Knowledge Distillation for Scalable Nonintrusive Load Monitoring

Giulia Tanoni, *Graduate Student Member, IEEE*, Lina Stankovic,
Vladimir Stankovic, *Senior Member, IEEE*, Stefano Squartini, *Senior Member, IEEE*,
and Emanuele Principi, *Member, IEEE*

***Abstract*—Smart meters allow the grid to interface with individual buildings and extract detailed consumption information using nonintrusive load monitoring (NILM) algorithms applied to the acquired data. Deep neural networks, which represent the state of the art for NILM, are affected by scalability issues since they require high computational and memory resources, and by reduced performance when training and target domains mismatched. This article proposes a knowledge distillation approach for NILM, in particular for multilabel appliance classification, to reduce model complexity and improve generalization on unseen data domains. The approach uses weak supervision to reduce labeling effort, which is useful in practical scenarios. Experiments, conducted on U.K.-DALE and REFIT datasets, demonstrated that a low-complexity network can be obtained for deployment on edge devices while maintaining high performance on unseen data domains. The proposed approach outperformed benchmark methods in unseen target domains achieving a F_1-score 0.14 higher than a benchmark model 78 times more complex.**

***Index Terms*—Deep learning (DL), knowledge distillation (KD), multilabel appliance classification, nonintrusive load monitoring (NILM), weak supervision.**

I. Introduction

ADVANCED metering infrastructure enables interaction between utilities and users via bidirectional communication [1]. It is expected that by 2025, most European countries will reach wide-scale smart meter roll-out to at least 80% of consumers [2]. Smart meters interface the grid to individual buildings, enabling a building's electricity consumption to be measured and managed remotely. Using smart meter readings, new opportunities arise for energy service providers that can give real-time personalized energy services within the home for users [3], and have better traceability of energy usage to propose strategies for saving energy and balancing energy supply and demand [4].

Continuous availability of energy consumption data has led to the development of advanced techniques to monitor loads inside buildings and provide users with improved awareness of their energy consumption and usage habits. One such technique is nonintrusive load monitoring (NILM) (see [5] for a recent review) which detects ON–OFF states of loads and estimates the power consumption of individual loads in the building based only on the building's aggregate meter readings. NILM has become a very active area of research with widespread smart meter installations in the residential sector.

Due to the availability of a large quantity of low-frequency electrical load measurements from smart meters, deep learning (DL) approaches have recently become popular, representing the current state of the art in NILM both for regression and classification tasks [5], [6], [7], [8], [9], [10], [11], [12], [13].

However, training and inference phases for DL-based approaches require significant memory and computational resources, which limits their scalability, requiring the use of high-performance processors in the cloud. When NILM is performed on the cloud, it involves transferring data from the consumer's premises to central servers, which results in additional transmission costs, raise privacy concerns, and causes delays in the system's response time [14], [15], [16], [17], [18]. These issues can be alleviated by performing training and inference on local devices at the user's end. This, however, requires DL models to have lower computational and memory requirements, as local (edge) devices are characterized by limited computational and memory resources. Moreover, recent studies have demonstrated that transfer learning techniques are necessary to achieve acceptable performance in unseen environments [19], [20]. However, this process requires additional training phases to adapt model parameters, which in turn increases the computational load on edge devices.

Several techniques have been proposed to reduce the complexity of DL-based NILM, such as pruning, tensor decomposition, matrix factorization, [14], [16], weights quantization [17], federated learning [15], and knowledge distillation (KD) [21]. However, there has been little attention in the recent literature on the transfer learning scenario for lower complexity NILM DL approaches, where training is performed on labeled datasets, and this knowledge is transferred to unseen buildings. This

Manuscript received 28 March 2023; revised 10 July 2023; accepted 16 October 2023. Date of publication 9 November 2023; date of current version 23 February 2024. Paper no. TII-23-1053. *(Corresponding author: Giulia Tanoni.)*

Giulia Tanoni, Stefano Squartini, and Emanuele Principi are with the Department of Information Engineering, Università Politecnica delle Marche, 60131 Ancona, Italy (e-mail: g.tanoni@pm.univpm.it; s.squartini@univpm.it; e.principi@univpm.it).

Lina Stankovic and Vladimir Stankovic are with the Department of Electronic and Electrical Engineering, University of Strathclyde, G1 1XQ Glasgow, U.K. (e-mail: lina.stankovic@strath.ac.uk; vladimir.stankovic@strath.ac.uk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2023.3328436.

Digital Object Identifier 10.1109/TII.2023.3328436

work proposes a method based on KD and weak supervision to reduce the complexity of a neural network for multilabel appliance classification. KD enables transferring knowledge from a large teacher network to a smaller student network by training the latter with soft labels from the former [22], [23]. To make the proposed solution scalable and improve KD, transfer learning and complexity reduction are needed. Transfer learning typically requires collecting new data directly from the target environment to fine-tune pretrained models, requiring the engagement of end users for data annotation, which is made simpler by weak supervision. Previous studies have demonstrated the effectiveness of weak labels in improving performance in both disaggregation [24] and multilabel appliance classification tasks [7], [13], and this study proposes using weak labels to jointly distil knowledge and reduce network complexity during transfer learning. The method uses a Convolutional Recurrent Neural Network (CRNN), which has been successfully used in a centralized NILM scenario [7]. In the experiments, the study presents several networks with reduced complexity that retain the main components of the initial model and investigates the trade-off between accuracy and complexity.

The rest of this article is organized as follows. Section II reviews the existing literature for multilabel appliance classification and complexity reduction techniques followed by our contributions. Section III presents the NILM problem formulation. Section IV describes the proposed method. Section V provides a detailed description of the experimental setup, and Section VI discusses the obtained results. Finally, Section VII concludes this article.

II. Related Work and Contributions

This section presents a brief review of multilabel NILM classification approaches, followed by a review of complexity reduction methods for NILM models. Finally, we highlight the research gaps and main contributions of the article.

A. Multilabel Appliance Classification

Multilabel appliance classification approaches rely on the use of one network to learn the joint probability of multiple appliances and classify their activation state. Basu et al. [25] initially proposed a supervised multilabel classifier for NILM, unlike popular one-DL-model per appliance approaches. Also, in [9], a CNN followed by three different fully connected sub-networks is implemented for multilabel state and event type classification. Deep Blind Compressed Sensing is proposed in [10] for multilabel device state detection.

Semisupervised learning strategies have also been proposed for NILM, using a teacher-student framework [6]. Recently, in [11], a semisupervised KD approach has been proposed to improve the transferability on target environments. Differently, in [7], the authors adopt a weakly supervised multilabel approach to reduce the labeling effort to train a CRNN, using both weakly labeled data (labels provided for a group of consecutive samples, e.g., for a 4 hours period) and strongly labeled data (i.e., labeled sample-by-sample). Successively, a transfer learning approach based on weak labels has been proposed in [13].

B. Complexity Reduction in DL-Based NILM

In the literature, complexity reduction approaches for NILM have mainly focused on neurons and filters pruning [14], [15], [16], [26], tensor decomposition [14] and coefficient quantization [17], [26]. In [14], filters are pruned based on their importance defined by L-norms and the change in loss caused by removing a specific filter and neurons. After pruning, the model is re-trained one or more times on a subset of training data, based on the adopted pruning strategy. In [15], pruning techniques are used to reduce the complexity of a large Sequence-to-Point model [27] in a federated learning framework. The authors addressed also transfer learning by using unlabeled data from the target domain. Sykiotis and colleagues [26] presented an edge optimization framework that applies coefficient quantization and pruning to reduce the model's complexity incrementally until specified performance and edge deployment requirements are met. Barber et al. [16] propose two ways to reduce the complexity of the Sequence-to-Point CNN network, using dropout and a smaller number of CNN filters and applying four pruning strategies on the learned weights. The magnitude-based approach, implemented in the TensorFlow Model Optimization toolkit, was revealed to be the best compromise between reduction and accuracy of the model. In [17], the authors propose a post-training MobileNet compression, reducing the model size and inference time with the TensorFlow Lite tool for quantization, where the precision is reduced from 32 to 8 b. Peng and colleagues [21] presented a framework based on KD to obtain a multilayer perceptron network. A similar approach was followed in [28], where KD has been used to obtain a CNN from an ensemble of convolutional networks each having higher complexity. The addressed task, in this case, is multiclass single-label classification, i.e., only one appliance is assumed active at each time instant. Differently, in [18] the authors directly designed a lightweight CNN architecture, suitable for deployment on edge devices.

