Improving the Identification of Residential Loads Using Deep Transfer Learning Feature Extraction: A Comparative Analysis

J. S. Corrêa[1,*], D. L. Cavalca[2,†] and R. A. S. Fernandes[1,‡]
[1]Department of Electrical and Computer Engineering, University of São Paulo, São Carlos, Brazil
[2]Graduate Program in Computer Science, Federal University of São Carlos, São Carlos, Brazil
Email: *juan.salin@hotmail.com, †diegoluizcavalca@gmail.com, ‡ricardo.asf@usp.br

2025 International Joint Conference on Neural Networks (IJCNN) | 979-8-3315-1042-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/IJCNN64981.2025.11229208

***Abstract*—The growing need for home energy management systems to demand response programs has sought solutions that allow the identification of residential loads, mainly based on nonintrusive load monitoring (NILM). However, when performed on data acquired by meters with low sampling rates, this identification is a nontrivial and challenging task. In this sense, the present paper aims to analyze in a comparative way the use of deep transfer learning as a procedure to extract features. For this purpose, the time-series representing the electricity consumption were firstly windowed. Next, each window was transformed into a bidimensional image through the recurrence plot technique. Thus, pre-trained deep neural network models of five different architectures were fine tuned to the NILM domain. Subsequently, the last layers of these models were removed to obtain a feature array. Therefore, the set of feature arrays was used to train and validate shallow machine learning algorithms (Support Vector Machine, Multilayer Perceptron and Extreme Gradient Boosting). This comparative analysis demonstrated the robustness of a deep transfer learning feature extraction based on the ResNet architecture when considering the Extreme Gradient Boosting as the classifier responsible to identify the loads, reaching an average f1-score of 81% and a standard deviation of 1.8%.**

***Index Terms*—Deep feature extraction, deep learning, machine learning, nonintrusive load monitoring, recurrence plot.**

I. Introduction

Nowadays, one of the world's biggest challenges is the fight against global warming and the increase in the planet's temperature caused by carbon gas emissions that accelerate the process. However, 27% of emissions come from the energy sector, where 30% are from buildings. Therefore, approaches that can reduce household energy consumption are of great relevance, especially considering that there is a growing trend in demand in this area [1]–[3]. In the context of home energy management systems, using a large quantity of monitoring devices (like smart sockets) can be expensive and unfeasible to the task of load identification [4]. For this reason, nonintrusive load monitoring (NILM) research area aims to disaggregate the whole home consumption using only measurements acquired in the main panel [5].

Nevertheless, NILM approaches face some issues in the residential energy sector, such as: (i) the operating behavior and consumption of the loads can change based on model and manufacturer; (ii) the datasets can vary in function of quantity of monitoring devices and their sampling rates; (iii) the identification of loads in real-time to allow embedded hardware solutions; (iv) the ability to generalize the load's identification in different scenarios; and (v) the consumption overlaps, as some loads can be powered on in the same time interval.

The research in NILM presented high accurate results when dealing with measurements acquired in high frequency, i.e., meters with sampling rates greater than 200 samples per cycle. However, this kind of approach is impractical in many real-world applications due to the availability of low cost commercial meters able to store and process this amount of data in real-time [4]. On the other hand, approaches based on measurements acquired in lower sampling rates still have room for improvement. Furthermore, as explained in [5], [6], the power signature, i.e., the behavior of each appliance can be divided into 4 categories: (i) binary state (ON/OFF); (ii) finite and discrete operating states; (iii) continuous operating, in which the consumption changes in a continuous scale; and (iv) cyclic operating, in which the consumption presents a periodic nature.

As in other research areas, the use of deep learning is also a trend in NILM. For this reason, some approaches have been proposed in this way. In [4], aiming to mitigate the problem of data unbalance, the authors proposed an approach based on Long-Short Term Memory to improve feature representation and learn usage patterns in REDD (The Reference Energy Disaggregation Data Set) dataset. Using the UK-DALE (The United Kingdom – Domestic Appliance-Level Electricity) dataset, in [7], the authors introduced an attention mechanism to update weight distribution to a CNN (Convolutional Neural Network), highlighting the most relevant features and reducing noise.

