Application of Neighborhood Component Analysis for Enhancing Load Recognition in Home Energy Management Systems

Thales W. Cabral, Fernando B. Neto, Eduardo R. de Lima, Gustavo Fraidenraich e Luís Geraldo P. Meloni

***Abstract*— The increasing residential demand for electricity has a direct effect on the balance between human activities and the environment. Technological solutions like the Home Energy Management System (HEMS) are essential for sustainable energy consumption. This paper proposes novel approaches to load recognition in HEMS, which main contributions are enhanced performance through the application of Neighborhood Component Analysis (NCA) jointly with (i) the optimized Support Vector Machine (SVM) achieving 96.60% accuracy, 96.49% F_1, and 0.9404 Kappa Index, respectively; (ii) the optimized k-nearest Neighbors (k-NN) achieving 95.89% accuracy, 95.87% F_1, and 0.9281 Kappa Index, respectively; (iii) the optimized Extreme Gradient Boosting (XGBoost) achieving 96.03% accuracy, 95.81% F_1, and 0.9302 Kappa Index, respectively; and (iv) improved results for training and inference times.**

***Keywords*— Neighborhood Component Analysis, NCA, Machine learning, HEMS, Load Recognition.**

I. Introduction

It is an undeniable fact that the rising electricity demand represents a global challenge. According to [1], the residential sector stands out as a significant consumer of energy, directly impacting the balance between human activities and environmental sustainability. In this sense, researchers have developed technologies to reduce electricity expenditure via efficient energy management. Consequently, solutions such as the Home Energy Management System (HEMS) are essential for sustainable energy consumption.

As per [2], HEMS can individually monitor the energy usage of household appliances. This system also can send spending alerts to the end user, identify device malfunctions, and provide consumption reports. Furthermore, a modern HEMS includes additional functions such as load disaggregation [3] and load recognition [4]. According to [5], load recognition is the task of determining the type of appliance in operation. Furthermore, load recognition improves load disaggregation techniques by boosting the precise identification of individual appliances after the disaggregation process. As reported by [6], load recognition also plays a crucial function in creating appliance databases by interpreting electrical signals and refining the process. Here, it is relevant to mention that the motivation of the present work is to enhance the load recognition process.

There are several strategies for load recognition in the literature. However, state-of-the-art approaches combine feature extraction techniques with Machine Learning (ML) models for this task. In [7], the authors use the Gramian Angular Difference Field (GADF) in the feature extraction stage and apply a Convolutional Neural Network (CNN) to identify household appliances, achieving 83.33% accuracy. The reference [8] extracts the patterns with Stockwell transform and employs Vector Projection Classification (VPC) to recognize devices, reaching 90.00% accuracy. The method presented by reference [9] performs 95.40% accuracy using operating patterns of equipment and Support Vector Machine (SVM) for the load recognition task. However, there are some unexplored gaps in the literature, such as methods that employ feature extraction techniques to improve separability between classes and appropriate ML models, able to enhance performance.

This article is an extension of the work [4], where we advanced the state-of-the-art load recognition approaches by applying Neighborhood Component Analysis (NCA), jointly with the Support Vector Machine (SVM), k-Nearest Neighbors (k-NN), and Extreme Gradient Boosting (XGBoost) models. This work presents the main contributions: (i) the first application of the NCA technique with the optimized SVM for load recognition, outperforming the reference work [4], achieving superior values of accuracy, F_1, and K.I. – 96.60%, 96.49%, and 0.9404, respectively; (ii) the pioneering combination of the NCA and optimized k-NN in a load recognition task, also surpassing the performance of the work [4], reaching higher values of accuracy, F_1, and K.I. – 95.89%, 95.87%, and 0.9281, respectively; (iii) the first use of the NCA with the optimized XGBoost for load recognition, winning the work [10], yielding higher values of accuracy, F_1, and K.I. – 96.03%, 95.81%, and 0.9302, respectively; and (iv) enhanced overall performance for training and inference times.