The reviewed literature mainly addresses complexity reduction for power profile reconstruction [14], [15], [16], [17], [26] with few works considering the performance drop on unseen target domains and the related transfer learning methods for reducing it [15], [18]. This problem is of high practical importance as the *source domain* data used to train the networks are usually statistically different to the *target domain* data that are processed when the network is deployed in the final environment. The statistical difference between the two domains can be attributed to various factors, such as the types of appliances, the measuring equipment, and the building size. As demonstrated by the recent literature, the mismatch between training and testing domains leads to poor performance, and transfer learning is necessary to achieve acceptable results [20], [29]. The authors in [20] transfer the features extracted by the CNN layers of the Sequence-to-Point network across appliances and households in different regions and fine-tune the regression layer. Differently, in [29], the authors combined federated learning and meta-learning, where a group of meta-learned models are trained locally using metering data from residential communities. It is worth noting that both [20], [29] do not propose a complexity reduction method and deal with power profile reconstruction. In the complexity reduction literature, only [18] has evaluated the method in a data domain that differs from the training one, and only [15] addressed transfer learning. Both papers address power profile reconstruction, i.e., the regression task.

C. Contributions

In light of the reviewed literature, the following research gaps can be identified.

1) Complexity reduction based on KD has never been addressed for multilabel appliance classification, i.e., when more than one appliances can be active at the same time instant.
2) Transfer learning has not been previously addressed jointly with complexity reduction, particularly in the KD framework, and for multilabel appliance classification.
3) Weak labels have never been used in KD, particularly for complexity reduction on the multilabel appliance classification task.

To fill these gaps, this article proposes a multilabel appliance classification method based on KD. The proposed approach reduces neural network model complexity in terms of trainable parameters and computational load while reducing the performance degradation on unseen target domains by integrating transfer learning. The proposed framework integrates weak labels, annotations that provide superior performance compared to unlabeled data and that require less annotation effort compared to strong labels [7]. We investigate a real-world scenario where the network model is initially trained on a large quantity of publicly available measurements, annotated with strong and weak labels. Then, only weakly annotated data are available, labeled by end-users in a target environment, to fine-tune the network [13] and to distil the less complex model. Note that in the proposed method, end users are asked only for weak information about their appliance usage, with a significant reduction of labeling effort compared to strong labels and improved performance compared to unlabeled data.

The same fine-tuning dataset has been considered to distil the knowledge to the student network. Thus, in our method, two types of labels are exploited during distillation: 1) soft labels from the fine-tuned teacher (which may not be entirely accurate), and 2) the ground-truth weak labels of the target data domain. Using the pretrained and fine-tuned teacher, we also improve the convergence of KD and student learning, mitigating performance degradation.

The experimental evaluation has been conducted on two public datasets, U.K.-DALE [30] and REFIT [31]: both have been used as different *source domain* scenarios and a subset of REFIT houses as *target domain*. Moreover, the proposed approach has been compared to three benchmark recently presented methods, EdgeNILM [14], LightweightCNN [18], and [7].

To the best of our knowledge, this work is the first to address KD to reduce architecture complexity jointly with transfer learning for multilabel appliance classification. Specifically, the contributions of our work are the following: 1) a new method for classifying multiple appliances at every time instant using a low complexity network that can be deployed on the edge (Section IV); 2) deep neural network complexity reduction and transfer learning, jointly addressed resulting in a decrease of the performance gap when the target and training domain are different (Section IV-A); and 3) integrating weak labels in the distillation framework to improve generalization and reduce the data annotation effort in the target domain (Section IV-B).

III. Problem Statement

The total power measurement of a building can be modeled as the sum of all M power loads of the building plus noise $\epsilon(t)$, from measurement error and unknown loads

$$y(t) = \sum_{m=1}^{M} x_m(t) + \epsilon(t) \qquad (1)$$

where $x_m(t)$ is the power consumed by the appliance m at the time instant t. In multilabel appliance classification, the aim is to predict sample-by-sample the state of each of K appliances of interest from the aggregate power measurements $y(t)$, where $K \leq M$. Let $s_m(t)$ be the state that indicates if appliance m is ON at time sample t ($s_m(t) = 1$), i.e., if $x_m(t)$ is greater than a power threshold, or OFF ($s_m(t) = 0$). Then the task is to find $s_m(t) \in \{0, 1\}$, for all $m = 1, \ldots, K$ and $t = 1, \ldots, N$.

We divide the input signal $y(t)$ into a series of J disjointed windows of size L samples where the jth window is represented by the vector

$$\mathbf{y}_j = [y(jL), \ldots, y(jL + L - 1)]^T \in \mathbb{R}^{L\times 1}. \qquad (2)$$

Then, we define the corresponding series of J disjointed windows of labels as

$$\mathbf{S}_j = [\mathbf{s}(jL), \mathbf{s}(jL + 1), \ldots, \mathbf{s}(jL + L - 1)] \in \mathbb{R}^{K\times L} \qquad (3)$$

and call them *strong* labels. Note that above $\mathbf{s}(t) = [s_1(t), \ldots, s_m(t)]$ is a strong label vector at time stamp t. In addition, for each jth window, we define the one-hot vector $\mathbf{w}_j = [w_1, \ldots, w_K] \in \mathbb{R}^{K\times 1}$ as *weak* labels, where $w_m = 1$ means the appliance mth is on for a complete operating cycle inside the jth window.

IV. Proposed Methodology

Fig. 1 shows the proposed KD framework. Two learning phases, pretraining and fine-tuning, are performed on the teacher network and one, distillation, on the student. The teacher network is initially trained on a large dataset of active power measurements and the corresponding strong and weak labels $\{\mathbf{y}_j, \mathbf{S}_j, \mathbf{w}_j\} \in D_1$. Then, the network is fine-tuned on a smaller set $\{\mathbf{y}_j, \mathbf{w}_j\} \in D_2$ without any strong labels.

To ensure the practicality of the proposed architecture, all learning phases are based on weak supervision [32]. That is, it is assumed that only the large teacher network has access to exact event labels (*strong* labels) in the pretraining phase, while the student network is created locally, at the target environment, with access to *weak* labels only. For example, the teacher can be trained using a large public source domain dataset, and fine-tuning is performed using easier-to-collect weak labels from

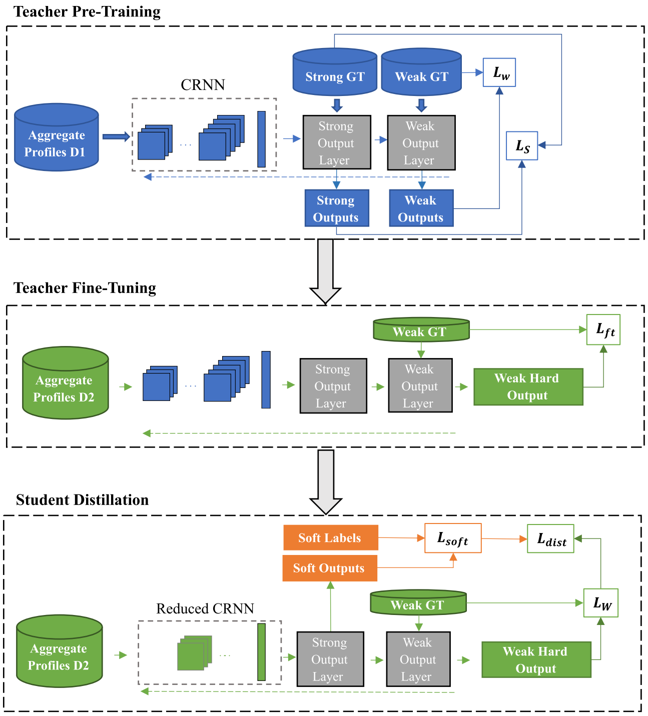

Fig. 1. Proposed knowledge distillation framework for NILM. "GT" stands for "Ground Truth."

the target domain (collected, e.g., periodically from the targeted house via an app).

The method is based on a weak supervised distillation approach in which the network takes as input a series of J disjointed windows of $y(t)$ of size L and produces as output a series of J disjointed windows of predictions for K classes

$$\hat{\mathbf{S}}_j = [\hat{\mathbf{s}}(jL), \hat{\mathbf{s}}(jL+1), \ldots, \hat{\mathbf{s}}(jL+L-1)] \in \mathbb{R}^{K\times L} \tag{4}$$

where $\hat{\mathbf{s}}(t) \in \mathbb{R}^{K\times 1}$, contains predictions (ON/OFF) for each of K appliances of interest at time stamp t.

