10], Qaisar and Alsharif [14], and Soe and Belleudy [15]. Adopting this procedure supports the definition of classifier structural parameters and can provide additional performance gains. Other papers are not concerned with evaluating the reliability of the system.

2.2. Feature Extraction

Feature extraction concerns the process of transforming relevant characteristics from raw data to create more compact and informative representations. The extracted features describe distinctive properties of the data and practitioners widely apply this approach across several areas, such as image processing, according to Nixon and Aguado [23] and Chowdhary and Acharjya [24], signal processing, in line with Gupta et al. [25] and Turhan-Sayan [26], and ML according to Musleh et al. [27] and Kumar and Martin [28]. One of the advantages of some feature extraction techniques is the reduction in data dimensionality, thereby decreasing computational complexity.

Several techniques exist for feature extraction. The approach choice depends on the nature of the data, the task concerned, and the computational cost involved. Some studies, according to Veeramsetty et al. [29] and Laakom et al. [30], employ autoencoders for compact data representations. However, this kind of architecture can make methods computationally expensive. Alternative investigations use computational techniques that are less resource-intensive, such as in Reddy et al. [31] with LDA, Fang et al. [32] with Independent Component Analysis (ICA), and Bharadiya [33] and Kabir et al. [34] with PCA.

Currently, more modern methods employ NCA to eliminate redundant information to reduce computational cost, according to Ma et al. [35]. NCA is a technique focusing on learning a distance metric in the feature space, optimizing the similarity between points without necessarily decreasing the dimensionality of the data. As per Goldberger et al. [36], the NCA technique is based on *k*-NN and stands out for optimizing a distance metric to enhance the quality of features, especially in classification tasks where the distinction between classes is crucial.

According to Singh-Miller et al. [37], the NCA algorithm uses the training set as the input, i.e., $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_Q\}$ with $\mathbf{x}_i \in \mathbb{R}^Q$, and y_i with the set of labels $\{y_1, y_2, \cdots, y_Q\}$. The algorithm needs to learn a projection matrix $\mathbf{A}$ of dimension $q \times Q$, which it uses to project the training vectors $\mathbf{x}_i$ into a low-dimensional representation of dimension q. To obtain low-dimensional projection, NCA requires learning a quadratic distance metric γ, that optimize the performance of k-NN. The distance γ between two points, $\mathbf{x}_i$ and $\mathbf{x}_j$, is

$$\gamma_{ij} = \gamma(\mathbf{x}_i, \mathbf{x}_j) = \| \mathbf{A}(\mathbf{x}_i - \mathbf{x}_j) \|^2, \tag{1}$$

where $\| \cdot \|$ is the Frobenius norm and $\mathbf{A}$ is a linear transformation matrix. In the NCA technique, each point i chooses another point j as its neighbor from among k points with a probability P_{ij} and assumes the class label of the selected point. According to Goldberger et al. [36], through γ_{ij}, P_{ij}, and the optimization of the objective function $f(\mathbf{A})$, NCA calculates a vector representation in a low-dimensional space ($\mathbf{x}_i^{(q)}$). The vector in low-dimensionality can be represented by

$$\mathbf{x}_i^{(q)} = \mathbf{A}\mathbf{x}_i, \tag{2}$$

It is possible to produce a matrix in low-dimensional space, depending on the implementation. This scenario is subject to the input data and the number of components required for the application. Detailed information regarding the NCA algorithm is available at Goldberger et al. [36].

2.3. Extreme Learning Machine (ELM)

The ELM presents a visionary structure in the ML field, standing out for its computational efficiency and conceptual simplicity. In contrast to many conventional neural networks, where all parameters need adjustment during training, ELMs adopt a unique strategy by randomly fixing the weights of the hidden layer and focusing solely on learning the weights of the output layer. This methodology enables a short training time. Furthermore, the simplified architecture of ELMs facilitates implementation, making them an appealing choice for applications requiring computational efficiency and robust performance in supervised learning tasks.

As per the formal description of ELM in accordance with Huang et al. [38], for different samples $(\mathbf{x}_i, \mathbf{t}_i)$, where $\mathbf{x}_i = [x_{i1}, x_{i2}, \cdots, x_{in}]^T \in \mathbb{R}^n$ and $\mathbf{t}_i = [t_{i1}, t_{i2}, \cdots, t_{im}]^T \in \mathbb{R}^m$, the output of an ELM is

$$\sum_{i=1}^{L} \boldsymbol{\beta}_i g(\langle \mathbf{w}_i \,,\, \mathbf{x}_j \rangle + b_i) = \mathbf{t}_j, \;\; j = 1, 2, 3, \;\cdots, N, \tag{3}$$

in which $\mathbf{w}_i = [w_{i1}, w_{i2}, \cdots, w_{in}]^T$ is the input weight vector connecting the ith hidden neuron, $\boldsymbol{\beta}_i = [\beta_{i1}, \beta_{i2}, \cdots, \beta_{im}]^T$ is the output weight vector connecting the ith hidden neuron, b_i is the bias, L is the number of hidden neurons, $g(\cdot)$ is the activation function, and $\langle \cdot \,,\, \cdot \rangle$ is the inner product. Nevertheless, the Equation (3) can be written in the matrix form as

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \tag{4}$$

$\mathbf{H}$ is the output matrix of the hidden layer of the neural network and can be expressed as follows:

$$\mathbf{H} = \begin{bmatrix} g(\langle \mathbf{w}_1 \,,\, \mathbf{x}_1 \rangle + b_1) & \cdots & g(\langle \mathbf{w}_L \,,\, \mathbf{x}_1 \rangle + b_L) \\ \vdots & \ddots & \vdots \\ g(\langle \mathbf{w}_1 \,,\, \mathbf{x}_N \rangle + b_1) & \cdots & g(\langle \mathbf{w}_L \,,\, \mathbf{x}_N \rangle + b_L) \end{bmatrix}, \tag{5}$$

where

$$\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \vdots \\ \boldsymbol{\beta}_L^T \end{bmatrix} \text{ and } \mathbf{T} = \begin{bmatrix} \boldsymbol{t}_1^T \\ \vdots \\ \boldsymbol{t}_N^T \end{bmatrix} \tag{6}$$

However, we can solve the system described in (4) through the Moore–Penrose pseudo-inverse of the $\mathbf{H}$, depicted as $\mathbf{H}^{\dagger}$, where $\mathbf{H}^{\dagger} = \left[(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T \right]$. Consequently, we can determine the output weights by

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^{\dagger}\mathbf{T} = \left[\left(\mathbf{H}^T\mathbf{H} \right)^{-1} \mathbf{H}^T \right] \mathbf{T} \tag{7}$$

Figure 1 illustrates the standard ELM, one of the ML models of the proposed system. Later in the manuscript, our approach includes modifications to the standard ELM to achieve enhanced results.