It is relevant to mention that the obtained values for the evaluation metrics are higher than the previously cited works. In addition, our proposed approaches, NCA with optimmized SVM and NCA with optimmized XGBoost, have faster training times when compared to competing approaches. In this case, NCA with optimmized XGBoost is approximately 8 times faster than its rivals. For the inference time, NCA with

Thales W. Cabral, Department of Communications, University of Campinas, Campinas-SP, e-mail: t264377@dac.unicamp.br, Fernando Bauer, Copel Distribution S.A., Curitiba-PR, e-mail: fernando.bauer@copel.com; Eduardo R. de Lima, Instituto de Pesquisas Eldorado, e-mail: eduardo.lima@eldorado.org.br, Gustavo Fraidenraich, Department of Communications, University of Campinas, Campinas-SP, e-mail: gf@decom.fee.unicamp.br, Luís Geraldo P. Meloni, Department of Communications, University of Campinas, Campinas-SP, e-mail: meloni@unicamp.br. The authors thank the Eldorado Research Institute and Copel Distribution S.A. for funding this research within the scope of the ANEEL P&D project PD-02866-0508/2019. This study was partially financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior – Brasil (CAPES) – Finance Code 001.

optimmized XGBoost is about 310 times faster than its adversaries. Finally, this work is part of a research project titled “Open Middleware and Energy Management System for the Home of the Future.” It involves collaborative efforts among the University of Campinas, the Eldorado Research Institute, and the Brazilian energy provider Companhia Paranaense de Energia (COPEL).

The remainder of the paper is organized as follows: Section II presents the proposed approach for load recognition. Section III presents the performance metrics. Section IV discusses the results. Section V outlines the main conclusions.

II. Proposed System for Enhancing Load Recognition

Figure 1 illustrates a typical HEMS system with smart outlets and a controller. The smart outlets can collect several data, such as the active and reactive powers, power factor, voltage, and current of household appliances. This information is sent to the controller via communication protocols like Wi-SUN, Wi-Fi, or Bluetooth. A modern HEMS can send energy consumption alerts to the client. Furthermore, a HEMS with ML algorithms can process the data locally or send it to the cloud. Then, after reading the data, the load recognition system starts to work.

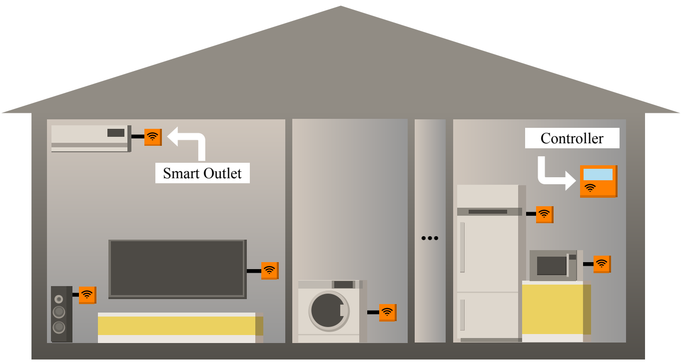

Fig. 1. Typical Home Energy Management System (adapted from [6]).

Figure 2 illustrates a short flowchart of the proposed load recognition system using the active power feeds the data processing block. At this point, the system detects the moments of activity of household appliances using the ON/OFF state detection described in [11]. Although event detection is not the focus of this manuscript, such event detection uses level 1 detail coefficients from the Wavelet Transform, as per [11]. After detection, the system transforms the selected segments into images and rearranges the pixels, respectively, according to [4]. As per reference [4], we transform all the rows of the image into a column vector $\mathbf{x}$ of size J. Each vector $\mathbf{x}$ represents an image. Then, for I-generated images, this process generates a set of $\mathbf{x}$ vectors. At the end of the data processing block, we have a dataset represented by the matrix $\mathbf{X}_{J\times I}$, which we name dataset $\mathbf{X}$. In the sequel, for the feature extraction, we partition $\mathbf{X}$ into training set $\mathbf{X}_{\text{(train)}}$ and test set $\mathbf{X}_{\text{(test)}}$.

For the next block in Figure 2, we employ NCA as per Algorithm 1 in the feature extraction methodology. At this stage, the algorithm requires baseline conditions such as the initial number of components and the decision threshold. There are no restrictions regarding the choice of a value for the initial number of components, as Algorithm 1 employs an optimized search to define the optimal number of components. For the threshold (ξ), we adopted the suggestions as per reference [4]. It is relevant to mention that the system obtains the optimal number of components by comparing the Cumulative Explained Variance (CEV) for the k-th component with ξ. Once done, Algorithm 1 generates the transformed data with reduced dimensionality, $\mathbf{X}^{(r)}_{\text{(train,NCA)}}$ and $\mathbf{X}^{(r)}_{\text{(test,NCA)}}$.