The distillation process is performed using the teacher–student strategy described in [22]. The following sections detail the teacher and the student training methodology. The final subsection is dedicated to the teacher architecture and the factors that influence the dimension of the network.

A. Teacher Learning

The teacher model implements the function $g_\phi(\cdot)$ with parameters $\boldsymbol{\phi}$ and it is initially pretrained using both strongly and weakly labeled data, i.e., the dataset D_1. The loss function is defined as

$$L_{pt} = L_s + \lambda L_w \tag{5}$$

where the two losses are the binary cross-entropy (BCE) function calculated on the strong predictions and on the weak predictions, respectively, as

$$L_s(\mathbf{S}_j, \hat{\mathbf{S}}_j) = -\frac{1}{K}\frac{1}{L}\sum_{m=1}^{K}\sum_{t=1}^{L}[s_m(t)\log(\hat{s}_m(t)) + (1-s_m(t))\log(1-\hat{s}_m(t))] \tag{6}$$

$$L_w(\mathbf{w}_j, \hat{\mathbf{w}}_j) = -\frac{1}{K}\sum_{m=1}^{K}[w_m\log(\hat{w}_m) + (1-w_m)\log(1-\hat{w}_m)] \tag{7}$$

where $\hat{s}_m$ represents the sample-by-sample state predictions, $\hat{w}_m$ represents the weak state predictions, $w_m \in \{0,1\}$ is the weak ground-truth label, for each window of size L. The rationale behind the use of BCE loss for multilabel multiclass problems is that the task is reduced to multiple binary classification problems, one for each appliance. Individual BCE loss terms are calculated for each output neuron, then they are summed to obtain the final loss.

Unlike previous works on distillation [22], [33], before being employed in the distillation process, the teacher network here is fine-tuned on a subset of data D_2 from the target environment using weak labels only. The fine-tuning loss L_{ft} is formulated as the focal loss [34], with γ set to 0.2

$$L_{ft}(\mathbf{w}_j, \hat{\mathbf{w}}_j) = -\frac{1}{K}\sum_{m=1}^{K}[w_m(1-\hat{w}_m)^\gamma\log(\hat{w}_m) + (1-w_m)\hat{w}_m^\gamma\log(1-\hat{w}_m)]\,. \tag{8}$$

Generally, positive and negative samples are highly unbalanced, as the latter are significantly more represented. Moreover, preliminary experiments on the validation set showed that the classification of negative samples is significantly less challenging, with specificity values around 0.99. This, motivated us to use the focal loss proposed in [34] instead of the binary cross-entropy loss. The focal loss focuses better on incorrect instances of the underrepresented class (positive samples in our case), while down-weighting the contribution of correctly classified samples related to the mostly represented class (negative samples in our case). In this way, the loss helps the teacher in learning about the target domain data available before distillation, particularly when using the coarser information from weak labels. We experimentally verified on the validation set that using the focal loss reduces the presence of false positive and negative predictions and increases the true positives depending on the appliance. All network layers have been fine-tuned since we have verified on the validation set that better performance is obtained by retraining the entire network.

B. Student Knowledge Distillation

The student model implements the function $f_\alpha(\cdot)$ with parameters $\boldsymbol{\alpha}$. The weakly labeled dataset D_2 exploited to fine-tune the teacher network has also been employed in the distillation process. Thus, the distillation loss function is defined as

$$L_{\text{dist}} = \beta L_{\text{soft}}\left(\sigma\left(\frac{\mathbf{Z}_j^{st}}{T}\right), \sigma\left(\frac{\mathbf{Z}_j^{te}}{T}\right)\right) + (1-\beta)\theta(e)L_w(\hat{\mathbf{w}}_j^{st}, \mathbf{w}_j) \tag{9}$$

where L_{soft} is the BCE, as in (6), calculated on the soft outputs of the student $\tilde{\mathbf{S}}_j^{st} = \sigma(\mathbf{Z}_j^{st}/T)$ and the soft labels from the

```
Require: Datasets $D_1$ and $D_2$, $Teacher$ $g_{\boldsymbol{\phi}}(\cdot)$ pre-trained on
    $D_1$ and fine-tuned on $D_2$, $Student$ $f_{\boldsymbol{\alpha}}(\cdot)$, $\theta(\cdot)$ function
    to balance losses magnitude.
  for $e$ in $epochs$ do
    for each minibatch $B$ do
      $\tilde{\mathbf{S}}^{te}_{j\in B} \leftarrow g_\phi(\mathbf{y}_{j\in B})$;
      $\tilde{\mathbf{S}}^{st}_{j\in B}, \hat{\mathbf{w}}^{st}_{j\in B} \leftarrow f_\alpha(\mathbf{y}_{j\in B})$;
      $L_{dist} \leftarrow \beta L_{soft}(\tilde{\mathbf{S}}^{st}_{j\in B}, \tilde{\mathbf{S}}^{te}_{j\in B}) + (1 - \beta)\theta(e) L_w(\hat{\mathbf{w}}^{st}_{j\in B}, \mathbf{w}_{j\in B})$;
      Update $\boldsymbol{\alpha}$ using Adam Optimiser to minimise $L_{dist}$
      loss.
    end for
  end for
```

Fig. 2. Pseudo-code for the Student distillation process.

teacher $\tilde{\mathbf{S}}^{te}_j = \sigma(\mathbf{Z}^{te}_j/T)$ with σ being the sigmoid function, and $\mathbf{Z}^{st}_j$ and $\mathbf{Z}^{te}_j$ the logits from the student and the teacher, respectively. L_w is the BCE computed on the weak predictions $\hat{\mathbf{w}}^{st}_j$ of the student and $\mathbf{w}_j$ the weak ground-truth, as in (7). $\theta(e)$ is a dynamic weight that balances the magnitude of the two losses based on the following formula $\theta(e) = 10^{-G(e)}$, where $G(e)$ is obtained by $G(e) = \log_{10}(\mathcal{L}_w(e)) - \log_{10}(\mathcal{L}_{\text{soft}}(e))$, e is the training epoch, and $\mathcal{L}_w(e)$ and $\mathcal{L}_w(e)$ are the total losses for epoch e. β balances the contribution of the teacher knowledge and the weak ground-truth to guide the training process. T is the temperature parameter used to soften teacher predictions [22]. β and T have been defined for each network architecture experimentally, based on the performance on the validation set. Fig. 2 shows the pseudo-code for the student distillation process.

C. Neural Network Architectures

The teacher network is based on a CRNN, previously used in [7]. The network is composed of $H = 3$ convolutional blocks, each containing a convolutional layer with $F \cdot H$ filters ($F = 32$), with kernel size equal to $k_e = 5$, a batch normalization layer, a rectified linear unit (ReLU) activation and a dropout layer with probability equal to 0.1. The stride d is 1 and the padding modality is "same." The recurrent subpart is composed of a bidirectional gated recurrent units (GRUs) layer, with 64 units (U). The final part of the network is composed of a dense layer with K neurons followed by a sigmoid activation function that produces the appliances' state sample by sample. After the dense layer, the *linear softmax* pooling layer followed by a sigmoid activation layer, produces the weak prediction. We choose linear softmax pooling over other functions proposed in literature as it is shown to reduce the incongruities between strong and weak labels leading to improved performance [7], [35].

The total number of trainable parameters for the convolutional subpart can be computed as

$$N_{\text{CNN}} = \sum_{h=1}^{H} (k_e d \cdot F_{h-1} + 1) F_h + n_{BN} \tag{10}$$

with $F_h = F \cdot h$, and $n_{BN} = 4F_h$ that represents the number of parameters associated to the batch normalization (two trainable plus two nontrainable). F_{h-1} is the number of feature maps in the input for the hth layer while F_h is the number of feature maps in the output. When $h = 1$ the F_0 is the dimension of the input data. Thus, N_{CNN} mainly depends on the number of convolutional blocks. The recurrent subpart has a number of parameters N_{RNN} computed as [36]

$$N_{\text{RNN}} = 2[3(U^2 + UF_H + 2U)] \tag{11}$$

where the last term depends on the used framework and is $2U$ for Keras and PyTorch. N_{RNN} depends on the number of recurrent units considered U, biases, and the input dimension F_H. Equations (10) and (11) indicate that the number of convolutional blocks and the number of recurrent units are the main factors that increase the total number of parameters and hence the overall complexity. In this work, several student architectures with reduced complexities are evaluated in the edge computing direction. The various student architectures are presented in Section V.