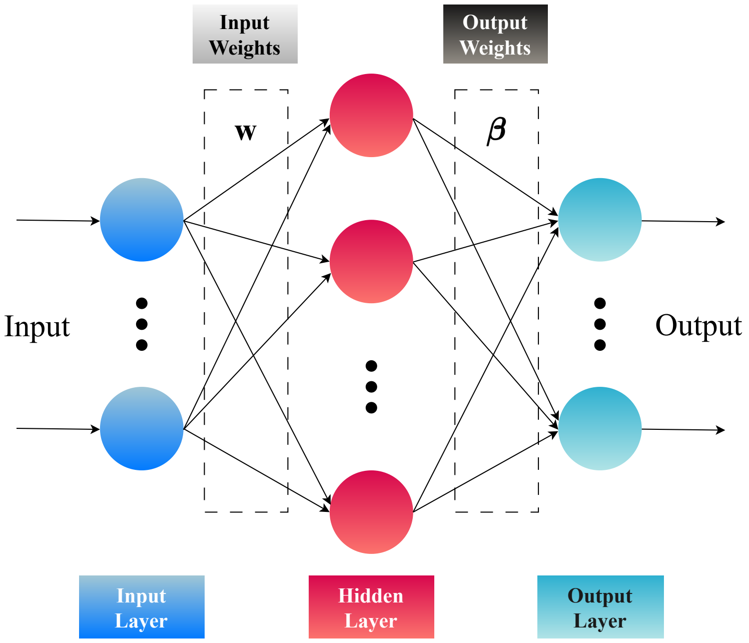

Figure 1. ELM-standard model (adapted from Zhao et al. [39]).

3. Proposed System

Figure 2 provides an overview of the proposed system for load recognition, where the collection and transmission of data are carried out through smart outlets and the controller, respectively. Both the smart outlets and the controller are illustrated in the blue color. In the second panel, the collected data are highlighted in blue, while the preliminary data processing blocks are represented in light gray. In the sequel, we have the feature extraction stage utilizing the NCA technique, represented in red. Subsequently, in dark gray, in the last part of the flowchart, Figure 2 depicts the stage responsible for the ML model optimization, aiming for the enhanced performance of the classifiers. At the end, depicted in light blue, the system presents the type of appliance in operation.

Figure 2a depicts the HEMS system consisting of the controller and smart outlets. The controller enables the processing of data either locally or its transmission to a cloud server. Moreover, it can execute pre-trained algorithms and send consumption alerts to the end user. The solution is minimally invasive. In practice, users only need to provide internet access to the controller and pair it with the smart outlets.

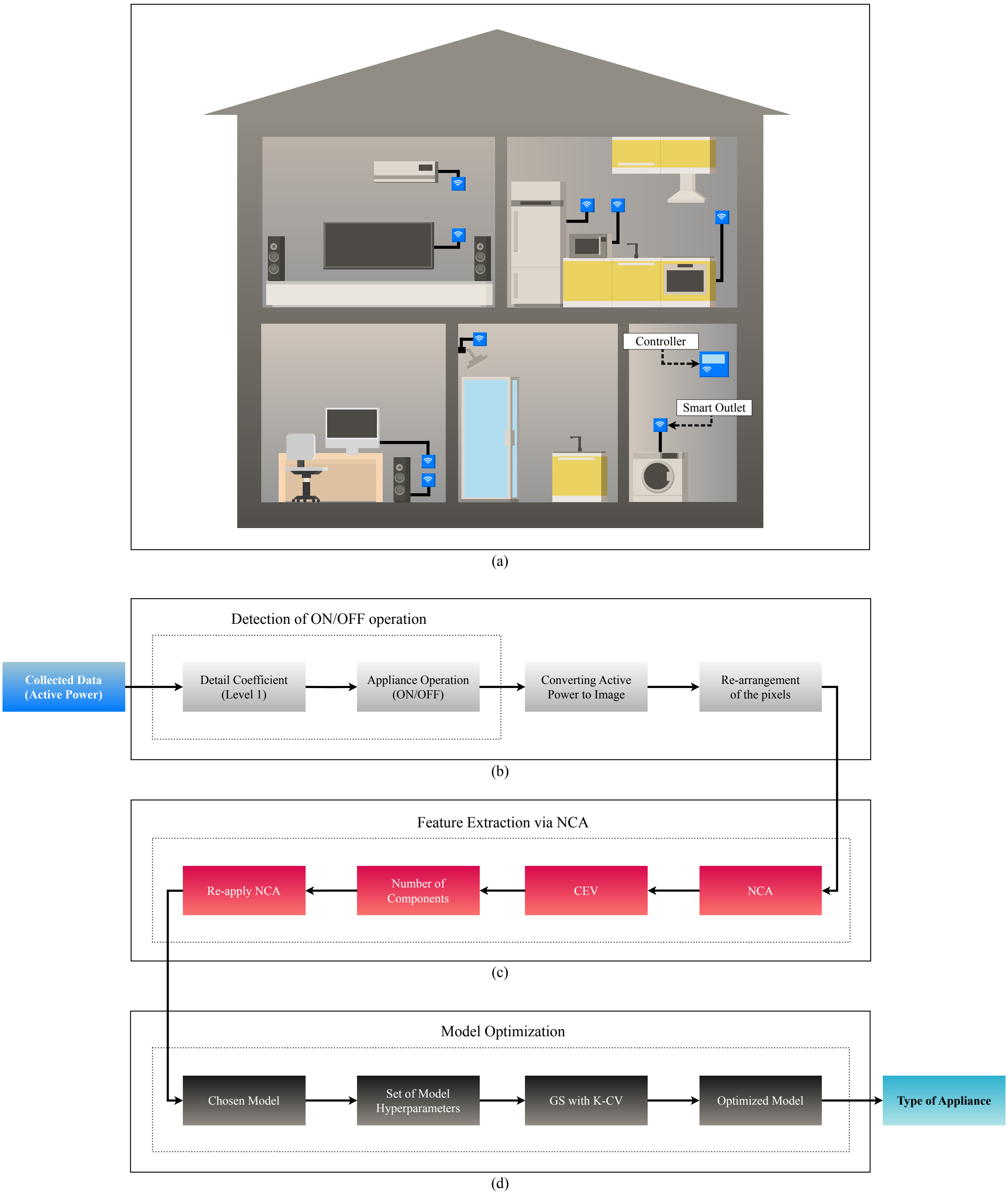
Controller
Smart Outlet
(a)
Detection of ON/OFF operation
Collected Data (Active Power)
Detail Coefficient (Level 1)
Appliance Operation (ON/OFF)
Converting Active Power to Image
Re-arrangement of the pixels
(b)
Feature Extraction via NCA
NCA
CEV
Number of Components
Re-apply NCA
(c)
Model Optimization
Chosen Model
Set of Model Hyperparameters
GS with K-CV
Optimized Model
Type of Appliance
(d)

Figure 2. Comprehensive visualization of the Load Recognition System. This figure outlines the four stages of the process, starting with (**a**) the collection of active power through HEMS. The next phase (**b**) involves the detection of appliances' ON/OFF status and preliminary data handling. In the sequel, (**c**) feature extraction is conducted using the NCA technique. The process culminates with (**d**) the optimization of ML models for improved classifier performance and the identification of the operational appliance type.