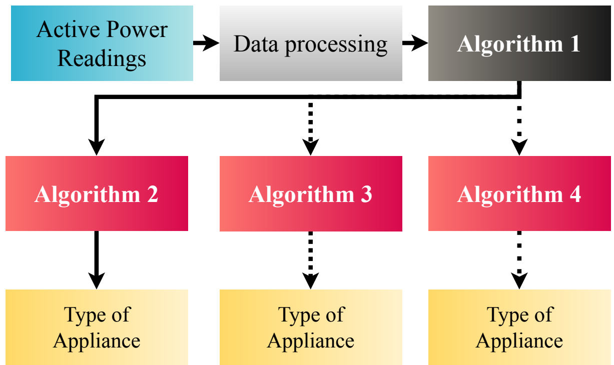

Fig. 2. Summary of the Proposed Load Recognition System.

Algorithm 1 Feature extraction using neighborhood component analysis and cumulative explained variance.

Input: $\mathbf{X}_{\text{(train)}}$ training data and the $\mathbf{X}_{\text{(test)}}$ test data, initial number of components (η), threshold (ξ)

Output: training set $\mathbf{X}^{(r)}_{\text{(train,NCA)}}$ and testing set $\mathbf{X}^{(r)}_{\text{(test,NCA)}}$

1: first method:
Train the NCA with $\mathbf{X}_{\text{(train)}}$, with η initial components and find the transformed data $\mathbf{X}^{(\eta)}_{\text{(train,NCA)}}$

2: second method:
Estimate the covariance matrix $\mathbf{C}$ from the $\mathbf{X}^{(\eta)}_{\text{(train,NCA)}}$, compute the eigenvalues λ_i via $\mathbf{C} = \mathbf{\Lambda}\, diag(\lambda_1, \lambda_2, \cdots, \lambda_\eta)\, \mathbf{\Lambda}^{-1}$, and order the eigenvalues in descending sequence. Here, $\mathbf{\Lambda}$ is the eigenvectors matrix and $diag$ refers to the diagonal matrix with the eigenvalues.

3: third method:
Obtain the optimized number r of components via CEV:
  Create variable r and assign zero value
  Compute $\text{CEV}_k = \sum_{j=1}^{k} \frac{\lambda_j}{\sum_{i=1}^{\eta} \lambda_i}$
    if $\text{CEV}_k \geq \xi$
      $r \leftarrow$ number of k-th component
    end if

4: fourth method:
Re-train the NCA with $X_{\text{(train)}}$ using only r components. Next, apply NCA to generate transformed data $\mathbf{X}^{(r)}_{\text{(train,NCA)}}$ and $\mathbf{X}^{(r)}_{\text{(test,NCA)}}$.
return $\mathbf{X}^{(r)}_{\text{(train,NCA)}}$ and $\mathbf{X}^{(r)}_{\text{(test,NCA)}}$

For the subsequent stage, as shown in Figure 2, our approach proposes three different algorithms for enhancing the classifiers. In this case, algorithms 2, 3 and 4 are related to the load identification as per SVM, k-NN, and XGBoost models, respectively. However, it is necessary to present the key parameters of these models before describing the proposed algorithms.

SVM is a flexible ML architecture whose outstanding feature is its ability to handle non-linearly separable data. In our context, the hyperparameter C is a regularization hyperparameter that balances the creation of a well-defined decision boundary to separate the classes and the classification error minimization during training. The hyperparameter γ determines the extent of the kernel function's influence [12]. Our system finds the optimal values of C and γ through hyperparameter search based on Grid Search with K-Fold Cross-Validation. Considering such parameters, Algorithm 2 presents an approach to identify the type of appliance in operation via the optimized SVM.