V. Experimental Setup

A. Dataset

Two widely used real-world datasets, U.K.-DALE [30] and REFIT [31], have been used to evaluate the proposed method. U.K.-DALE contains data from five buildings sampled every 6 s. REFIT contains data from 20 houses sampled at 8 s. The datasets have been balanced with the same procedure, as in [7], and REFIT was resampled to 6 s as U.K.-DALE. In our experiments, we used the cleaned REFIT dataset, where gaps and outliers were addressed, as explained in detail, in the dataset publication [31]. Similarly, we preprocessed U.K.-DALE, as suggested in [30], to remove gaps both in the aggregate and appliances data. The appliances considered are Kettle (KE), Microwave (MW), Dishwasher (DW), Washing Machine (WM), Toaster (TOA), and Washer Dryer (WD) since they are present in most households and also present in most of the houses in both datasets. A subset of houses from REFIT (2, 4, 8, 9, 15) is used as a test set from which the set for fine-tuning D_2 the teacher network has been extracted (30% of the total number of windows). The fine-tuning set is the same as that used for the distillation of the student.

To evaluate our approach in practical scenarios, we consider two different pretraining sets D_1 for the teacher: 1) Houses 5, 6, 7, 10, 12, 13, 16, 17, 18, and 19 of REFIT; 2) Houses 1, 3, 4, and 5 of U.K.-DALE. These houses are selected based on the availability of the six appliances of interest. The first scenario is to evaluate the method in more favorable conditions when the pretraining domain is similar to the target data domain. The second allows us to evaluate the method performance when the pretraining and target data domains are statistically different [37]. The validation sets contain 20% of data from each training house. Input data are normalized using the mean and the standard deviation estimated on the pretraining sets.

TABLE I

PRETRAINING SETS CHARACTERISTICS. MEAN POWER (EXPRESSED IN WATT) REFERS TO THE MEAN POWER ESTIMATED IN A COMPLETE ACTIVATION WHILE THE DURATION (EXPRESSED IN MINUTES) REFERS TO THE LENGTH

Dataset	Kettle		Microwave		Toaster		Washing Machine		Dishwasher		Washer Dryer	
	Mean Power	Duration	Mean Power	Duration	Mean Power	Duration	Mean Power	Duration	Mean Power	Duration	Mean Power	Duration
UK-DALE	1968	2.15	969	1.9	1437	3.38	512	85.8	802	106	504	85
REFIT	2066	2.4	961	2.5	1148	1.9	301	91.8	598	97	1060	30.4

B. Hyperparameters

The input sliding window dimension L in the teacher model is the first hyperparameter that influences the distillation process. Table I shows the duration and average power values for all the appliances of interest. For long-activation appliances, the window size L is fixed to 4 h and 15 min (2550 samples), as in [7], where the authors selected this length to ensure a complete activation is contained within a window. Instead, we examine a series of reduced window lengths for short-activation appliances (around 2–4 min) after having analyzed activations in both pretraining datasets. We identified a total of four window lengths equally distributed from 55 min (540 samples) to 4 h and 15 min (2550 samples). We chose a minimum time interval of 55 min as it is appropriate for weak labels annotation. Thus, the selected window sizes are 55 min (540 samples), 2 h and 2 min (1210 samples), 3 h and 8 min (1880 samples), and 4 h and 15 min (2550 samples). A smaller window for short-activation appliances makes weak labels more effective during the training phase, and multiple activations inside the same window can be accurately detected. Section VI-A presents a comparison of the results obtained with windows of different lengths. We note that, from a practical point of view, using the one-hour windows for these appliances is a reasonable length for accurately assigning weak labels since users are less likely to remember appliances used within less than one-hour windows confidently.

The parameter β has been varied in the range 0.3–0.9 with a step of 0.2, and T has been tested with values [0.5, 0.7, 0.9, 2]. β and T have been optimized for each student network based on the validation set that has also been used to find the best threshold to quantize the network predictions. The learning rate used is 0.002. The number of epochs has been set to 1000, and early stopping with patience equal to 30 epochs has been used to avoid overfitting. The batch size is set to 64.

C. Architecture Complexity Evaluation

As introduced in Section IV-C, to reduce the student architecture we maintain the main components of the teacher network, and change the parameters of both convolutional and recurrent subparts. First, we reduce the number of convolutional blocks and consider two structures, one with $H = 2$ and one with $H = 1$. Then, we fix $H = 1$ and start to decrease U by a factor of 2 to further reduce the architecture dimension and computational complexity. Table II reports the N_{CNN} and N_{RNN} for each architecture while Table III reports the number of floating point operations (FLOPs) and the dimension of the models to evaluate the reduction in terms of size and runtime [14]. The student models are named with the number of H convolutional blocks and recurrent units U, e.g., student 2H-64U denotes a student architecture with $H = 2$ convolutional blocks and $U = 64$ units. The model named student has the same architecture of the teacher. As shown in Table III, the window dimension significantly affects the number of FLOPs.

TABLE II

TOTAL NUMBER OF PARAMETERS FOR CONVOLUTIONAL AND RECURRENT SUBPARTS ARE REPORTED FOR THE TEACHER NETWORK AND EACH STUDENT ARCHITECTURE

Network	N_{CNN}	N_{RNN}
Teacher	52486	74496
Student	52486	74496
Student 2H-64U	10880	49920
Student 1H-64U	320	37632
Student 1H-32U	320	12672
Student 1H-16U	320	5219

TABLE III

NUMBER OF FLOPS AND SIZES OF TEACHER AND STUDENT NETWORKS FOR DIFFERENT WINDOW LENGTHS (IN SAMPLES) AND NUMBER OF CLASSES $K = 3$ AND $K = 6$

Network	Window size	Classes	FLOPs	Size (KB)
Teacher	540 (55 min)	3	56.4 M	551
Student			56.4 M	551
Student 2H-64U			11.9 M	284
Student 1H-64U			0.7 M	185
Student 1H-32U			0.5 M	85
Student 1H-16U			0.3 M	55
Teacher	2550 (4h 15 min)	6	267.8 M	552
Student			267.8 M	552
Student 2H-64U			57.8 M	285
Student 1H-64U			5.1 M	186
Student 1H-32U			3.1 M	87
Student 1H-16U			2.1 M	55

D. Benchmark Methods

In the experiments, we compared our method with two existing techniques in literature that propose complexity reduction for NILM [14], [18] and with [7] that is a recent multilabel appliance classification approach that proposed a CRNN architecture, similar to the one proposed by our work. None of the works presented in Section II proposes a complexity reduction approach for multilabel appliance classification. Therefore, the works [14] and [18] were adapted for this task. EdgeNILM [14] uses pruning and tensor decomposition, and in the experiments we used the source code made available by the authors to ensure reproducibility. To adapt the network to multilabel appliance classification, we modified the last layer of the sequence-to-point CNN with a sigmoid function to produce the state probability and used the BCE loss function during training. As in [14], we trained a separate network for each appliance and applied the 60% iterative pruning complexity reduction method because in [14] it produced the average lowest disaggregation error. A window size of 99 samples was adopted for EdgeNILM for all the appliances, based on the results presented in [14].

The LightweightCNN proposed in [18] is based on a model design approach, and it consists of only two convolutional layers and one dense layer. The lightweight network was implemented and trained within the same framework of EdgeNILM for a fair comparison, using a window size of 199 samples [18]. As with EdgeNILM, for this approach, we trained a separate network for each appliance.

Finally, we compare the proposed method with [7], where the authors adopted a CRNN structure trained with weakly labeled data. This method is identified as WL-NILM. In this way, we demonstrate the effectiveness and novelty of our method in terms of complexity-performance improvement also when compared with an approach that uses a CRNN and weak labels during training.

The same postprocessing applied for our method was applied to the raw predictions of benchmark methods, using a threshold for each network optimized on the validation set.

E. Evaluation Metrics

Four metrics commonly used in NILM classification literature have been considered to evaluate our method. Defining true positive (TP) as the number of correctly classified active samples, false positive (FP) as the number of inactive samples incorrectly classified as active, and false negative (FN) as the number of active samples incorrectly classified as inactive, we used the recall (R) and precision (P) defined as $\mathrm{R} = TP/(TP + FN)$, $\mathrm{P} = TP/(TP + FP)$, to evaluate the percentage of active samples that are not detected and percentage of inactive samples predicted as active, respectively. The F_1-score is the harmonic mean between precision and recall and is formulated as $\mathrm{F}_1 = 2 \cdot \mathrm{P} \cdot \mathrm{R}/(\mathrm{P} + \mathrm{R})$. The load estimation is evaluated using the total energy correctly assigned (TECA) [38] defined as follows:

$$\mathrm{TECA} = 1 - \frac{\sum_k \sum_t |\hat{x}_k(t) - \bar{x}_k(t)|}{2 \sum_t \bar{y}(t)} \tag{12}$$

with $\bar{y}(t) = \sum_k \bar{x}_k(t)$. $\hat{x}_k(t)$ and $\bar{x}_k(t)$ are, respectively, the product of the average power consumed by appliance k at the time instant t and estimated states $\hat{s}_k(t)$ and the ground-truth states $s_k(t)$. The average power consumed by each appliance has been assigned based on the average power consumed by the appliances in the training set.