According to Figure 2b, the system features an ON/OFF operational state detection stage via Discrete Wavelet Transform (DWT) to determine when the appliance is in operation. An appliance registers as operating when it exhibits non-zero active power values, even after employing a prior filtering of potential noise. To determine whether an appliance is in operation as demonstrated in Lemes et al. [6], we only need the level-1 detail coefficients obtained by decomposing the active power using DWT with the Daubechies 4 mother wavelet. Subsequently, the system transforms the segments with the detected activities into images, as described in Cabral et al. [7]. In this procedure, the proposed strategy converts the curve representing the electrical activity of the appliance into black pixels and the background into white pixels, where the system has the flexibility to adjust the pixel resolution based on computational cost requirements. Therefore, each generated image contains a cycle of appliance activity. Following this, the system rearranges the image rows into a column vector of size I. Afterward, the method generates a set of vectors containing N images and places this set into a matrix $\mathbf{S}$, with dimensions $I \times N$.

Figure 2c illustrates the processing chain for feature extraction via NCA. In our case, we apply NCA to the pixel matrix $\mathbf{S}$, transforming the data from $\mathbf{S}$, with high dimensionality, into the matrix $\mathbf{S}^{(q)}_{(\mathrm{NCA})}$, with low dimensionality (q). Once we have the learned transformations applied to the data, we can estimate the variance of the transformed data for each dimension or component. Thus, it is feasible to assess the fraction of variance contained in each dimension and calculate the Cumulative Explained Variance (CEV) as more dimensions or components are incorporated. Then, we can choose the optimal number of components using CEV. The different components are defined based on the CEV, as per Algorithm 1. To detail this stage, Algorithm 1 specifies all the procedures of feature extraction through NCA. In particular, Algorithm 1 encompasses the sequence of procedures necessary to calculate the CEV, determine the number of components through CEV, use the NCA technique with the optimized number of elements, and obtain the transformed data via NCA.

Figure 2d represents the processing chain dedicated to optimizing the ML model. During this phase, the system feeds the ML models with the transformed data. For the ELM model, the processing chain conducts a hyperparameter search through GS and K-CV, employing the candidate set for the number of neurons and the number of folds. Algorithm 2 details the instructions for the ELM model optimization process. It is relevant to mention that our system can employ the optimization process in ELM or RELM. Nonetheless, there is a difference between the ELM and RELM models. Such a particularity is the regularization coefficient. This element is a parameter used in ML techniques, such as ridge regression, to control the model's fit to the data. The function of the regularization coefficient is to impose a penalty on the magnitude of the model coefficients, preventing them from reaching excessively high values. Such a strategy minimizes the risk of overfitting by preventing the model from overly adjusting to the data.

The regularization coefficient is a mechanism that aids in achieving an appropriate balance between bias and variance. An elevated regularization coefficient leads to an increase in the bias and a decrease in the variance. In contrast, a small regularization coefficient allows for greater flexibility in model fitting to the data, promoting an increase in variance. Such a coefficient acts as a regulator, influencing the synthesis of a simpler or a more complex model. Therefore, appropriately adjusting the regularization coefficient results in a more balanced trade-off between bias and variance.

For this reason, the regularization coefficient inclusion prevents the model from becoming excessively specific to the training data, making it more robust to new datasets unseen during training. This effect enhances the capability of the model generalization. Thus, a model with a suitable regularization coefficient tends to preserve the relevance of identified patterns, even in different contexts from the training data. Therefore, it is necessary to carefully seek the appropriate value for the regularization coefficient, ensuring that the model effectively enhances its generalization capability.

Algorithm 1 Feature extraction using neighborhood component analysis with component selection based on cumulative explained variance

```
Input: Image dataset generated ($\mathbf{S}$), initial number of components ($\phi$), threshold ($\psi$)
Output: training set $\mathbf{S}^{(q)}_{(\text{tr,NCA})}$ e testing set $\mathbf{S}^{(q)}_{(\text{ts,NCA})}$
first method:
    Split the $\mathbf{S}$ database and derive the $\mathbf{S}_{(\text{tr})}$ training data and the $\mathbf{S}_{(\text{ts})}$ test data.
second method:
    Train the NCA with $\mathbf{S}_{(\text{tr})}$, using $\phi$ initial components. After training, obtain the transformed dataset $\mathbf{S}^{(\phi)}_{(\text{tr,NCA})}$
third method:
    Estimate the covariance matrix $\mathbf{C}_{(\text{NCA})}$, according to Lemes et al. [5], for the transformed data $\mathbf{S}^{(\phi)}_{(\text{tr,NCA})}$
fourth method:
    Calculate the eigenvalues $e_i$ through $\mathbf{C}_{(\text{NCA})} = \mathbf{E} \cdot diag(e_1, e_2, \cdots, e_\phi) \cdot \mathbf{E}^{-1}$, where $\mathbf{E}$ is the eigenvectors matrix e $diag(e_1, e_2, \cdots, e_\phi)$ is the diagonal matrix containing the eigenvalues
fifth method:
    Order the eigenvalues in descending sequence: $e_1 \geq e_2 \geq e_3 \geq \cdots \geq e_\phi$
sixth method:
    Compute the proportion of variance explained by each eigenvalue $\rho_i = \frac{e_i}{\sum_{j=1}^{\phi} e_j}$
seventh method:
    Obtain the CEV for the $i$-th component: $\text{CEV}_i = \sum_{j=1}^{i} \rho_j$
eighth method:
    Create the $q$ variable to receive the optimized number of components and initialize $q = 0$
    Determine the optimized number of components
        if $\text{CEV}_i \geq \psi$
            $q \leftarrow$ number of $i$-th component,
        end if
ninth method:
    Re-train NCA with $\mathbf{S}_{(\text{tr})}$, this time utilizing only $q$ components. After to the training process, apply NCA to generate the transformed datasets for both the training set, denoted as $\mathbf{S}^{(q)}_{(\text{tr,NCA})}$, and the test set, denoted as $\mathbf{S}^{(q)}_{(\text{ts,NCA})}$
    return $\mathbf{S}^{(q)}_{(\text{tr,NCA})}$ e $\mathbf{S}^{(q)}_{(\text{ts,NCA})}$
```

Obtaining the most suitable regularization coefficient for the RELM model through trial and error can be a considerable challenge. Once again, this objective becomes feasible through hyperparameter search strategies, such as GS and K-CV. Structurally, adding a regularization coefficient to the ELM architecture to obtain the RELM model involves modifying the pseudo-inverse of **H**, as per Equation (8). This approach is an elegant strategy that ensures excellent results.

$$\hat{\beta}_\lambda = \mathbf{H}^\dagger \mathbf{T} = \left[\left(\mathbf{H}^T \mathbf{H} + \lambda \mathbf{I} \right)^{-1} \mathbf{H}^T \right] \mathbf{T} \tag{8}$$

Nevertheless, in RELM optimization, it is necessary to include candidates set for the regularization coefficient. In this phase, the system considers the set of candidates for the number of neurons, the set of candidates for the regularization coefficient, and the number of folds during the hyperparameter search. In this case, Algorithm 3 describes the RELM model optimization process in detail.