Algorithm 2 Load identification using the optimized SVM

Input: Possible values for the hyperparameter C $(C_1, C_2, \cdots, C_n)$, possible values for the hyperparameter γ $(\gamma_1, \gamma_2, \cdots, \gamma_n)$, $\mathbf{X}^{(r)}_{(\text{train,NCA})}$, $\mathbf{X}^{(r)}_{(\text{test,NCA})}$, number of folds (K)
Output: Type of Appliance
1: first method:
  Load the possible values for both hyperparameters: $C_1, C_2, \cdots, C_n$ and $\gamma_1, \gamma_2, \cdots, \gamma_n$
2: second method:
  Apply Grid Search with K-fold Cross-Validation
   Split $\mathbf{X}^{(r)}_{(\text{train,NCA})}$ in K folds
   Train the model across each fold
   Measure mean accuracy
   Allocates the mean accuracy for current hyperparameters
   Select the best hyperparameters via highest mean accuracy: $C_{(\text{optimal})}$ and $\gamma_{(\text{optimal})}$
3: third method:
  Training the model with $C_{(\text{optimal})}$ and $\gamma_{(\text{optimal})}$
4: fourth method:
  Testing the optimized SVM using $\mathbf{X}^{(r)}_{(\text{test,NCA})}$
  return Type of Appliance

For a given input datum, the k-NN algorithm discerns the k nearest data points within the confines of the training dataset. Subsequently, the algorithm ascertains the classification outcome by conducting a majority vote among the labels attributed to these proximate entities [12]. However, the effectiveness of the algorithm depends on the choice of k. Thus, a suitable choice for this hyperparameter is essential. To achieve this objective, the proposed approach employs the Grid Search combined with K-Fold Cross-Validation to ascertain the optimal value for k. Considering these factors, Algorithm 3 presents a method to identify the type of appliance in function via the optimized k-NN.

XGBoost is an ensemble method; thus, it produces decision trees and obtains predictions sequentially [13]. For this algorithm, the system needs to determine the optimal choice

Algorithm 3 Load identification using the optimized k-NN

Input: Possible values for the hyperparameter k $(k_1, k_2, \cdots, k_n)$, $\mathbf{X}^{(r)}_{(\text{train,NCA})}$, $\mathbf{X}^{(r)}_{(\text{test,NCA})}$, number of folds (K)
Output: Type of Appliance
1: first method:
  Load the possible values for hyperparameter k: $k_1, k_2, \cdots, k_n$
2: second method:
  Apply Grid Search with K-fold Cross-Validation
   Split $\mathbf{X}^{(r)}_{(\text{train,NCA})}$ in K folds
   Train the model across each fold
   Measure mean accuracy
   Allocates the mean accuracy for current hyperparameters
   Select the best hyperparameter via highest mean accuracy: $k_{(\text{optimal})}$
3: third method:
  Training the model with $k_{(\text{optimal})}$
4: fourth method:
  Testing the optimized k-NN using $\mathbf{X}^{(r)}_{(\text{test,NCA})}$
  return Type of Appliance

of hyperparameters concerning maximum depth and number of estimators. For this purpose, the proposed method applies the hyperparameter search via Grid Search with K-Fold Cross-Validation. Considering these criteria, Algorithm 4 presents an approach to identify the type of appliance in operation through the optimized XGBoost.

Algorithm 4 Load identification using the optimized XGBoost

Input: Possible values for the 'max depth' hyperparameter u $(u_1, u_2, \cdots, u_n)$, values for the 'number of estimators' hyperparameter w $(w_1, w_2, \cdots, w_n)$, $\mathbf{X}^{(r)}_{(\text{train,NCA})}$, $\mathbf{X}^{(r)}_{(\text{test,NCA})}$, number of folds (K)
Output: Type of Appliance
1: first method:
  Load the possible values for both hyperparameters: $u_1, u_2, \cdots, u_n$ and $w_1, w_2, \cdots, w_n$
2: second method:
  Apply Grid Search with K-fold Cross-Validation
   Split $\mathbf{X}^{(r)}_{(\text{train,NCA})}$ in K folds
   Train the model across each fold
   Measure mean accuracy
   Allocates the mean accuracy for current hyperparameters
   Select the best hyperparameters via highest mean accuracy: $u_{(\text{optimal})}$ and $w_{(\text{optimal})}$
3: third method:
  Training the model with $u_{(\text{optimal})}$ and $w_{(\text{optimal})}$
4: fourth method:
  Testing the optimized XGBoost using $\mathbf{X}^{(r)}_{(\text{test,NCA})}$
  return Type of Appliance

III. Performance Metrics

The current work uses the metrics of accuracy, weighted average F1-Score (F_1), and Kappa Index (K.I.) to analyze the performance of the proposed approach. It is essential to mention that such metrics depend on indicators: true-positive (TP), true-negative (TN), false-positive (FP), and false-negative (FN).