VI. Results and Discussion

A. Window Length Impact on Teacher Performance

We present the teacher performance for Kettle, Microwave, and Toaster to evaluate the best window length for classifying short-activation appliances. In this way, we validate the hypothesis that using a window shorter than the one used in [7] leads to improved performance. Figs. 3 and 4 report the teacher performance after fine-tuning on the target data for different window lengths for aforementioned appliances when pretraining is performed on REFIT and U.K.-DALE, respectively. Although only the results on the test set are reported, the performance on the validation set reflects the performance on the test set.

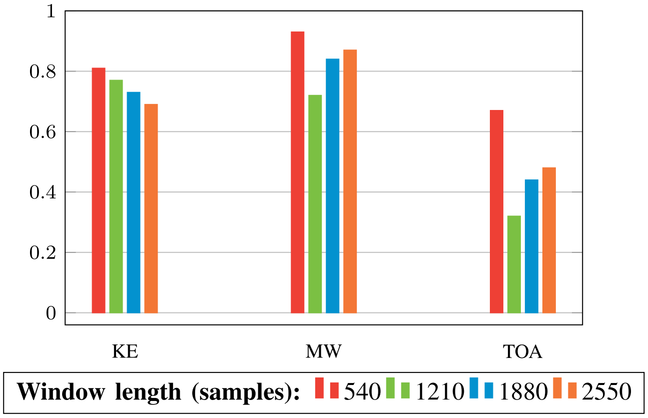

Fig. 3. Window analysis based on the test set: F_1-scores when the Teacher is pretrained with REFIT.

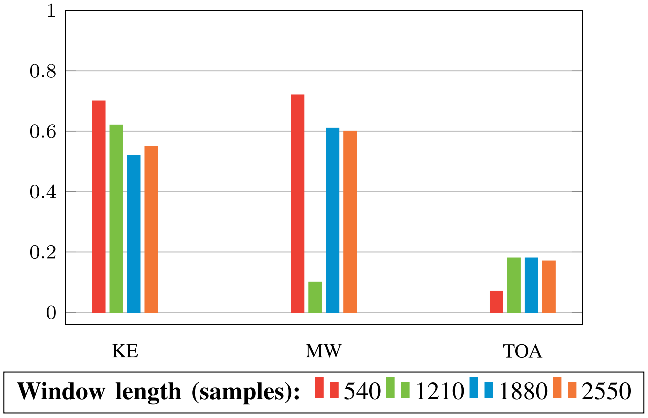

Fig. 4. Window analysis based on the test set: F_1-scores when the Teacher is pretrained with U.K.-DALE.

Both figures show that the reduced length window of 540 samples enables more effective detection of the appliances' states. This is confirmed for both pretraining set conditions and all the appliances except for the Toaster. The Toaster's performance is affected by the statistical differences in power and duration between the activations in the pretraining and the test set, leading to a small drop in an already poor performance. For Microwave, the difference in duration between activations from different domains is reduced when the network focuses on a shorter time window.

B. Student Distillation Results

Tables IV and V present the results obtained with different student architectures, compared to the teacher performance for all the $K = 6$ appliances. When using U.K.-DALE for pretraining (Table V), the student network shows similar performance to the teacher network with slight improvement for Kettle, Dishwasher, and Washing Machine. Similarly, when the teacher is pretrained with REFIT, the results are either improved or similar for Kettle, Toaster, Washing Machine, and Dishwasher. A significant drop in performance is observed only for Washer Dryer due to low

TABLE IV
PERFORMANCE COMPARISON BETWEEN THE TEACHER (D_1 = REFIT) AND THE STUDENT NETWORKS

	Teacher			Student			Student 2H-64U			Student 1H-64U			Student 1H-32U			Student 1H-16U		
Appliance	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	PR	R	F_1	P	R	F_1
KE	0.89	0.74	0.81	0.88	0.75	0.81	0.89	0.66	0.76	0.89	0.74	**0.81**	0.88	0.74	0.80	0.86	0.81	**0.83**
MW	0.88	0.98	0.93	0.85	0.98	0.91	0.82	0.98	0.90	0.83	0.98	0.90	0.85	0.98	0.91	0.74	0.93	0.82
TOA	0.52	0.94	0.67	0.77	0.74	**0.76**	0.76	0.73	**0.74**	0.77	0.71	**0.74**	0.79	0.75	**0.77**	0.03	0.0	0.0
WM	0.57	0.91	0.70	0.56	0.95	**0.71**	0.62	0.92	**0.74**	0.61	0.92	**0.74**	0.58	0.94	**0.72**	0.58	0.93	**0.71**
DW	0.35	0.97	0.51	0.36	0.98	**0.52**	0.38	0.97	**0.55**	0.38	0.97	**0.55**	0.39	0.98	**0.56**	0.42	0.93	**0.58**
WD	0.93	0.52	0.67	0.97	0.40	0.57	0.98	0.38	0.55	0.97	0.44	0.60	0.91	0.61	**0.73**	0.98	0.34	0.51
AVG.	0.69	0.84	0.71	0.73	0.80	**0.73**	0.74	0.77	**0.71**	0.72	0.83	**0.72**	0.73	0.83	**0.75**	0.60	0.66	0.57

Improved performance are in bold and underlined while equal performance are in bold for the reduced student architectures.

TABLE V
PERFORMANCE COMPARISON BETWEEN THE TEACHER (D_1 = U.K.-DALE) AND THE STUDENT NETWORKS

	Teacher			Student			Student 2H-64U			Student 1H-64U			Student 1H-32U			Student 1H-16U		
Appliance	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	PR	R	F_1	P	R	F_1
KE	0.60	0.84	0.70	0.61	0.86	**0.71**	0.56	0.82	0.67	0.58	0.90	**0.71**	0.62	0.87	**0.73**	0.59	0.88	**0.70**
MW	0.57	0.99	0.72	0.53	0.99	0.69	0.50	0.99	0.66	0.62	0.97	**0.75**	0.61	0.95	**0.75**	0.68	0.97	**0.80**
TOA	0.22	0.04	0.07	0.26	0.04	0.07	0.31	0.03	0.05	0.06	0.01	0.02	0.07	0.0	0.0	0.25	0.04	**0.07**
WM	0.56	0.69	0.62	0.57	0.74	**0.65**	0.62	0.67	**0.65**	0.70	0.40	0.51	0.52	0.78	**0.63**	0.69	0.35	0.46
DW	0.49	0.84	0.62	0.50	0.87	**0.63**	0.48	0.88	**0.62**	0.38	0.92	0.54	0.39	0.93	0.55	0.39	0.92	0.54
WD	0.79	0.77	0.78	0.76	0.79	0.78	0.75	0.79	0.77	0.79	0.76	**0.78**	0.75	0.80	**0.78**	0.73	0.82	0.77
AVG.	0.54	0.70	0.59	0.54	0.72	0.59	0.57	0.65	0.57	0.52	0.66	0.55	0.49	0.72	0.57	0.55	0.66	0.56

Improved performance are in bold and underlined while equal performance are in bold for the reduced student architectures.

recall. This is because Washer Dryer activations in the test set are longer than the activations in REFIT pretraining set (approximately 82 min versus 30 min). These statistical differences cause the network to miss or underestimate more activations, producing more false negatives. As shown in Table V, when the student architecture is reduced, differences between domains become more critical because the network loses the last convolutional block related to higher level features. In fact, for the student 2H-64U network in Table V, N_{CNN} reduces by 79% and N_{RNN} by 33% compared to the teacher, while the F_1-score reduces only by 3.4%, on average, due to a decrease in recall not compensated by the slight increase in precision. In contrast, for the same student 2H-64U architecture distilled by the teacher pretrained on REFIT, in Table IV, the performance improves for the Toaster, Washing Machine, and Dishwasher, and remains stable for other appliances, except for Washer Dryer due to low recall. In the smaller student 1H-64U network, N_{CNN} reduces by 99% and N_{RNN} by 49%, while the F_1-score decreases by 6.8% due to both recall and precision drop after the distillation from the teacher pretrained on U.K.-DALE (Table V). This important reduction of high-level features affects the performance, particularly for Toaster, Dishwasher, and Washing Machine. Nonetheless, Kettle and Microwave are more accurately classified while Washer Dryer maintains stable performance. When the Teacher is pretrained on REFIT, the F_1-score of student 1H-64U improves by 1.4% on average compared to the teacher, with stable performance for Kettle, an improvement for Toaster, Dishwasher, and Washing Machine, with an exception for Microwave and Washer Dryer that slightly decrease. In this case, the network produces fewer false activations compared to the teacher network, as confirmed by the higher precision.