Algorithm 2 Optimizing extreme learning machine through hyperparameter tuning with grid search and K-fold cross-validation for enhanced performance

```
Input: Candidates for the number of neurons $(\eta_1, \eta_2, \cdots, \eta_n)$, $\mathbf{S}^{(q)}_{(\text{tr,NCA})}$, $\mathbf{S}^{(q)}_{(\text{ts,NCA})}$, number of folds (K)
Output: Optimized ELM
  first method:
  Load the candidates for the hyperparameters: candidates for the number of neurons $(\eta_1, \eta_2, \cdots, \eta_n)$
  second method:
  Employ GS with K-CV
      Divide $\mathbf{S}^{(q)}_{(\text{tr,NCA})}$ in K folds
      Train the model for each fold
      Compute accuracy
      Mean accuracy
      Attributes the average accuracy to the present set of hyperparameters
      Choose the hyperparameter set with the highest average accuracy achieved, i.e, $\eta_{(\text{optimal})}$
  third method:
  Train the model using $\eta_{(\text{optimal})}$
  fourth method:
  Testing the optimized model with $\mathbf{S}^{(q)}_{(\text{ts,NCA})}$
  return Optimized ELM
```

Algorithm 3 Optimizing regularized extreme learning machine through hyperparameter tuning with grid search and K-fold cross-validation for improved performance

```
Input: Candidates for the number of neurons $(\eta_1, \eta_2, \cdots, \eta_n)$, candidates for the regularization coefficient $(\lambda_1, \lambda_2, \cdots, \lambda_n)$, $\mathbf{S}^{(q)}_{(\text{tr,NCA})}$, $\mathbf{S}^{(q)}_{(\text{ts,NCA})}$, number of folds (K)
Output: Optimized RELM
  first method:
  Load the candidates for the hyperparameters: candidates for the number of neurons $(\eta_1, \eta_2, \cdots, \eta_n)$ and candidates for the regularization coefficient $(\lambda_1, \lambda_2, \cdots, \lambda_n)$
  second method:
  Employ GS with K-CV
      Divide $\mathbf{S}^{(q)}_{(\text{tr,NCA})}$ in K folds
      Train the model for each fold
      Compute accuracy
      Mean accuracy
      Attributes the average accuracy to the present set of hyperparameters
      Choose the hyperparameter set with the highest average accuracy achieved, i.e, $\lambda_{(\text{optimal})}$ and $\eta_{(\text{optimal})}$
  third method:
  Train the model using $\lambda_{(\text{optimal})}$ and $\eta_{(\text{optimal})}$
  fourth method:
  Testing the optimized model with $\mathbf{S}^{(q)}_{(\text{ts,NCA})}$
  return Optimized RELM
```

4. Performance Evaluation Metrics

To evaluate the performance of the proposed system, this work uses the metrics of accuracy, F_1, and $\mathcal{K}$, according to Cabral et al. [7]. These metrics rely on measures of true-positive (TP), true-negative (TN), false-positive (FP), and false-negative (FN).

The accuracy, defined in (9), is responsible for measuring the number of instances correctly classified in the test set and presents the overall performance of the models.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{9}$$

Depending on the nature of the devices, each appliance can produce a different number of events. It is possible to consider this effect in the F_1-Score metric to provide a fair performance analysis. However, it is necessary to consider the size of the set of instances of a class (d) and the size of the dataset (D), as per Alswaidan and Menai [40] and Guo et al. [41]. To achieve it, this study employs the weighted average F_1-score, denoted as F_1, according to (10)

$$\begin{aligned} \text{F}_1 &= \frac{1}{D} \sum d \times \text{F}_1\text{-Score} \\ &= \frac{1}{D} \sum d \times \left[\frac{2 \times \text{TP}}{2 \times \text{TP} + 1 \times (\text{FN} + \text{FP})} \right] \end{aligned} \tag{10}$$

This work applies the Kappa index to verify the agreement of the proposed strategy. As per Matindife et al. [11], Kappa operates within the interval of $[-1, 1]$. A value of -1 signifies an absence of agreement, 0 represents agreement occurring by chance, and 1 indicates perfect agreement. This manuscript defines Kappa index, denoted as $\mathcal{K}$, by (11)

$$\mathcal{K} = \frac{2 \times (\text{TP} \times \text{TN} - \text{FN} \times \text{FP})}{(\text{TP} + \text{FP}) \times (\text{FP} + \text{TN}) + (\text{TP} + \text{FN}) \times (\text{FN} + \text{TN})} \tag{11}$$

Each metric offers a unique perspective on the performance of ML models, ensuring a holistic assessment. Accuracy is essential as it provides an overall success rate of the model. F_1 was selected to address a potential imbalance among classes in used datasets. This metric offers a more nuanced view of the model in scenarios when the class distribution is imbalanced. The $\mathcal{K}$ evaluates the agreement between the predicted and observed classifications, providing insights into the model's performance regarding reliability. By employing these three metrics together, we aim to present a comprehensive evaluation of the performance of our model, capturing its effectiveness in classification and its robustness in different evaluation parameters. This multifaceted approach allows us to validate the model's utility in diverse scenarios, ensuring its reliability and applicability.

5. Results and Discussions

This study assesses the proposed solution using two databases, REDD from Kolter and Johnson [42] and REFIT from Murray et al. [43]. These databases encompass two households characterized by distinct measurement frequencies, types of equipment, and equipment number. The deliberate selection of these databases ensures a comprehensive evaluation of the method generalization. The initial procedure is to homogenize the resolution at 32 × 32 pixels for both databases. The system generates 4609 images from the REDD database and 2723 images from the REFIT database. Procedures involving training in the feature extraction and in the model optimization consider a data partition of 80% for training and 20% for testing in both databases, respectively. The proposed system applies feature extraction via NCA. Related work involving PCA indicates that the REFIT database requires more components than REDD. Concerning this, the proposed solution adopts $\phi = 300$ for REFIT and $\phi = 100$ for REDD. It is necessary to mention that these ϕ values are just initial values; our system endogenously determines the most appropriate number of components via CEV. For this task, we use a threshold of $\psi = 0.99$ for both REFIT and REDD. The threshold of $\psi = 0.99$ was selected based on previous studies that indicate the effective extraction of relevant components. For example, Cabral et al. [7] employ this value for the threshold to ensure that only components with significant contributions to the data variance are retained. In the sequence, the system initiates the optimization process for ML models. For both datasets, the hyperparameter tuning procedures use

K = 10 folds. According to Kuhn et al. [44], opting for K=10 is advocated because it can generate test error rate estimates unaffected by undue bias or excessive variance. Moreover, we also selected 10 to balance performance reliability and computational efficiency. In this search, the number of neurons $(\eta_1, \eta_2, \cdots, \eta_n)$ ranged from 100 to 1000, with a step size of 100, for the ELM and RELM models, respectively. The step size 100 was chosen to foster the model convergence without significantly compromising computational time. In addition, for the RELM model, we need to include the candidates for the regularization coefficient $(\lambda_1, \lambda_2, \cdots, \lambda_n)$ in the hyperparameter search. Thus, the set of the values for the regularization coefficient candidates ranged from 0.0001 to 0.1, with the search step increasing 10 times from one value to the next in the mentioned sequence.