Accuracy evaluates the global performance of the models [12]. In this paper, accuracy is expressed according to

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \,. \tag{1}$$

Due to the characteristics of the appliances, each device can generate a varying number of events. Thus, it is necessary to consider this effect for a fair analysis. For this purpose, this study uses the F_1 to incorporate this impact into the F1-Score metric [12]. So, F_1 is defined as per

$$\text{F}_1 = \frac{1}{Q} \sum q \times \left[\frac{2 \times \text{TP}}{2 \times \text{TP} + 1 \times (\text{FN} + \text{FP})} \right] , \tag{2}$$

where q is the size of the set of instances of a class, and Q is the size of the data set.

Furthermore, it is necessary to verify the concordance of the system concerning the predicted value and the expected value, which is addressed by K.I. [4]. This statistic metric can vary within a range from -1 to 1. A K.I. value of -1 represents no agreement, 0 indicates chance agreement, and 1 means complete agreement. This document defines K.I. as per

$$\text{K.I.} = \frac{2 \times (\text{TP} \times \text{TN} - \text{FN} \times \text{FP})}{(\text{TP} + \text{FP}) \times (\text{FP} + \text{TN}) + (\text{TP} + \text{FN}) \times (\text{FN} + \text{TN})} \,. \tag{3}$$

IV. Results and Discussions

This work uses real-world data from the REDD public database [14]. This dataset provides active power measurements at a frequency of 1/3 Hz. The residence chosen from this database includes several devices such as a dishwasher, washer-dryer, heat pump, microwave, oven, kitchen oven, refrigerator, bathroom Ground Fault Interrupters (GFI) outlet, stove, lighting, and an unknown device.

It is worth mentioning that after the identification process regarding the operational state of the equipment (either on or off), the proposed system transforms the identified segments into images possessing a resolution of 32×32 pixels, generating a total of 4609 images. From this total, the system reserves 80% for training and 20% for testing, whereby only the training data participates in the Grid Search with K-fold Cross-Validation procedures. In addition, according to [4], we have adopted ξ = 0.99. For a η = 100, Algorithm 1 determined the optimal number of r components equal to 25. Finally, it is necessary to point out that all the results reported in Tables from III to VI are average values from 50 iterations.

A. Comparative analysis of k-NN-based approaches

In this scenario, we established the possible values for the hyperparameter k $(k_1, k_2, \cdots, k_n)$ as per reference [4] to guarantee a direct comparison. Hence, the options for k range from 1 to 10, with an increment of 1. We have kept the value of K=10 for the hyperparameter search. Then, we run the Algorithm 3 for the k-NN approach (NCA-k-NN). Here, the Algorithm appointed the optimal value for k equal to 1, i.e., $k_{(\text{optimal})} = 1$.

Table I compares the performances of the NCA-k-NN and PCA-k-NN. Once again, all metrics at Table I demonstrate a performance enhancement with the NCA technique. The NCA-k-NN achieved improvements of 2.40% in accuracy, 2.42% in F_1, and 3.93% in K.I., compared to the PCA-k-NN.

TABLE I
Performance of approaches using the k-NN model

Method	**Accuracy**	$\mathbf{F_1}$	**K.I.**
PCA-k-NN [4]	93.49%	93.45%	0.8916
NCA-k-NN	95.89%	95.87%	0.9281

Table II presents the training and inference times associated with the NCA-k-NN and PCA-k-NN. In this case, the NCA-k-NN pair takes two times more training time than the competing pair. However, for both PCA-k-NN and NCA-k-NN, the training time is ultra short – 0.001 s and 0.002 s, respectively.