Langevin et. al. [8] proposed a variational autoencoder framework using the UK-DALE and REFIT datasets for disaggregation task. In [9], the authors present a novel model named as MUSENILM, that applies multi-scale attention mechanism on UK-DALE and REDD datasets to extract temporal features. The authors of [10] proposed a multi-event identification methodology with a multi-head self-attention mechanism, ex-

This work was supported by the São Paulo Research Foundation (FAPESP) [Grant Numbers 2022/00750-9, 2019/15192-9].

tracting contextual information of events aiming to solve the difficulty in identifying multiple variable-length events in one window.

Following the aforementioned context, this paper proposes a framework based on the transformation of power consumption time-series into bidimensional images using the Recurrence Plot (RP) technique, extracting features automatically using deep transfer learning and performing the load identification afterwards. Thus, five different architectures with models pre-trained for the ImageNet domain were used as feature extractors. In order to compare and analyze the behavior of each deep transfer learning feature extraction technique, three shallow machine learning algorithms were considered, that are: Support Vector Machine (SVM), Multilayer Perceptron (MLP) and Extreme Gradient Boosting (XGBoost).

The remainder of the paper was organized as follows. Section II presents a brief review on transfer learning strategies commonly applied in the NILM context. Section III presents an overview of each deep neural network architecture to be used in the deep transfer learning feature extraction stage. Next, the proposed approach used to analyze the feature extraction stage is detailed in Section IV. The results were presented, compared and discussed in Section V. At last, the conclusions were drawn in Section VI.

II. Brief Review on Transfer Learning in NILM

According to [11], the process of solving a task A by adapting or reusing a model that was trained for a task B is called transfer learning. Typically, the transfer learning process is applied in two ways, namely through fine-tuning or by using the original model as a feature extractor. In the first approach, a controlled retraining of an original model is performed, where the goal is to slightly modify the weights of this model so that it can solve a different task. Depending on the amount of data available, one can choose to change the weights of only a few or many layers of the network. In the second approach, pre-trained deep neural networks are also used as feature extractors to encode the images into useful descriptive representations, extracted directly from the output of an intermediate layer, resulting in an embedding (non-linear feature vector) of the input sample.

In [12], the authors proposed data transformations considering the UK-DALE e REDD datasets. They evaluated the behavior of a fine-tuning transfer learning strategy only for fridge. The temporal data (collected every 1 Hz) were transformed into 2D images through Gramian angular difference fields (GADF). These images were used as inputs to a CNN VGG16 pre-trained model. The proposed approach reached an average f1-score of 75%.

The feasibility of using transfer learning was also analyzed by [13], where the authors noted that only fully connected layers require fine-tuning optimization.

The study proposed by [14] offers transfer learning results, confirming the relevance and robustness of the selected features, which were learned on a proposed dataset and successfully transferred to a larger-scale dataset.

Finally, transfer learning was also applied to an approach based on the signature of the appliances (represented by their V-I trajectories) [15]. However, the effectiveness of the pre-trained model was demonstrated empirically, that is, there is no statistical proof of its accuracy.

It is important to note that the aforementioned literature corroborates the importance and effectiveness of transfer learning as a highly relevant stage for the load labeling task in NILM, providing valuable insights to improve the accuracy and generalization of energy consumption disaggregation models.

III. Deep Neural Network Architectures

As the approach proposed in this paper was focused in compare different deep neural networks to extract features and improve the identification of residential loads, next is presented an overview of the following architectures: VGG (Visual Geometry Group), ResNet (Residual Network), Inception, MobileNet and EfficientNet.

A. VGG

The VGG architecture was proposed by [16] as a CNN model to be used in computer vision applications. The key feature is its depth, with 16 or 19 layers, mainly composed of convolutional layers followed by max-pooling layers. Convolutional layers apply filters to the input image, extracting features at different spatial scales. On the other hand, max-pooling layers reduce the spatial dimensions of feature maps while keeping the most important information.