The student 1H-32U (N_{RNN} reduced by 83%) represents a good compromise between complexity reduction and performance. This architecture improves teacher performance in both pretraining scenarios. This behavior shows that this architecture helps to improve the student generalization ability independently of the pretraining set characteristics.

For the student 1H-16U (N_{RNN} reduced by 93%), the F_1-score decreases for appliances with longer activations (26% for Washing Machine, 13% for Dishwasher, and 1% for Washer Dryer), while Kettle, Microwave, and Toaster have increased performance, compared to the Teacher pretrained on U.K.-DALE. Particularly, activations of Washing Machine are not well detected while more false activations have been produced for Dishwasher and Washer Dryer. The performance indicates that the number of recurring units may be too small to learn patterns of household appliances with longer activation, when the domains are very different. In fact, the student 1H-16U distilled from the teacher pretrained on REFIT has good performance for Kettle and longer activations appliances, like Dishwasher and Washing Machine, while for Microwave and Washer Dryer, the performance is reduced by 12% and 24%, respectively. It has to be noted that each reduced student architecture reports an improvement for the Washing Machine and Dishwasher, suggesting that when domains are similar the classification of these appliances is positively influenced by complexity reduction. Conversely, Microwave performance slightly decreases compared to the teacher for each student configuration due to the higher presence of false activations. Washer Dryer and Kettle are more dependent on the structure of the student, while Toaster seems to be independent except for the student 1H-16U, where performance falls to 0%. The same holds when the reduced student networks are distilled from a teacher pretrained on U.K.-DALE, mainly due to teacher capability.

Due to the differences between the domains and loads characteristics, all the appliances are more influenced by the student structure, and performance varies for each architecture. Nonetheless, student 1H-32U performs better than the other

TABLE VI
PERFORMANCE COMPARISON BETWEEN THE TEACHER (D_1 = REFIT) AND THE REDUCED STUDENT NETWORKS IN TERMS OF TECA

Appliances	Teacher	Student	Student 2H-64U	Student 1H-64U	Student 1H-32U	Student 1H-16U
KE, MW, TOA	0.827	**0.832**	0.820	**0.828**	**0.827**	0.822
WM, DW, WD	0.648	**0.657**	**0.656**	**0.680**	**0.740**	0.644

Improved and equal performance are reported in bold for each student architecture.

TABLE VII
PERFORMANCE COMPARISON BETWEEN THE TEACHER (D_1 = U.K.-DALE) AND THE REDUCED STUDENT NETWORKS IN TERMS OF TECA

Appliances	Teacher	Student	Student 2H-64U	Student 1H-64U	Student 1H-32U	Student 1H-16U
KE, MW, TOA	0.627	**0.631**	0.608	0.624	**0.663**	**0.642**
WM, DW, WD	0.725	0.719	0.713	0.688	0.674	0.669

Improved and equal performance are reported in bold for each student architecture.

TABLE VIII
RESULTS IN TERMS OF F_1-SCORE OF THE PROPOSED APPROACH AND BENCHMARK METHODS TRAINED WITH D_1 = REFIT AND TESTED ON REFIT

	Appliance						
Model	**KE**	**MW**	**WM**	**DW**	**TOA**	**WD**	**Average**
EdgeNILM Unpruned [14]	0.81	0.41	0.19	0.31	0.21	0.41	0.39
EdgeNILM Pruned 60% [14]	**0.82**	0.29	0.19	0.31	0.11	0.51	0.37
LightweightCNN [18]	0.74	0.65	0.34	0.62	0.11	0.32	0.46
WL-NILM [7]	0.74	0.71	0.54	0.43	0.25	0.02	0.45
Teacher	0.81	**0.93**	0.67	0.70	0.51	0.67	0.71
Student 1H-32U	0.80	0.91	**0.77**	**0.72**	**0.56**	**0.73**	**0.75**

Best results are reported in bold.

TABLE IX
RESULTS IN TERMS OF F_1-SCORE OF THE PROPOSED APPROACH AND BENCHMARK METHODS TRAINED WITH D_1 = U.K.-DALE AND TESTED ON REFIT

	Appliance						
Model	**KE**	**MW**	**WM**	**DW**	**TOA**	**WD**	**Average**
EdgeNILM Unpruned [14]	0.64	0.01	0.43	0.19	0.02	0.23	0.25
EdgeNILM Pruned 60% [14]	0.68	0.03	-	0.07	0.02	-	0.13
LightweightCNN [18]	**0.75**	0.33	0.51	0.53	0.06	0.42	0.43
WL-NILM [7]	0.73	0.07	0.10	0.44	0.04	0.14	0.26
Teacher	0.70	0.72	**0.62**	0.62	0.07	0.68	**0.59**
Student 1H-32U	0.73	**0.75**	0.0	**0.63**	**0.55**	**0.78**	0.57

Best results are reported in bold.

structures, with the smallest performance degradation (3% for U.K.-DALE pretraining) and highest performance improvement (6% for REFIT pretraining) with a reduction of 10× in number of parameters, coherently in both pretraining scenarios. This outcome can be motivated by a good balance between the number of convolutional blocks, that extract only local features, and the number of recurrent units that take the features as input. The results in Tables VI and VII show a comparison between the network structures in terms of TECA, where long- and short-duration appliances are considered separately. For appliances with shorter activations, when the teacher is pretrained with U.K.-DALE, there is a decrease in energy estimation of 3% for student 2H-64U and of 0.5% for student 1H-64U. For other architectures, the energy is estimated better than the teacher, or the performance is similar. On the other hand, the TECA for long-activation appliances progressively reduces with the student architecture reduction due to the slight progressive degradation of either precision and recall, especially for student 1H-16U, for which the activations are underestimated for Washing Machine and overestimated for the Dishwasher. This result shows the variability of performance depending on the student structure for long-activation appliances influenced by the appliances' characteristics that are very different between the two domains in terms of power values and duration. With shallow architectures, transfer learning process does not sufficiently improve the model. When the pretraining is performed with REFIT, the TECA is either similar or improved for long-activation appliances, because of data statistical similarity between the source and target environment in this case, except for Washer Dryer. The same holds for short-activation appliances, with a decrease of only 0.8%.

In summary, we observe that in the same domain the proposed method reduces the complexity and improves the performance (Table IV). When domains are different, the performance is similar but the complexity is significantly reduced (Table V). The proposed method reduces the complexity and maintain acceptable performance, reducing, in the best case, 86× the FLOPs, and 10× the number of parameters.

C. Comparison With Benchmark Methods

Tables VIII and IX report the results of the proposed method compared to benchmark approaches. For EdgeNILM we also reported the results of the model before pruning, and we included the teacher performance to facilitate evaluation and comparison the methods.

In both pretraining domains, the proposed approach outperforms the benchmark methods on average and for almost all the appliances. The Kettle is the only exception, where the LightweightCNN and pruned EdgeNILM achieve slightly better F_1-score, respectively, when trained using the U.K.-DALE and REFIT datasets.

Pruning improved the performance of EdgeNILM on the Kettle and Washer Dryer appliances when pretrained with D_1 = REFIT, but the performance of the other appliances remained relatively stable. On average, the performance of EdgeNILM Pruned 60% are worse than EdgeNILM Unpruned.

LightweightCNN demonstrated better performance on average for all the appliances compared to EdgeNILM, in particular for Microwave, Washing Machine, and Dishwasher. Instead, compared to WL-NILM, LightweightCNN is less effective for all the appliances except the Washer Dryer. Nonetheless, the proposed student network has a higher F_1-score compared to EdgeNILM Pruned 60%, LightweightCNN, and WL-NILM with an absolute increment of 0.38, 0.29, and 0.30, respectively.