5.1. Scenario Using the REFIT Dataset

For the first analysis scenario, this study utilizes the REFIT dataset. REFIT comprises active power measurements from 20 residences, recorded at a frequency of 1/8 Hz. For the REFIT scenario, we consider appliances of household 1. This household includes freezers, washer-dryers, washing machines, dishwashers, computers, televisions, and electric heaters. For this dataset, the system generated 4609 images representing the activities of the operational household appliances.

In this scenario, the system generates the results of Table 1 using the REFIT database from Murray et al. [43]. To preserve the layout of the manuscript, we show just some of the 300 components in Table 1. As highlighted in Table 1, using a threshold of $\psi = 0.99$ for the CEV, it is necessary to employ $q = 228$ components. Here, we use the GS and K-CV for a hyperparameter search of ELM and RELM. In this manner, for both models, the search specified the value of $\eta_{\text{(optimal)}} = 400$ for the number of neurons, respectively. Furthermore, the hyperparameter search determined $\lambda_{\text{(optimal)}} = 0.01$ for the RELM architecture.

Table 1. Evolution of the CEV according to the increment in the number of components (Comp.) for the REFIT dataset.

Comp.	CEV	Comp.	CEV	Comp.	CEV	Comp.	CEV	Comp.	CEV
1	0.0862	53	0.8421	105	0.9330	157	0.9682	209	0.9858
5	0.2761	57	0.8539	109	0.9368	161	0.9700	213	0.9867
9	0.3983	61	0.8643	113	0.9404	165	0.9717	217	0.9876
13	0.4907	65	0.8736	117	0.9438	169	0.9733	221	0.9885
17	0.5669	69	0.8820	121	0.9469	173	0.9748	225	0.9894
21	0.6275	73	0.8896	125	0.9499	177	0.9763	227	0.9898
25	0.6792	77	0.8966	129	0.9527	181	0.9777	228	0.9900
29	0.7202	81	0.9031	133	0.9553	185	0.9790	229	0.9902
33	0.7505	85	0.9091	137	0.9578	189	0.9802	233	0.9910
37	0.7751	89	0.9147	141	0.9601	193	0.9814	237	0.9917
41	0.7961	93	0.9198	145	0.9623	197	0.9826	241	0.9924
45	0.8135	97	0.9245	149	0.9644	201	0.9837	245	0.9931
49	0.8287	101	0.9290	153	0.9664	205	0.9847	249	0.9938

Upon checking the comparison in Table 2, it is evident that a slight performance difference exists among the ELM and RELM models concerning the accuracy metric. However, the RELM architecture exhibited the highest value, 97.24%, for accuracy. This advantage persists for the RELM model in the F_1 metric, with a value of 97.14%. Regarding the agreement between the predicted and expected values, the RELM model achieves the highest value of $\mathcal{K}$, 0.8300.

Table 3 compares the training times of the ELM and RELM models. In this scenario, the ELM classifier has the longest training time, at 0.191 s, followed by the RELM at 0.082 s. These results highlight the RELM model as having the shortest training time.

It is important to note that the obtained results suggest that the RELM architecture outperforms the previous model, ELM. We can extend this analysis further by comparing the second-best model in the literature for the same task, SVM in Cabral et al. [7], which achieves 96.88% accuracy, 96.61% to F_1, and 0.8375 to $\mathcal{K}$. In contrast, the RELM classifier surpasses these metrics with values of 97.24% accuracy, 97.14% to F_1, and 0.8300 to $\mathcal{K}$. Additionally, when examining the training time among the mentioned models, SVM has a longer training time in seconds, 0.469, compared to RELM, which only requires 0.123 of a second.

Table 2. Performance of the classifiers for the REFIT Scenario.

Classifier	Accuracy	F_1	$\mathcal{K}$
ELM	96.88%	96.59%	0.7910
RELM	97.24%	97.14%	0.8300

Table 3. Training time in seconds for the REFIT Scenario.

RELM	ELM
0.082	0.191

5.2. Scenario Using the REDD Dataset

REDD contains the active power readings of six homes at a frequency of 1/3 Hz. We use measurements from residence 1 for the REDD scenario. Within this household, a range of appliances is present, including an oven, refrigerator, dishwasher, kitchen oven, lighting, washer-dryer, microwave, bathroom Ground Fault Interrupters (GFI) outlet, heat pump, stoven, and an unidentified device. In this case, the system generates 2723 images illustrating the activities of household appliances.

In this scenario, the method produces the results displayed in Table 4, employing the REDD database from Kolter and Johnson [42]. As indicated by Table 4, to meet the threshold $\psi = 0.99$ imposed by the CEV, it is necessary to utilize 25 components, i.e., $q = 25$, for the REDD dataset. As mentioned earlier, GS and K-CV determine the most suitable hyperparameters for the models. For the ELM architecture, the hyperparameter search specified $\eta_{(\text{optimal})} = 100$. In this scenario, the search found $\eta_{(\text{optimal})} = 400$ and $\lambda_{(\text{optimal})} = 0.1$ for the RELM architecture.

Table 4. Evolution of the CEV according to the increment in the number of components (Comp.) for the REDD dataset.

Comp.	CEV	Comp.	CEV	Comp.	CEV	Comp.	CEV	Comp.	CEV
1	0.8992	7	0.9631	13	0.9788	19	0.9860	25	0.9904
2	0.9223	8	0.9669	14	0.9803	20	0.9868	26	0.9909
3	0.9387	9	0.9704	15	0.9817	21	0.9876	27	0.9915
4	0.9474	10	0.9732	16	0.9829	22	0.9884	28	0.9920
5	0.9533	11	0.9754	17	0.9841	23	0.9891	29	0.9924
6	0.9586	12	0.9771	18	0.9851	24	0.9897	30	0.9929

Table 5 presents a direct model performance comparison. The results reveal a significant difference in terms of accuracy when we compare the ELM and RELM models, this advantage is approximately 2.7% for RELM. Concerning F1, the difference between the ELM and RELM models is even more noteworthy. Here, the RELM architecture presents an advantage of 2.78% for the F_1 metric. Ultimately, the RELM classifier achieves the highest agreement between predicted and expected values, with $\mathcal{K} = 0.9388$.

Table 6 provides a training time comparison of the employed ML models. In this scenario, the training time for the RELM is 0.123 of a second, followed by the ELM, with 0.045 of a second. Although, in this scenario, the ELM has the shortest training time, this difference is indeed minimal.