B. ResNet

This convolutional architecture was introduced by [17], being its main characteristic the use of residual connections, which allow information to be transmitted directly from one layer to another, bypassing the intermediate layers. A fundamental component in ResNet is the residual block that implements two 3×3 convolutional layers and a residual connection (identity). This component assists in mitigating the problem of gradient vanishing, which is common in deep neural networks.

C. Inception

It was proposed by [18] and stands out for its use of a combination of convolution kernel sizes. Its convolutional block is composed of four parallel paths, the first three of which use convolutions of different sizes to extract information from different spatial scales, while the fourth path employs a max-pooling layer followed by convolution to adjust the number of channels.

D. MobileNet

Aiming to reach a balance between precision and computational burden, the MobileNet was proposed by a Google's research team [19] to allow computer vision models to be executed in platforms with limited hardware resources, mainly smartphones and tablets. Using depthwise separable convolutions, this technique splits the default convolution process into a depthwise convolution (responsible to apply individual convolutions to each input signal) and a pointwise convolution (a 1×1 convolution to combine the results of these convolutions divided by channels). Regarding the other CNN components, MobileNet uses both normalization batch and ReLU (Rectified Linear Unit) activation functions in both layers.

E. EfficientNet

The EfficientNet was another algorithm proposed by a Google's research team [20] to establish an efficient architecture regarding resources, while still offering a good performance in deep learning tasks. The fundamental concept of the architecture is compound scaling, raising simultaneously and balancing the: (i) number of deep layers (depth); (ii) number of channels by layer (width); and (iii) the input image resolution.

IV. Proposed Approach

The proposed approach was implemented by considering the REDD, since it represents a benchmark dataset widely analyzed in the related literature. An overview of the proposed approach can be visualized in Fig. 1.

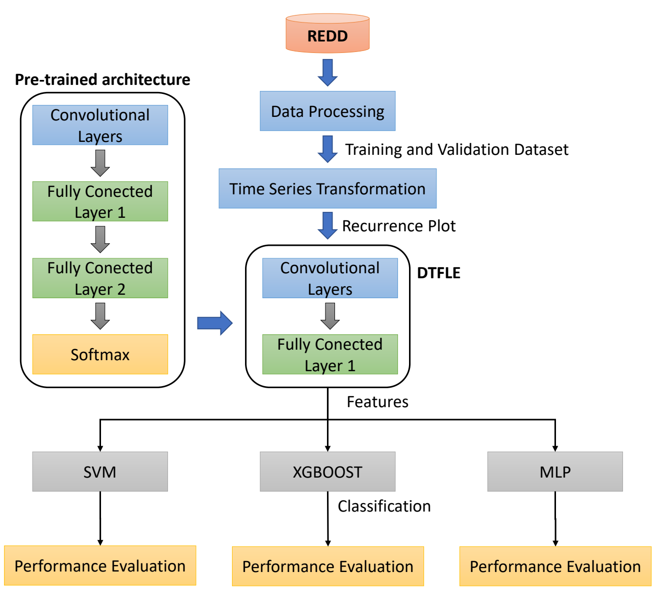

Fig. 1. Overview of the proposed approach.

A. Data processing

The REDD is divided into low and high frequency measurements acquired from six houses at The United States. The data used in this research was extracted from the house #3, since it is the most investigated in the literature. Also, only the low frequency data were considered, which were collected in a 1 Hz sample rate.

Thus, the resulting data were divided into training and validation sets as shown in Table I, following the same temporal interval (starting and ending dates) given in [12]. In addition, with the highest representation of energy consumption in the house, the loads chosen for this research were: fridge, dishwasher, washer dryer #1, washer dryer #2 and microwave.

TABLE I
Starting and ending dates of extracted data.