TABLE II
Training and inference times measured in seconds

Method	**Training time**	**Inference time**
PCA-k-NN [4]	0.001	0.168
NCA-k-NN	0.002	0.150

B. Comparative analysis of SVM-based approaches

To ensure a fair comparison, we have defined the possible values for the hyperparameters C $(C_1, C_2, \cdots, C_n)$ and γ $(\gamma_1, \gamma_2, \cdots, \gamma_n)$ according to reference [4]. Consequently, the possible values of C are 1, 10, 100, and 1000. So, the possible values of γ are 1, 0.1, and 0.001. Furthermore, we have maintained the value of K=10 for the search of hyperparameters. When running Algorithm 2 for the SVM approach (NCA-SVM), the system finds the optimal values $C_{(\text{optimal})}$ = 1000 and $\gamma_{(\text{optimal})}$ = 0.001.

Table III compares the proposed method, the NCA-SVM pair, with the competing approach [4], composed of the PCA-SVM pair. The result in Table III indicates an improvement in all the metrics associated with the NCA-PCA pair, suggesting that the proposed method has an evident competitive advantage. In this scenario, the NCA-SVM pair achieved the highest accuracy, F_1, and K.I. – 96.60%, 96.49%, and 0.9404, respectively – exhibiting enhanced performance when using NCA.

TABLE III
Performance of approaches using the SVM model

Method	**Accuracy**	$\mathbf{F_1}$	**K.I.**
PCA-SVM [4]	96.31%	96.36%	0.9381
NCA-SVM	96.60%	96.49%	0.9404

Table IV shows the training and inference times of the proposed approach, the NCA-SVM pair, and the competing method [4], consisting of the PCA-SVM pair. In this case, the NCA-SVM pair saves approximately 5.39% of the training time compared to the PCA-SVM pair, ensuring that the proposed approach is faster than the competitor.

TABLE IV
TRAINING AND INFERENCE TIMES MEASURED IN SECONDS

Method	Training time	Inference time
PCA-SVM [4]	0.167	0.075
NCA-SVM	0.158	0.048

C. Comparative analysis of XGBoost-based approaches

In this scenario, this study compares the strategies based on XGBoost present in the work [10]. Then, for a proper match with reference [10], we employ the same sets of possible hyperparameter values. Consequently, the possible values for the max depth u $(u_1, u_2, \cdots, u_n)$ are within the interval from 10 to 100, with the step size for the search of 10. We also apply the same interval and step size in the search for the number of estimators w $(w_1, w_2, \cdots, w_n)$.

Table V summarizes the performance of various approaches involving the XGBoost architecture. In this context, we can extract useful information about these results. The minimum difference in performance lies between the PCA-XGBoost and NCA-XGBoost strategies, where the NCA-XGBoost strategy demonstrates a slight edge, yielding a 0.04% improvement in accuracy and 0.06% increase in K.I. Conversely, the most noteworthy performance gap is between the LLE-XGBoost and NCA-XGBoost. In this case, the NCA-XGBoost approach has percentage advantages in terms of accuracy, F_1, and K.I. – 2.00%, 2.14%, and 3.74%, respectively.

TABLE V
PERFORMANCE OF APPROACHES USING THE XGBOOST MODEL

Method	Accuracy	F_1	K.I.
LLE-XGBoost [10]	94,03%	93.67%	0.8954
ICA-XGBoost [10]	95.77%	95.57%	0.9257
PCA-XGBoost [10]	95.99%	95.81%	0.9296
NCA-XGBoost	96.03%	95.81%	0.9302

Table VI shows the training and inference times for the XGBoost approaches cited in this manuscript. In this scenario, the slightest performance advantage lies between the LLE-XGBoost and NCA-XGBoost methods. The NCA-XGBoost strategy has a training time of 1.84 times faster than that of the LLE-XGBoost, and its inference time is 193.81 times quicker than the LLE-XGBoost. On the other hand, the highest difference in performance is between the PCA-XGBoost and NCA-XGBoost strategies. The NCA-XGBoost approach has training and inference times faster than the PCA-XGBoost – 8.10 and 310.82 times faster, respectively.