In conclusion, the results suggest that the RELM model is the preferred choice for load recognition in this dataset. When compared again to the second-best model in the literature, SVM, the superiority of RELM is evident across all metrics. SVM achieves 96.31%, 96.36%, and 0.9381 for accuracy, F_1, and $\mathcal{K}$, respectively, while RELM reaches 96.53% for accuracy, 96.48% for F_1, and 0.9388 for $\mathcal{K}$. Furthermore, RELM has the shortest training time. While SVM requires 0.167 s for training, RELM only needs 0.082 s.

Table 5. Performance of the classifiers for a REDD Scenario.

Classifier	Accuracy	F_1	$\mathcal{K}$
ELM	93.82%	93.70%	0.8913
RELM	96.53%	96.48%	0.9388

Table 6. Training time in seconds for REDD Scenario.

ELM	RELM
0.045	0.123

5.3. State-of-the-Art Methods Comparison

Table 7 compares the proposed system with 11 state-of-the-art methods for load recognition. In this manner, Table 7 demonstrates that all the approaches may exhibit common structural characteristics, such as feature extraction procedures and earning models. However, solely the proposed system and the works of Cabral et al. [7] and Heo et al. [13] include event detection stages. Only the proposed system and the reference Cabral et al. [7] contain procedures for optimizing ML models.

Feature extraction procedures are vital structures for load recognition systems. However, the more characteristics we need to extract, the more computationally expensive the system becomes. The feature extraction procedures employed in the works of Qaisar and Alsharif [14], Zhiren et al. [16], Mian Qaisar and Alsharif [8], Soe and Belleudy [15], Faustine and Pereira [10], and Heo et al. [13] extract more than two types of characteristics from the signals, i.e., these methods need more information about the electrical signals to work. In this sense, the methods of Cabral et al. [7], Huang et al. [12], and our system use techniques to decrease the computational complexity. In the case of Cabral et al. [7] and Huang et al. [12], they use the PCA and in our case, we use the NCA technique. But comparing our system with the reference Cabral et al. [7], we require a smaller volume of data due to the utilization of a reduced number of components through NCA. While Cabral et al. [7] employ 269 components for REFIT and 35 for REDD, we require only 228 for REFIT and 25 for REDD.

The works of Faustine and Pereira [10], Matindife et al. [11], and De Baets et al. [9] use a more computationally complex architecture as a learning model, the CNN. However, as the last column of Table 7 shows, using more complex models does not guarantee a superior result. In this sense, the work of Matindife et al. [11] achieves 83.33% accuracy, while our method obtained 97.24% accuracy with RELM. Furthermore, only our study and the reference Cabral et al. [7] apply GS with K-CV to improve the performance of ML models. Nevertheless, the novelties in our work provide superior performance compared to Cabral et al. [7]. While reference Cabral et al. [7], the second-highest performing, shows 96.88% accuracy, using SVM for the REFIT dataset, we achieve 97.24% accuracy using RELM for the same dataset. It is relevant to point out that there is no consensus on the most suitable dataset for analyzing the methods. But in terms of metrics, most papers use accuracy as the principal evaluation metric, followed by F1-Score or a variation thereof, such as F_1 in Cabral et al. [7]. Once again, our method shows the highest value for both metrics, 97.24% for accuracy and 97.14% for F_1, while the reference Cabral et al. [7] shows 96.88% for accuracy and 96.61% for F_1. Our method also has the highest values for the $\mathcal{K}$ metric, demonstrating that our system has the highest rate of agreement for the results reached.

Table 7. Comparison of state-of-the-art approaches.

Load Recognition Strategies	Event Detection Stage	Feature Extraction Stage	Learning Model	Model Optimization	Metrics	Best Result	Model for the Best Result	Dataset for the Best Result
Our System	DWT	NCA	ELM and RELM	GS with K-CV	Accuracy, F_1, and $\mathcal{K}$	97.24% of Accuracy	RELM	REFIT
Ref. [7]	DWT	PCA	DT, *k*-NN, RF, and SVM	GS with K-CV	Accuracy, F_1, and $\mathcal{K}$	96.88% of Accuracy	SVM	REFIT
Ref. [14]	None	Extraction of electrical operating patterns	*k*-NN and SVM	None	Accuracy	95.40% of Accuracy	SVM	ACS-F2
Ref. [16]	None	Extraction of electrical quantity	ELM, AdaBoost-ELM, and SVM	None	Accuracy	94.80% of Accuracy	AdaBoost-ELM	Private
Ref. [8]	None	Extraction of energy consumption patterns from appliances	ANN and *k*-NN	None	Accuracy	94.40% of Accuracy	ANN	ACS-F2
Ref. [15]	None	Extraction of electrical operating patterns	CART, *k*-NN, LDA, LR, NB, and SVM	None	Accuracy	94.05% of Accuracy	*k*-NN	ACS-F1
Ref. [10]	None	Extraction of high-frequency features	CNN	None	F_1-eb and F_1-macro	94.00% of F_1-macro	CNN	PLAID
Ref. [13]	RMS Threshold	APF	HT-LSTM	None	Accuracy and F_1-Score	90.04% of Accuracy	HT-LSTM	PLAID
Ref. [19]	None	Stockwell transform	VPC	None	Identification percentage	90.00% of Accuracy	VPC	Private
Ref. [11]	None	GADF	CNN	None	Accuracy, precision, recall, F_1-Score, and $\mathcal{K}$	83.33% of Accuracy	CNN	Private
Ref. [9]	None	VI trajectories	CNN	None	F_1-macro, precision, and recall	77.60% of F_1-macro	CNN	PLAID
Ref. [12]	None	PCA	LSTM-BP	None	F_1-Score	45.49% of F_1-Score	LSTM-BP	REDD

By examining Table 7, it is worth mentioning that the accuracy reported in studies can be affected by the metrics and datasets used. For this reason, reliable studies present more than one metric for performance analysis and use more than one database. Studies that focus only on accuracy are limited in terms of method reliability. Therefore, additional performance metrics, such as F_1 and $\mathcal{K}$, are essential. Moreover, the dataset can influence the performance results. Studies that use more than one database tend to present a more robust analysis of the model's performance.

6. Conclusions

This manuscript presents significant improvements in the area of load recognition. This work is the first to use NCA for enhanced feature extraction and RELM to classify household appliances. Furthermore, our study is also a pioneer in verifying NCA-ELM and NCA-RELM pairs in load recognition. When employing RELM, our analysis unveils an exceptionally short training time of less than 1 s for both databases, REFIT and REDD. Specifically, during the examination of training time, we attained a training duration of 0.082 s with RELM. This duration is shorter than that achieved with the SVM architecture in Cabral et al. [7], which was the state of the art up to the present, with a time of 0.167 s. By analyzing the accuracy metrics, F_1 and $\mathcal{K}$, the superiority of RELM is evident. When compared to the state-of-the-art, RELM outperforms SVM in all the metrics. Whereas the SVM shows values of 96.88%, 96.61%, and 0.8375 for accuracy, F_1, and $\mathcal{K}$ in the REFIT database, RELM achieves values of 97.24%, 97.14%, and 0.8300 for accuracy, F_1, and $\mathcal{K}$ in the same database. The superiority of RELM extends to the REDD dataset, where SVM shows 96.31% accuracy, 96.36% F_1, and 0.9381 $\mathcal{K}$, whereas RELM reaches 96.53%, 96.48%, and 0.9388, respectively, for the same metrics. The proposed system demonstrates that the joint use of NCA and RELM is a viable and more robust alternative for load recognition, making the NCA-RELM pair a reliable and promising implementation.