Subset	Start date	End date
Training	Apr 16th, 2011	May 16th, 2011
Validation	May 17th, 2011	May 30th, 2011

Next, the time-series were processed as a set of sliding windows. However, as small windows can result in loss of useful information to characterize certain loads, while large windows may include noise and/or redundant information.

In this sense, as the authors of [21] have evaluated both the sampling rate and the influence of fixed and variable window sizes in the REDD, we just parameterized the proposed approach using the same 2 seconds of sample rate and the window sizes presented in Table II.

TABLE II
Size of the sliding window determined for each load, according with [21].

Load	Window size (in seconds)
Fridge	720
Dishwasher	2,040
Washer dryer #1	90
Washer dryer #2	2,040
Microwave	900

From the time-series windowed, each window was labeled, allowing to obtain the training and validation sets. For this purpose, it was necessary to relate the measurements acquired from the main panel and the individual loads. This relationship was defined by using the timestamps, making possible to create a binary array with dimension equal to the number of loads (i.e., five). In this sense, the use of a binary array allows the labeling process to define more than one load per window, that is, the overlapping or simultaneous operation.

B. Time-series transformation

In order to overcome the challenges of nonlinear pattern identification for NILM time-series and due to the use of CNNs as techniques to extract features, the temporal windows were transformed into 2D images. More specifically, RPs were generated, being represented as a $L \times L$ matrix, where L represents the window size. This technique was proposed by [22] to demonstrate recurrences in a state-space trajectory, which can be expressed as:

$$R_{i,j} = \theta \left(\epsilon - \| \vec{x_i} - \vec{x_j} \| \right), \vec{x} \in \Re^m, \quad i,j = 1,2,\ldots,N \quad (1)$$

where N is the number of states $\vec{x_i}$; ϵ is the radius of the neighborhood (threshold) at the current state; $\|.\|$ is a neighborhood norm, usually the Euclidean norm; $\theta(.)$ is the Heaviside function; and m is the dimension of immersion.

Considering this definition, an RP can be analyzed as follows: (a) if $R_{ij} = 1$, the state is given as recurrent and a black pixel is set to the image matrix; or (b) if $R_{ij} = 0$, the state is given as non-recurrent and, consequently, a white pixel is set in the matrix.

C. Deep transfer learning feature extraction

In the NILM context, the deep transfer learning feature extraction was firstly proposed by [21], where the base deep learning algorithm was a CNN VGG pre-trained in the ImageNet database [23] domain. Due to the use of a model pre-trained in a different domain, the residential consumption domain needs to be presented to the model by means of transfer learning. For this reason, fine tuning strategy was employed.

In order to investigate the impact of different deep neural network architectures in extracting features from RPs, those mentioned in section III were considered. It is important to highlight that all of them are available in the TensorFlow open source library and their characteristics are presented in Table III.

TABLE III
CHARACTERISTICS OF ARCHITECTURES PRE-TRAINED IN THE IMAGENET DATABASE DOMAIN.

Model	Number of neurons	Depth
EfficientNetB0	5.3M	132
EfficientNetB1	7.9M	186
EfficientNetB2	9.2M	186
EfficientNetB3	12.3M	210
EfficientNetB4	19.5M	258
EfficientNetB5	30.6M	312
EfficientNetB6	43.3M	360
EfficientNetB7	66.7M	438
InceptionV3	23.9M	189
MobileNet	4.3M	55
MobileNetV2	3.5M	105
ResNet50	25.6M	107
ResNet50V2	25.6M	103
ResNet101	44.7M	209
ResNet101V2	44.7M	205
ResNet152	60.4M	311
ResNet152V2	60.4M	307
VGG19	143.7M	19

Based on these models, the transfer learning strategy adjusts the synaptic weights to the NILM context. After finalizing this process, the last classificatory layers of each model were removed to obtain a feature array instead of the data label, as presented in Fig. 1. This way, the set of feature arrays resulted from each model was used as input to shallow machine learning algorithms, as detailed next.

D. Identification of residential loads

From the feature arrays extracted using different deep neural network architectures, SVM, MLP and XGBoost models were trained and validated. It is worth mentioning that five binary classifiers were used, i.e., each one responsible for identifying a specific load.