TABLE VI
TRAINING AND INFERENCE TIMES MEASURED IN SECONDS

Method	Training time	Inference time
PCA-XGBoost [10]	10.176	3.419
ICA-XGBoost [10]	6.806	2.541
LLE-XGBoost [10]	2.314	2.132
NCA-XGBoost	1.256	0.011

V. CONCLUSION

As an evolution of the work [4], this article proposes novel approaches to load recognition in HEMS. Among the proposed innovations are the application NCA jointly with the optimized SVM, with the optimized k-NN, with the optimized XGBoost, and also enhanced results for training and inference times. For all the proposed approaches, i.e., NCA-SVM, NCA-k-NN, and NCA-XGBoost, there is a performance improvement when compared to adversarial methods. Regarding this manuscript, the NCA-SVM pair reaches the highest accuracy, F_1, and K.I. – 96.60%, 96.49%, and 0.9404, respectively. Concerning performance gains, the NCA-k-NN pair yields the highest boosts in terms of accuracy, F_1, and K.I. – 2.40%, 2.42%, and 3.93%, respectively. Nevertheless, the NCA-XGBoost has the best-enhanced performance regarding the efficiency in training and inference times – 8.10 and 310.82 times faster than the rival, respectively – surpassing the adversarial method. The results demonstrate that the joint use of the NCA with ML models is a more robust alternative for load recognition.

REFERENCES

[1] T.-Y. Kim and S.-B. Cho, "Predicting residential energy consumption using cnn-lstm neural networks," *Energy*, vol. 182, pp. 72–81, 2019.

[2] L. L. Motta, L. C. Ferreira, T. W. Cabral, D. A. Lemes, G. d. S. Cardoso, A. Borchardt, P. Cardieri, G. Fraidenraich, E. R. de Lima, F. B. Neto, *et al.*, "General overview and proof of concept of a smart home energy management system architecture," *Electronics*, vol. 12, no. 21, p. 4453, 2023.

[3] D. A. M. Lemes, T. W. Cabral, G. Fraidenraich, L. G. P. Meloni, E. R. De Lima, and F. B. Neto, "Load disaggregation based on time window for hems application," *IEEE Access*, vol. 9, pp. 70746–70757, 2021.

[4] T. W. Cabral, D. A. M. Lemes, G. Fraidenraich, F. B. Neto, E. R. de Lima, and L. G. P. Meloni, "High-reliability load recognition in home energy management systems," *IEEE Access*, vol. 11, pp. 31244–31261, 2023.

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

[6] T. W. Cabral, F. B. Neto, E. R. d. Lima, G. Fraidenraich, and L. G. Meloni, "Load recognition in home energy management systems based on neighborhood components analysis and regularized extreme learning machine," *Sensors*, vol. 24, no. 7, p. 2274, 2024.

[7] L. Matindife, Y. Sun, and Z. Wang, "Image-based mains signal disaggregation and load recognition," *Complex & Intelligent Systems*, vol. 7, no. 2, pp. 901–927, 2021.

[8] V. Borin, C. Barriquello, and A. Campos, "Approach for home appliance recognition using vector projection length and Stockwell transform," *Electronics Letters*, vol. 51, no. 24, pp. 2035–2037, 2015.

[9] S. M. Qaisar and F. Alsharif, "Event-driven system for proficient load recognition by interpreting the smart meter data," *Procedia Computer Science*, vol. 168, pp. 210–216, 2020.

[10] T. W. Cabral, D. A. M. Lemes, F. B. Neto, E. R. de Lima, G. Fraidenraich, and L. G. P. Meloni, "Reconhecimento de cargas em sistemas residenciais de gestão de energia por meio de algoritmos baseados em árvores de decisão," in *XLI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2023)*, (São José dos Campos, São Paulo), SBrT, 2023.

[11] D. A. Lemes, T. W. Cabral, L. L. Motta, G. Fraidenraich, E. R. de Lima, F. B. Neto, and L. G. Meloni, "Low runtime approach for fault detection for refrigeration systems in smart homes using wavelet transform," *IEEE Transactions on Consumer Electronics*, 2023.

[12] L. A. Demidova, "Two-stage hybrid data classifiers based on svm and knn algorithms," *Symmetry*, vol. 13, no. 4, p. 615, 2021.

[13] D. Zhang, L. Qian, B. Mao, C. Huang, B. Huang, and Y. Si, "A data-driven design for fault detection of wind turbines using random forests and xgboost," *Ieee Access*, vol. 6, pp. 21020–21031, 2018.

[14] J. Kolter and M. Johnson, "Redd: A public data set for energy disaggregation research," *Artif. Intell.*, vol. 25, 01 2011.