The main drivers of the differences between the proposed method and competitors can be attributed to the innovative use of NCA with the RELM model. Feature extraction through NCA provides superior class separability and the application of the ELM yields higher reliability in identifying household appliances. This double focus is the main guide behind the improved performance of our system compared to existing approaches and paves the way for new possibilities for load recognition in HEMS systems. In this context, we suggest verifying additional datasets to evaluate their real-time implications for future research.

Author Contributions: Conceptualization, T.W.C. and L.G.P.M.; methodology, T.W.C. and L.G.P.M.; software, T.W.C.; validation, T.W.C. and L.G.P.M.; formal analysis, T.W.C. and L.G.P.M.; investigation, T.W.C. and L.G.P.M.; writing—original draft preparation, T.W.C.; writing—review and editing, T.W.C., E.R.d.L., F.B.N., G.F. and L.G.P.M.; project administration, F.B.N. and E.R.d.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Companhia Paranaense de Energia under Grant COPEL Projet: ANEEL-PD-02866-0508/2019.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The access to the data underlying the findings of this study is not available due to privacy considerations and in accordance with company operational policies.

Conflicts of Interest: Author F.B.N. was employed by the company Companhia Paranaense de Energia. Author F.R.d.L. was employed by the company Instituto de Pesquisa Eldorado. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. The authors declare that this study received funding from Companhia Paranaense de Energia. The funder was not involved in the study design, collection, analysis, interpretation of data, the writing of this article, however, it was involved in the decision to submit it for publication.

Abbreviations

The following abbreviations are used in this manuscript:

ACS-F1	Appliance Consumption Signature-Fribourg 1
ACS-F2	Appliance Consumption Signature-Fribourg 2
APF	Amplitude-Phase-Frequency
ANN	Artificial Neural Networks
BLUED	Building-Level fully labeled Electricity Disaggregation
CART	Classification and Regression Trees
CEV	Cumulative Explained Variance
CNN	Convolutional Neural Network
DWT	Discrete Wavelet Transform
DT	Decision Tree
ELM	Extreme Learning Machine
GADF	Gramian Angular Difference Field
GFI	Ground Fault Interrupter
GS	Grid Search
HEMS	Home Energy Management System
HT-LSTM	Hilbert Transform Long Short-Term Memory
K-CV	K-fold Cross-Validation
k-NN	*k*-Nearest Neighbors
LDA	Linear Discriminant Analysis
LR	Logistic Regression
LLR	Log Likelihood Ratio
LSTM	Long Short-Time Memory
LSTM-BP	Long Short-Time Memory Back Propagation
MIT	Massachusetts Institute of Technology
ML	Machine Learning
NB	Naive Baye
NCA	Neighborhood Components Analysis
PCA	Principal Component Analysis
PLAID	Plug Load Appliance Identification Dataset
RELM	Regularized Extreme Learning Machine
RF	Random Forest
REDD	Reference Energy Disaggregation Dataset
REFIT	Personalised Retrofit Decision Support Tools For UK Homes Using Smart Home Technology
RMS	Root Mean Square
SVM	Support Vector Machines
VPC	Vector Projection Classification
WHITED	Worldwide Household and Industry Transient Energy Dataset

References

1. Kim, T.Y.; Cho, S.B. Predicting residential energy consumption using CNN-LSTM neural networks. *Energy* **2019**, *182*, 72–81. [CrossRef]
2. Rashid, H.; Singh, P. Monitor: An abnormality detection approach in buildings energy consumption. In Proceedings of the 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC), Philadelphia, PA, USA, 18–20 October 2018; pp. 16–25.
3. Bang, M.; Engelsgaard, S.S.; Alexandersen, E.K.; Skydt, M.R.; Shaker, H.R.; Jradi, M. Novel real-time model-based fault detection method for automatic identification of abnormal energy performance in building ventilation units. *Energy Build.* **2019**, *183*, 238–251. [CrossRef]
4. Motta, L.L.; Ferreira, L.C.; Cabral, T.W.; Lemes, D.A.; Cardoso, G.d.S.; Borchardt, A.; Cardieri, P.; Fraidenraich, G.; de Lima, E.R.; Neto, F.B.; et al. General Overview and Proof of Concept of a Smart Home Energy Management System Architecture. *Electronics* **2023**, *12*, 4453. [CrossRef]
5. Lemes, D.A.M.; Cabral, T.W.; Fraidenraich, G.; Meloni, L.G.P.; De Lima, E.R.; Neto, F.B. Load disaggregation based on time window for HEMS application. *IEEE Access* **2021**, *9*, 70746–70757. [CrossRef]
6. Lemes, D.A.; Cabral, T.W.; Motta, L.L.; Fraidenraich, G.; de Lima, E.R.; Neto, F.B.; Meloni, L.G. Low Runtime Approach for Fault Detection for Refrigeration Systems in Smart Homes Using Wavelet Transform. *IEEE Trans. Consum. Electron.* **2023**. [CrossRef]