1) MLP: It has been the most used machine learning algorithm in the NILM context, due to its high generalization capability and accurate classification. The MLP model used in this paper was composed of 10 hidden layers with ReLU activation function (selected to deal with the gradient vanishing problem [24]) and 1 output layer with softmax activation function. Also, the backpropagation algorithm was employed since it is commonly applied to the NILM research field.

2) SVM: It was considered due to its deterministic behavior. The algorithm aims to determine a hyperplane that separates the classes, looking for the minor distance between the sample and the hyperplane to create a separation margin. In this sense, the training stage of the algorithm consists in determining the optimal hyperplane [25]. However, due to its high computational cost, usually, a tolerance level needs to be defined [26]. Furthermore, keeping in mind that NILM data is not linearly separable, in this paper a radial basis function kernel was used.

3) XGBoost: It is a decision tree-based model that uses a gradient boosting strategy [27]. The decision trees focus on the iterative data segmentation in smaller and specific regions until reaching an enough size to be labeled. Due to the boosting, instead of constructing one tree, several trees are generated with small intermediate nodes in a sequential manner by using the extracted information of its progenitor (previous tree). Finally, the gradient boosting is a variation of the boosting algorithm in which a minimization function is adopted [28].

E. Performance evaluation metric

To avoid the bias that can be generated due to class majority, the performance metric adopted was the f1-score, that represents a relation between precision and recall, expressed as:

$$Precision = \frac{TP}{TP + FP}, \tag{2}$$

$$Recall = \frac{TP}{TP + FN}, \tag{3}$$

$$f1score = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \tag{4}$$

with TP and FP being respectively the true and false positives, while FN is the false negatives. Since the REDD is unbalanced, especially when dealing with loads such as microwaves, the use of f1-score to measure the performance of classifiers is an adequate practice in machine learning in order to mitigate the data bias.

V. RESULTS AND DISCUSSIONS

The proposed approach was completely implemented using Python 3 and the computational experiments were run on a desktop computer with Intel Core i7-7700 CPU (3.60GHz), 16GB DDR4 RAM and an NVIDIA Quadro M5000. Next, the overall analysis is presented and discussed, followed by the comparative analysis performed between the deep transfer learning feature extraction techniques.

A. Overall performance analysis

The overall results obtained by each shallow machine learning algorithm (SVM, MLP and XGBoost) can be verified in Table IV. Still, the results can be analyzed in terms of deep neural network model used to extract features from the RPs.

TABLE IV
PERFORMANCE OF EACH SHALLOW MACHINE LEARNING CLASSIFIER COMBINED WITH DIFFERENT DEEP FEATURE EXTRACTION TECHNIQUE.

Feature Extraction	**f1-score (%)**		
	MLP	**SVM**	**XGBOOST**
EfficientNetB0	74.6	65.4	73.0
EfficientNetB1	65.4	64.8	71.4
EfficientNetB2	71.6	65.4	70.0
EfficientNetB3	70.8	59.2	69.2
EfficientNetB4	78.4	**75.0**	75.4
EfficientNetB5	74.0	66.6	68.0
EfficientNetB6	81.0	65.0	81.0
EfficientNetB7	77.0	73.4	78.6
InceptionV3	74.2	74.2	75.8
MobileNet	74.4	71.8	76.8
MobileNetV2	73.0	69.4	71.8
ResNet101	**84.2**	73.6	82.0
ResNet101V2	79.4	69.0	80.6
ResNet152	81.8	71.4	82.0
ResNet152V2	66.8	71.0	**83.0**
ResNet50	83.0	71.4	80.4
ResNet50V2	80.8	69.8	78.0
VGG19	75.4	70.6	79.4

In a general analysis, it can be seen that the MLP classifier when combined with the ResNet101 model as deep feature extraction technique demonstrated the best performance, obtaining an f1-score of 84.2%. A similar behavior can be observed when using the XGBoost combined with the ResNet152V2, obtaining an f1-score of 83.0%, i.e., only 1.2% less than the best combination. Contrarily, the highest f1-score using the SVM classifier achieved 75.0% when combined with the EfficientNetB4, i.e., presenting a difference of more than 8%.