TABLE X
MODEL SIZE (MB) AND FLOPS (M) FOR THE BENCHMARK METHODS AND THE PROPOSED APPROACH

Model	Size (MB)	FLOPS (M)
EdgeNILM Unpruned [14]	82.92	38.54
EdgeNILM Pruned 60% [14]	13.38	6.28
LightweightCNN [18]	12.78	6.12
WL-NILM [7]	0.55	267.8
Teacher	1.10	324.2
Student 1H-32U	**0.172**	**3.6**

Model size and the number of FLOPs are calculated on all the networks used to classify $K = 6$ appliances.

When pretrained with $D_1 = \text{U.K.-DALE}$, the differences among domains has a greater impact on EdgeNILM and WL-NILM, which show low performance for all the appliances except the Kettle. In particular, for EdgeNILM Pruned 60%, Washing Machine and Washer Dryer are not reported because the model was not able to learn with a high pruning percentage. Except for Kettle and Dishwasher, WL-NILM produces poor results like EdgeNILM for all other appliance. For LightweightCNN, performance only slightly decreases with respect to the other pretraining domain. Also in this domain, our approach is more effective on average, with an absolute increment of 0.44, 0.14, and 0.31, on EdgeNILM Pruned 60%, LightweightCNN, and WL-NILM respectively. Particularly for EdgeNILM and WL-NILM, the absence of transfer learning in the complexity reduction process largely affects the performance on a different domain.

Table X reports the model size and the FLOPs for each approach, considering the total number of networks involved in the classification of $K = 6$ appliances. It is worth noting that EdgeNILM pruned 60% and LightweightCNN have almost the same number of FLOPs and model size, although the latter approach has shown better performance. Instead, WL-NILM has a higher number of FLOPs compared to EdgeNILM and LightweightCNN, with performance that varies depending on the pretraining domain. Nonetheless, the proposed student has a number of FLOPs 1.74, 1.7, and 74.4 times smaller than EdgeNILM Pruned 60%, LightweightCNN, and WL-NILM respectively, despite using a larger or equal window dimension than the benchmark methods, a parameter that affects the number of FLOPs (Table III). Note that the model size of the proposed approach is 78, 74, and 3 times smaller than the benchmarks, while reporting superior performance. Considering both, the complexity of the architecture and the performance, the proposed student network is more efficient and effective than the benchmark methods in appliance classification. Fig. 5 shows a complexity-performance comparison among the benchmarks and the proposed method, where the circle dimension is proportional to the mean F_1-score computed on both D_1 pretraining datasets. WL-NILM and EdgeNILM Unpruned are on the opposite side of the plane, remarking that the difference in terms of FLOPs is mainly related to the window dimension of WL-NILM that is around 25 times wider. On the other hand, although our student network has the same window dimension, the number of FLOPs is largely reduced compared to WL-NILM while producing better predictions. Considering the model size, the same can be highlighted compared to the other approaches, that present larger sizes with lower performance.

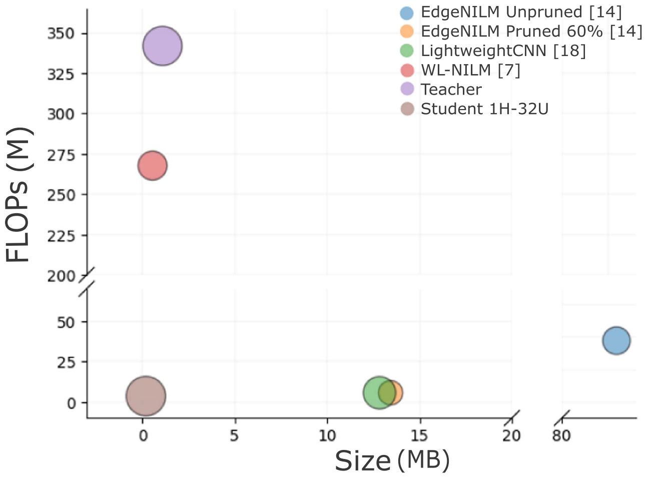

Fig. 5. Complexity-Performance comparison among benchmark methods and the proposed method. For each approach the dimension of the circle is proportional to the mean F_1-score of both D_1 scenarios. “FLOPs” are expressed in Millions (M) and “Size” is expressed in megabytes (MB).

VII. CONCLUSION

In this work, a joint complexity reduction and transfer learning approach for NILM was proposed to provide scalability and improve the performance on unseen target data domains. We adopted a teacher–student KD strategy, using weak supervision to reduce the labeling effort. We analyzed the teacher structure and proposed a distillation framework progressively reducing the complexity. To the best of our knowledge, this is the first study that combines KD and transfer learning to reduce deep neural network models’ complexity and improve their performance on unseen domains for multilabel appliance classification. The method was demonstrated to be effective in reducing complexity and maintain acceptable performance. Evaluated in two different practical scenarios, the method reduced the number of network parameters up to ten times compared to the teacher while maintaining performance. Moreover, we demonstrated that our approach is more effective and efficient compared to benchmark methods.

Future work will focus on designing a distillation method to alleviate the incorrect knowledge transferred from the teacher to the students using explainability tools. In addiction, the method will be considered jointly with the active learning procedure to increase the efficacy of the network in the deployment environment.

ACKNOWLEDGMENT

For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

REFERENCES

[1] Y. Kabalci, “A survey on smart metering and smart grid communication,” *Renewable Sustain. Energy Rev.*, vol. 57, pp. 302–318, 2016.
[2] E. Commission, D.-G. for Energy, C. Alaton and F. Tounquet, *Benchmarking Smart Metering Deployment in the EU-28: Final Report*. Brussels, Belgium: European Commission, 2020.

[3] Z. Su et al., "Secure and efficient federated learning for smart grid with edge-cloud collaboration," *IEEE Trans. Ind. Inform.*, vol. 18, no. 2, pp. 1333–1344, Feb. 2022.

[4] F. Luo, G. Ranzi, W. Kong, G. Liang, and Z. Y. Dong, "Personalized residential energy usage recommendation system based on load monitoring and collaborative filtering," *IEEE Trans. Ind. Inform.*, vol. 17, no. 2, pp. 1253–1262, Feb. 2021.

[5] P. A. Schirmer and I. Mporas, "Non-intrusive load monitoring: A review," *IEEE Trans. Smart Grid*, vol. 14, no. 1, pp. 769–784, Jan. 2023.

[6] Y. Yang, J. Zhong, W. Li, T. A. Gulliver, and S. Li, "Semisupervised multilabel deep learning based nonintrusive load monitoring in smart grids," *IEEE Trans. Ind. Inf.*, vol. 16, no. 11, pp. 6892–6902, Nov. 2020.

[7] G. Tanoni, E. Principi, and S. Squartini, "Multi-label appliance classification with weakly labeled data for non-intrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 14, no. 1, pp. 440–452, Jan. 2023.

[8] D. Li et al., "Transfer learning for multi-objective non-intrusive load monitoring in smart buildings," *Appl. Energy*, vol. 329, 2022, Art. no. 120223.

[9] L. D. S. Nolasco, A. E. Lazzaretti, and B. M. Mulinari, "DeepDFML-NILM: A new CNN-based architecture for detection, feature extraction and multi-label classification in NILM signals," *IEEE Sensors J.*, vol. 22, no. 1, pp. 501–509, Jan. 2022.

[10] S. Singh and A. Majumdar, "Multi-label deep blind compressed sensing for low-frequency non-intrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 13, no. 1, pp. 4–7, Jan. 2022.

[11] C.-H. Hur et al., "Semi-supervised domain adaptation for multi-label classification on nonintrusive load monitoring," *Sensors*, vol. 22, no. 15, 2022, Art. no. 5838.

[12] L. Wang, S. Mao, and R. M. Nelms, "Transformer for nonintrusive load monitoring: Complexity reduction and transferability," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18987–18997, Oct. 2022.

[13] G. Tanoni, E. Principi, L. Mandolini, and S. Squartini, "Weakly supervised transfer learning for multi-label appliance classification," in *Applied Intelligence and Informatics*. Berlin, Germany: Springer, 2022, pp. 360–375.

[14] R. Kukunuri et al., "EdgeNILM: Towards NILM on Edge devices," in *Proc. 7th ACM Int. Conf. Syst. Energy-Efficient Buildings, Cities, Transp.*, 2020, pp. 90–99.

[15] Y. Zhang et al., "FedNILM: Applying federated learning to NILM applications at the edge," *IEEE Trans. Green Commun. Netw.*, vol. 2400, no. 2, pp. 857–868, Jun. 2023.