7. Cabral, T.W.; Lemes, D.A.M.; Fraidenraich, G.; Neto, F.B.; de Lima, E.R.; Meloni, L.G.P. High-Reliability Load Recognition in Home Energy Management Systems. *IEEE Access* **2023**, *11*, 31244–31261. [CrossRef]
8. Mian Qaisar, S.; Alsharif, F. Signal piloted processing of the smart meter data for effective appliances recognition. *J. Electr. Eng. Technol.* **2020**, *15*, 2279–2285. [CrossRef]
9. De Baets, L.; Ruyssinck, J.; Develder, C.; Dhaene, T.; Deschrijver, D. Appliance classification using VI trajectories and convolutional neural networks. *Energy Build.* **2018**, *158*, 32–36. [CrossRef]
10. Faustine, A.; Pereira, L. Multi-label learning for appliance recognition in NILM using Fryze-current decomposition and convolutional neural network. *Energies* **2020**, *13*, 4154. [CrossRef]
11. Matindife, L.; Sun, Y.; Wang, Z. Image-based mains signal disaggregation and load recognition. *Complex Intell. Syst.* **2021**, *7*, 901–927. [CrossRef]
12. Huang, L.; Chen, S.; Ling, Z.; Cui, Y.; Wang, Q. Non-invasive load identification based on LSTM-BP neural network. *Energy Rep.* **2021**, *7*, 485–492. [CrossRef]
13. Le, T.-T.-H.; Heo, S.; Kim, H. Toward load identification based on the Hilbert transform and sequence to sequence long short-term memory. *IEEE Trans. Smart Grid* **2021**, *12*, 3252–3264. [CrossRef]
14. Qaisar, S.M.; Alsharif, F. Event-Driven System For Proficient Load Recognition by Interpreting the Smart Meter Data. *Procedia Comput. Sci.* **2020**, *168*, 210–216. [CrossRef]
15. Soe, W.T.; Belleudy, C. Load recognition from smart plug sensor for energy management in a smart home. In Proceedings of the 2019 IEEE Sensors Applications Symposium (SAS), Sophia Antipolis, France, 11–13 March 2019; pp. 1–6.
16. Zhiren, R.; Bo, T.; Longfeng, W.; Hui, L.; Yanfei, L.; Haiping, W. Non-intrusive load identification method based on integrated intelligence strategy. In Proceedings of the 2019 25th International Conference on Automation and Computing (ICAC), Lancaster, UK, 5–7 September 2019; pp. 1–6.
17. Gao, J.; Kara, E.C.; Giri, S.; Bergés, M. A feasibility study of automated plug-load identification from high-frequency measurements. In Proceedings of the 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Orlando, FL, USA, 14–16 December 2015; pp. 220–224.
18. Manit, J.; Youngkong, P. Neighborhood components analysis in sEMG signal dimensionality reduction for gait phase pattern recognition. In Proceedings of the 7th International Conference on Broadband Communications and Biomedical Applications, Melbourne, Australia, 21–24 November 2011; pp. 86–90.
19. Borin, V.; Barriquello, C.; Campos, A. Approach for home appliance recognition using vector projection length and Stockwell transform. *Electron. Lett.* **2015**, *51*, 2035–2037. [CrossRef]
20. Anderson, K.D.; Berges, M.E.; Ocneanu, A.; Benitez, D.; Moura, J.M. Event detection for non intrusive load monitoring. In Proceedings of the IECON 2012—38th Annual Conference on IEEE Industrial Electronics Society, Montreal, QC, Canada, 25–28 October 2012; pp. 3312–3317.
21. Norford, L.K.; Leeb, S.B. Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms. *Energy Build.* **1996**, *24*, 51–64. [CrossRef]
22. Le, T.T.H.; Kim, H. Non-intrusive load monitoring based on novel transient signal in household appliances with low sampling rate. *Energies* **2018**, *11*, 3409. [CrossRef]
23. Nixon, M.; Aguado, A. *Feature Extraction and Image Processing for Computer Vision*; Academic Press: Cambridge, MA, USA, 2019.
24. Chowdhary, C.L.; Acharjya, D.P. Segmentation and feature extraction in medical imaging: A systematic review. *Procedia Comput. Sci.* **2020**, *167*, 26–36. [CrossRef]
25. Gupta, V.; Mittal, M.; Mittal, V.; Saxena, N.K. A critical review of feature extraction techniques for ECG signal analysis. *J. Inst. Eng. (India) Ser. B* **2021**, *102*, 1049–1060. [CrossRef]
26. Turhan-Sayan, G. Real time electromagnetic target classification using a novel feature extraction technique with PCA-based fusion. *IEEE Trans. Antennas Propag.* **2005**, *53*, 766–776. [CrossRef]
27. Musleh, D.; Alotaibi, M.; Alhaidari, F.; Rahman, A.; Mohammad, R.M. Intrusion Detection System Using Feature Extraction with Machine Learning Algorithms in IoT. *J. Sens. Actuator Netw.* **2023**, *12*, 29. [CrossRef]
28. Kumar, H.; Martin, A. Artificial Emotional Intelligence: Conventional and deep learning approach. *Expert Syst. Appl.* **2023**, *212*, 118651. [CrossRef]
29. Veeramsetty, V.; Kiran, P.; Sushma, M.; Babu, A.M.; Rakesh, R.; Raju, K.; Salkuti, S.R. Active Power Load Data Dimensionality Reduction Using Autoencoder. In *Power Quality in Microgrids: Issues, Challenges and Mitigation Techniques*; Springer: Singapore, 2023; pp. 471–494.
30. Laakom, F.; Raitoharju, J.; Iosifidis, A.; Gabbouj, M. Reducing redundancy in the bottleneck representation of autoencoders. *Pattern Recognit. Lett.* **2024**, *178*, 202–208. [CrossRef]
31. Reddy, G.T.; Reddy, M.P.K.; Lakshmanna, K.; Kaluri, R.; Rajput, D.S.; Srivastava, G.; Baker, T. Analysis of dimensionality reduction techniques on big data. *IEEE Access* **2020**, *8*, 54776–54788. [CrossRef]
32. Fang, M.; Yu, Y.; Zhang, W.; Wu, H.; Deng, M.; Fang, J. High performance computing of fast independent component analysis for hyperspectral image dimensionality reduction on MIC-based clusters. In Proceedings of the 2015 44th International Conference on Parallel Processing Workshops, Beijing, China, 1–4 September 2015; pp. 138–145.
33. Bharadiya, J.P. A Tutorial on Principal Component Analysis for Dimensionality Reduction in Machine Learning. *Int. J. Innov. Sci. Res. Technol.* **2023**, *8*, 2028–2032.

34. Kabir, M.F.; Chen, T.; Ludwig, S.A. A performance analysis of dimensionality reduction algorithms in machine learning models for cancer prediction. *Healthc. Anal.* **2023**, *3*, 100125. [CrossRef]
35. Ma, Y.; Shan, C.; Gao, J.; Chen, H. A novel method for state of health estimation of lithium-ion batteries based on improved LSTM and health indicators extraction. *Energy* **2022**, *251*, 123973. [CrossRef]
36. Goldberger, J.; Hinton, G.E.; Roweis, S.; Salakhutdinov, R.R. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems*; ACM: New York, NY, USA, 2004; Volume 17.
37. Singh-Miller, N.; Collins, M.; Hazen, T.J. Dimensionality reduction for speech recognition using neighborhood components analysis. In Proceedings of the Eighth Annual Conference of the International Speech Communication Association, Antwerp, Belgium, 27–31 August 2007.
38. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
39. Zhao, S.; Chen, X.A.; Wu, J.; Wang, Y.G. Mixture extreme learning machine algorithm for robust regression. *Knowl.-Based Syst.* **2023**, *280*, 111033. [CrossRef]
40. Alswaidan, N.; Menai, M.E.B. Hybrid feature model for emotion recognition in Arabic text. *IEEE Access* **2020**, *8*, 37843–37854. [CrossRef]
41. Guo, Y.; Song, Q.; Jiang, M.; Guo, Y.; Xu, P.; Zhang, Y.; Fu, C.C.; Fang, Q.; Zeng, M.; Yao, X. Histological subtypes classification of lung cancers on CT images using 3D deep learning and radiomics. *Acad. Radiol.* **2021**, *28*, e258–e266. [CrossRef]
42. Kolter, J.; Johnson, M. REDD: A Public Data Set for Energy Disaggregation Research. *Artif. Intell.* **2011**, *25*, 59–62.
43. Murray, D.; Stankovic, L.; Stankovic, V. An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study. *Sci. Data* **2017**, *4*, 160122. [CrossRef] [PubMed]
44. Kuhn, M.; Johnson, K. *Applied Predictive Modeling*; Springer: New York, NY, USA, 2013; Volume 26.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.