B. Comparative analysis of deep feature extraction techniques

In terms of VGG architecture, only the VGG19 model was used to the deep feature extraction. This way, it can be noticed that f1-scores varied between 70.6% and 79.4%. However, when using the XGBoost classifier, the f1-score demonstrated its robustness compared to the other classifiers (increasing by at least 4.0%). Similarly, for Inception architecture, only the InceptionV3 model was considered, demonstrating a low variation in f1-scores (between 74.2% and 75.8%). Again, the XGBoost achieved the highest f1-score.

The results of MobileNet architecture combined with the classifiers were evaluated in relation to two models (MobileNet and MobileNetV2), ranging from 69.4% and 76.8%. Despite not presenting the best results, the MobileNet has a practical relevance, as it can be used embedded in hardware due to the low computational cost and fast response. This way, it can be noticed that the MobileNet model demonstrated some stability in terms of performance, as the f1-scores varied between 71.8% and 76.8%. In a practical way, these results are adequate for an embedded application, as it can be executed in the customers' smart devices.

Analyzing the results achieved with the EfficientNet architecture, it can be seen a similar behavior for MLP and XGBoost classifiers. Both, reached 81.0% as highest f1-scores considering the EfficientNetB6 model. In this sense, the results of all EfficientNet models can be summarized in terms of average and standard deviation, as presented in Table V.

TABLE V
AVERAGE AND STANDARD DEVIATION CALCULATED FOR EACH CLASSIFIER COMBINED WITH THE EFFICIENTNET MODELS.

Classifier	**Avg. f1-score (%)**	**Std. Dev. (%)**
MLP	74.1	4.9
SVM	66.9	5.1
XGBoost	73.3	4.6

Table V demonstrates the high sensitivity of each classifier to the choice of deep feature extraction model. Despite reaching high f1-scores by MLP and XGBoost when using the EfficientNetB6, this combination is not reliable.

The ResNet architecture, as verified in the general analysis, obtained the best results when combined with MLP and XGBoost. However, it is also important to analyze the averages and standard deviations considering all the ResNet models, as shown in Table VI.

TABLE VI
AVERAGE AND STANDARD DEVIATION CALCULATED FOR EACH CLASSIFIER COMBINED WITH THE RESNET MODELS.

Classifier	**Avg. f1-score (%)**	**Std. Dev. (%)**
MLP	79.3	6.4
SVM	71.0	1.6
XGBoost	81.0	1.8

From Table VI, the averages and standard deviations are useful to clarify that the MLP was highly sensitive to the change in ResNet models, being the XGBoost a recommendable and suitable algorithm.

VI. CONCLUSIONS

The usage of time-series transformations into bidimensional images from RP has shown itself as a robust alternative to residential load identification. In this paper, we have proposed the comparison of five deep neural network architectures to perform the stage of feature extraction. Only the models pre-trained with the ImageNet database domain were selected to compose the comparison analysis. Additionally, three shallow machine learning algorithms (MLP, SVM and XGBoost) were used as classifiers, i.e., to obtain the results and allow an adequate comparison between the feature extraction techniques.

Among the analyzed feature extractors, the ResNet neural network architecture reached the best results, mainly when combined with the XGBoost and MLP. Despite the high f1-scores achieved by using the MLP, the standard deviation was high when compared with the XGBoost. In addition, it is important to mention the results obtained by the models of MobileNet architecture, which demonstrated a stable behavior. This way, it can be clearly embedded in hardware or developed as a smart device app.

In future works, we intend to evaluate the performance of MobileNet architecture in a low cost hardware, the generalization capability of an approach based on ResNet as feature extraction and XGBoost as classifier to different residences, and the possibility of perform an adequate transfer learning strategy between different residences.