[16] J. Barber et al., "Lightweight non-intrusive load monitoring employing pruned sequence-to-point learning," in *Proc. 5th Int. Workshop Non-Intrusive Load Monit.*, 2020, pp. 11–15.

[17] S. Ahmed and M. Bons, "Edge computed NILM: A phone-based implementation using MobileNet compressed by tensorflow lite," in *Proc. 5th Int. NILM Workshop*, 2020, pp. 44–48.

[18] W. Luan et al., "Leveraging sequence-to-sequence learning for online non-intrusive load monitoring in edge device," *Int. J. Elect. Power Energy Syst.*, vol. 148, 2023, Art. no. 108910.

[19] J. Lin, J. Ma, J. Zhu, and H. Liang, "Deep domain adaptation for non-intrusive load monitoring based on a knowledge transfer learning network," *IEEE Trans. Smart Grid*, vol. 13, no. 1, pp. 280–292, Jan. 2022.

[20] M. D'Incecco, S. Squartini, and M. Zhong, "Transfer learning for non-intrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1419–1429, Mar. 2020.

[21] B. Peng, L. Qiu, T. Yu, L. Zhong, and Y. Liu, "Incorporating knowledge distillation into non-intrusive load monitoring for hardware systems deployment," in *Proc. IEEE 5th Conf. Energy Internet Energy Syst. Integration*, 2021, pp. 3054–3058.

[22] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 1–9.

[23] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *Int. J. Comput. Vis.*, vol. 129, no. 6, pp. 1789–1819, 2021.

[24] L. Serafini, G. Tanoni, E. Principi, S. Spinsante, and S. Squartini, "A multiple instance regression approach to electrical load disaggregation," in *Proc. 30th Eur. Signal Process. Conf.*, 2022, pp. 1666–1670.

[25] K. Basu, V. Debusschere, S. Bacha, U. Maulik, and S. Bondyopadhyay, "Nonintrusive load monitoring: A temporal multilabel classification approach," *IEEE Trans. Ind. Inform.*, vol. 11, no. 1, pp. 262–270, Feb. 2015.

[26] S. Sykiotis et al., "Performance-aware NILM model optimization for edge deployment," *IEEE Trans. Green Commun. Netw.*, vol. 32, no. 1, pp. 1434–1446, Sep. 2023.

[27] C. Zhang et al., "Sequence-to-point learning with neural networks for non-intrusive load monitoring," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 2604–2611.

[28] M. H. Phan, Q. Nguyen, S. L. Phung, W. E. Zhang, T. D. Vo, and Q. Z. Sheng, "CompactNet: A light-weight deep learning framework for smart intrusive load monitoring," *IEEE Sensors J.*, vol. 21, no. 22, pp. 25181–25189, Nov. 2021.

[29] R. Liu and Y. Chen, "Learning task-aware energy disaggregation: A federated approach," in *Proc. IEEE Conf. Decis. Control*, 2022, pp. 4412–4418.

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

[31] D. Murray et al., "An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study," *Sci. Data*, vol. 4, no. 1, 2017, Art. no. 160122.

[32] Z. H. Zhou, "A brief introduction to weakly supervised learning," *Nat. Sci. Rev.*, vol. 5, no. 1, pp. 44–53, 2018.

[33] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7130–7138.

[34] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[35] Y. Wang, J. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 31–35.

[36] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst.*, 2017, pp. 1597–1600.

[37] C. Klemenjak et al., "Investigating the performance gap between testing on real and denoised aggregates in non-intrusive load monitoring," *Energy Inform.*, vol. 4, 2021, Art. no. 3.

[38] J. Z. Kolter et al., "REDD: A public data set for energy disaggregation research," in *Proc. Workshop Data Mining Appl. Sustainability*, 2011, pp. 59–62.

Giulia Tanoni (Graduate Student Member, IEEE) was born in Recanati, Italy, in 1994. She received the bachelor's and M.S. degrees (with honors) in biomedical engineering, in 2018 and 2020, respectively, from the Università Politecnica delle Marche, Ancona, Italy, where she is currently working toward the Ph.D. degree in edge-centric computing with the Department of Information Engineering.

Her research interests include focusing on deep learning algorithms for smart energy systems.

Lina Stankovic received the B.Eng. degree Honors in electronic communications engineering and the Ph.D. degree in electronic communications engineering from Lancaster University, in 1999 and 2003, respectively.

She is currently a Reader (equivalent to Full Professor) with the Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, U.K. She has been particularly recognized for her contributions to the non intrusive load monitoring problem, having supervised the release of the widely used REFIT dataset, organizing NILM special sessions at IEEE International Conference on Acoustics, Speech and Signal Processing '23, ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation'20 and 2022, guest editing a Special Issue with 11 research articles on Practical NILM approaches with meaningful performance evaluation in the *Sensors* journal and publishing highly cited NILM papers. Her research interests include Signal and Information Processing in the AI era, specializing on representation, signal processing, temporal and spatial information mining, and data management for a range of signal types acquired from sensors including electrical signals, seismic/geoscience, health/biological and other environmental sensor data.

Dr. Stankovic was an Associate Editor for IEEE Transactions on Signal and Information Processing over Networks in 2018–2021.

Vladimir Stankovic (Senior Member, IEEE) received the Dr.- Ing. (Ph.D.) degree in computer engineering from the University of Leipzig, Leipzig, Germany, in 2003.

From 2003 to 2006, he was with Texas A&M University, College Station, TX, USA, first as Research Associate and then as a Research Assistant Professor. From 2006 to 2007, he was with Lancaster University, Lancaster, U.K. Since 2007, he has been with the Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, U.K., where he is currently a Professor. He has coauthored five book chapters and over 200 peer-reviewed research papers, including over 80 journal publications. His research interests include signal and information processing, hybrid intelligence, and interpretable machine learning with applications to smart home energy management, seismic signal analysis, computer vision, and personal healthcare.

Dr. Stankovic was an Associate Editor for IEEE Communications Letters and IEEE Transactions on Image Processing and Area Editor for Elsevier *Signal Processing: Image Communication*. He serves as Associate Editor-in-Chief for IEEE Transactions on Circuits and Systems for Video Technology and Area Editor for IEEE Transactions on Communications. He served as a Technical Program Committee co-Chair of European conference on signal processing and offers a comprehensive technical program (Eusipco) 2012 and General Chair of IEEE Multimedia Signal Processing Workshop 2017. He gave a number of tutorials at major conferences, including IEEE International Conference on Communications, IEEE International Conference on Acoustics, Speech, and Signal Processing, Eusipco, and IEEE Energy Conference.

Stefano Squartini (Senior Member, IEEE) was born in Ancona, Italy, in March 1976. He received the Italian Laurea degree (Honors) in electronic engineering and the Ph.D. degree in electronics and telecommunications from the Polytechnic University of Marche (UnivPM), Ancona, Italy, in 2002 and 2005, respectively.

In 2007, he was an Assistant Professor of Electrical Engineering with the Department of Information Engineering, UnivPM, where he has been a Full Professor, since 2020. He has authored or coauthored three international patents and more than 250 international scientific papers. His research interests include the area of computational intelligence and digital signal processing, with a special focus on audio processing and energy management.

Dr. Squartini was an Associate Editor for IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Cybernetics, and IEEE Transactions on Emerging Topics in Computational Intelligence. He joined the organizing and the technical program committees of more than 90 international conferences and workshops.

Emanuele Principi (Member, IEEE) was born in Senigallia, Italy, in 1978. He received the Italian Laurea with honors in electronic engineering and the Ph.D. degree in electronic, informatics, and telecommunications enginnering from the Università Politecnica delle Marche, Ancona, Italy, in 2004 and 2009, respectively.

He is currently a tenure track Assistant Professor of Electrical Engineering, from December 2019, with the Department of Information Engineering, Università Politecnica delle Marche, Ancona, Italy. He has authored and coauthored of many international scientific peer-reviewed articles. His research interests include digital signal processing and computational intelligence, with a special focus on smart grids, and audio processing.

Dr. Principi is an Associate Editor for *Neural Computing and Applications* and *Artificial Intelligence Review* both edited by Springer, from 2017. He joined the organizing and technical committees of several international conferences. He is member and secretary of the Adriatic section of the Italian Association of Electrotechnics, Electronics, Automation, Computer Science and Telecommunications. As of January 2021, he has been the chair of the IEEE CIS Task Force on Computational Audio Processing.

Open Access provided by 'Università Politecnica delle Marche' within the CRUI CARE Agreement