REFERENCES

[1] H. Rafiq, P. Manandhar, E. Rodriguez-Ubinas, O. Ahmed Qureshi, and T. Palpanas, "A review of current methods and challenges of advanced deep learning-based non-intrusive load monitoring (nilm) in residential context," *Energy and Buildings*, vol. 305, p. 113890, 2024.

[2] J. B. Gutiérrez and C. B. Castañon, "Comparative evaluation of machine learning classifier algorithms for electric load disaggregation based on average power signatures," in *IEEE XXXI International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, 2024, p. 1–7.

[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. 70 746–70 757, 2021.

[4] H. Hwang and S. Kang, "Nonintrusive load monitoring using an lstm with feedback structure," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.

[5] G. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[6] J. Kim, T.-T.-H. Le, and H. Kim, "Nonintrusive load monitoring based on advanced deep learning and novel signature," *Computational Intelligence and Neuroscience*, vol. 2017, pp. 1–22, 2017.

[7] J. Sun, M. Li, P. Shi, O. Li, J. Zhu, W. Hu, and Q. Guo, "Research on non-invasive load decomposition algorithm based on attention mechanism of convolutional neural network," in *Asian Conference on Frontiers of Power and Energy (ACFPE)*, vol. 42, 2022, p. 216–220.

[8] A. Langevin, M.-A. Carbonneau, M. Cheriet, and G. Gagnon, "Energy disaggregation using variational autoencoders," *Energy and Buildings*, vol. 254, p. 111623, 2022.

[9] G. Pan, H. Wang, T. Tian, Y. Luo, S. Xia, and Q. Li, "Research on non-intrusive load decomposition model based on parallel multi-scale attention mechanism and its application in smart grid," *Energy and Buildings*, vol. 312, p. 114210, 2024.

[10] R. Jiao, C. Li, G. Xun, T. Zhang, B. B. Gupta, and G. Yan, "A context-aware multi-event identification method for nonintrusive load monitoring," *IEEE Transactions on Consumer Electronics*, vol. 69, no. 2, p. 194–204, 2023.

[11] M. Hussain, J. J. Bird, and D. R. Faria, "A study on cnn transfer learning for image classification," in *Advances in Computational Intelligence Systems*. Springer, 2019, pp. 191–202.

[12] L. Kyrkou, C. Nalmpantis, and D. Vrakas, "Imaging time-series for nilm," in *International Conference on Engineering Applications of Neural Networks*. Springer, 2019, pp. 188–196.

[13] M. D'Incecco, S. Squartini, and M. Zhong, "Transfer learning for non-intrusive load monitoring," *IEEE Transactions on Smart Grid*, vol. 11, pp. 1419–1429, 2020.

[14] S. Houidi, D. Fourer, F. Auger, H. Sethom, and L. Miègeville, "Comparative evaluation of non-intrusive load monitoring methods using relevant features and transfer learning," *Energies*, vol. 14, p. 2726, 2021.

[15] L. Wang, S. Mao, B. Wilamowski, and R. Nelms, "Pre-trained models for non-intrusive appliance load monitoring," *IEEE Transactions on Green Communications and Networking*, vol. 6, pp. 56–68, 2022.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations (ICLR)*, 2015, pp. 1–14.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[20] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning (ICML)*, 2019, pp. 6105–6114.

[21] D. L. Cavalca and R. A. Fernandes, "Deep transfer learning-based feature extraction: An approach to improve nonintrusive load monitoring," *IEEE Access*, vol. 9, pp. 139 328–139 335, 2021.

[22] J.-P. Eckmann, S. O. Kamphorst, D. Ruelle *et al.*, "Recurrence plots of dynamical systems," *World Scientific Series on Nonlinear Science Series A*, vol. 16, pp. 441–446, 1995.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 248–255.

[24] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *14th International Conference on Artificial Intelligence and Statistics*, 2011, pp. 315–323.

[25] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley Sons, Ltd, 2012.

[26] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

[27] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[28] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378, 2002.