Knowledge Distillation for Scalable Non-Intrusive Load Monitoring

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

Abstract—Smart meters allow the grid to interface with individual buildings and extract detailed consumption information using Non-Intrusive 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 paper proposes a knowledge distillation approach for NILM, in particular for multi-label appliance classification, to reduce model complexity and improve generalisation on unseen data domains. The approach uses weak supervision to reduce labelling effort, which is useful in practical scenarios. Experiments, conducted on UK-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 F1-score 0.14 higher than a benchmark model 78 times more complex.

Index Terms—Deep Learning, Knowledge Distillation, Weak Supervision, Multi-label Appliance Classification, Non-Intrusive Load Monitoring.

I. INTRODUCTION

ADVANCED Metering Infrastructure (AMI) 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 personalised 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 Non-Intrusive 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 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]–[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]–[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 characterised 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 factorisation, [14], [16], weights quantisation [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 labelled datasets, and this knowledge is transferred to unseen buildings.

This work proposes a method based on KD and weak supervision to reduce the complexity of a neural network for multi-label 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

G. Tanoni, S. Squartini and E. Principi are with the Department of Information Engineering, Universita Politecnica delle Marche, Italy. (e-mail: g.tanoni@pm.univpm.it; e.principi,s.squartini@univpm.it).

L. Stankovic and V. Stankovic are with the Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, United Kingdom. (e-mail: {lina.stankovic,vladimir.stankovic}@strath.ac.uk.)
data directly from the target environment to fine-tune pre-trained 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 multi-label 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 centralised 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 paper is organised as follows. Section II reviews the existing literature for multi-label 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 the paper and presents future developments.

II. RELATED WORK AND CONTRIBUTIONS

This section presents a brief review of multi-label 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 paper.

A. Multi-label Appliance Classification

Multi-label 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 multi-label 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 multi-label state and event type classification. Deep Blind Compressed Sensing is proposed in [10] for multi-label device state detection.

Semi-supervised learning strategies have also been proposed for NILM, using a teacher-student framework [6]. Recently, in [11], a semi-supervised KD approach has been proposed to improve the transferability on target environments. Differently, in [7], the authors adopt a weakly supervised multi-label approach to reduce the labelling effort to train a CRNN, using both weakly labelled data (labels provided for a group of consecutive samples, e.g., for a 4 hours period) and strongly labelled data (i.e., labelled 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 [14–16], [26], tensor decomposition [14] and coefficient quantisation [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 unlabelled data from the target domain. Sykiotis and colleagues [26] presented an edge optimisation framework that applies coefficient quantisation 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 quantisation, where the precision is reduced from 32-bit to 8-bit. Peng and colleagues [21] presented a framework based on KD to obtain a Multi-Layer 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 multi-class 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 [14–17], [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:

- complexity reduction based on knowledge distillation has never been addressed for multi-label appliance classification, i.e., when more than one appliances can be active at the same time instant;
- transfer learning has not been previously addressed jointly with complexity reduction, particularly in the knowledge distillation framework, and for multi-label appliance classification;
- weak labels have never been used in knowledge distillation, particularly for complexity reduction on the multi-label appliance classification task.

To fill these gaps, this paper proposes a multi-label 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 unlabelled 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, labelled 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 labelling effort compared to strong labels and improved performance compared to unlabelled 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: soft labels from the fine-tuned teacher (which may not be entirely accurate) and the ground-truth weak labels of the target data domain. Using the pre-trained and fine-tuned teacher, we also improve the convergence of knowledge distillation and student learning, mitigating performance degradation.

The experimental evaluation has been conducted on two public datasets, UK-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 knowledge distillation to reduce architecture complexity jointly with transfer learning for multi-label appliance classification. Specifically, the contributions of our work are the following: (i) 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); (ii) 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); (iii) integrating weak labels in the distillation framework to improve generalisation 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 modelled as the sum of all $M$ power loads of the building plus noise $\epsilon(t)$, from measurement error and unknown loads:

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

where $x_m(t)$ is the power consumed by appliance $m$ at the time instant $t$. In multi-label 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 $j$-th window is represented by the vector:

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

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

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

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

IV. Proposed Methodology

Figure 1 shows the proposed KD framework. Two learning phases, Pre-training 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 $\{y_j, S_j, w_j\} \in D_1$. Then, the network is fine-tuned on a smaller set $\{y_j, w_j\} \in D_2$ 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 pre-training 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 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 $J$
Knowledge distillation for scalable non-intrusive load monitoring

A. Teacher Learning

The Teacher model implements the function $g_\phi(\cdot)$ with parameters $\phi$ and it is initially pre-trained using both strongly and weakly labelled data, i.e., the dataset $D_1$. The loss function is defined as:

$$L_{pt} = L_s + \lambda L_w,$$

where the two losses are the Binary Cross-Entropy (BCE) function calculated on the strong predictions and on the weak predictions, respectively, as:

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

$$L_w(w_j, \hat{w}_j) = -\frac{1}{K} \sum_{m=1}^{K} w_m \log(\hat{w}_m) + (1 - w_m) \log(1 - \hat{w}_m),$$

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 multi-label multi-class 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 $\gamma$ set to 0.2:

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

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 re-training the entire network.

B. Student Knowledge Distillation

The Student model implements the function $f_\alpha(\cdot)$ with parameters $\alpha$. The weakly labelled 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:

$$L_{dist} = \beta L_{soft} \left( \sigma(\frac{Z_j^{st}}{T}), \sigma(\frac{Z_j^{te}}{T}) \right) + (1 - \beta) L_w(\hat{w}_j^{st}, w_j),$$

where $L_{soft}$ is the BCE, as in (6), calculated on the soft outputs of the student $\hat{S}_j^{st} = \sigma(Z_j^{st}/T)$ and the soft labels $s_j^{st}$. 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.
**Require:** Datasets $D_1$ and $D_2$, Teacher $g_{\phi}(\cdot)$ pre-trained on $D_1$ and fine-tuned on $D_2$, Student $f_{\alpha}(\cdot)$, $\theta(\cdot)$ function to balance losses magnitude.

for $e$ in epochs do
  for each minibatch $B$ do
    $S^{te}_{j\in B} \leftarrow g_{\phi}(y_{j\in B})$;
    $S^{st}_{j\in B}, \hat{w}^{st}_{j\in B} \leftarrow f_{\alpha}(y_{j\in B})$;
    $L_{\text{dist}} \leftarrow \beta L_{\text{softmax}}(S^{st}_{j\in B}, \hat{S}^{te}_{j\in B}) + (1 - \beta)(\theta(e) L_w(\hat{w}^{st}_{j\in B}, w_{j\in B})$;
    Update $\alpha$ using Adam Optimiser to minimise $L_{\text{dist}}$ loss.
  end for
end for

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

from the teacher $\hat{S}^{te}_{j} = \sigma(Z^{te}_{j}/T)$ with $\sigma$ being the sigmoid function, and $Z^{te}_{j}$ and $Z^{st}_{j}$ the logits from the Student and the Teacher, respectively. $L_w$ is the BCE computed on the weak predictions $\hat{w}^{st}_{j}$ of the student and $w_{j}$ the weak ground-truth, as in (7). $\theta(e)$ is a dynamic balance that weights the magnitude of the two losses based on the following formula $\theta(e) = 10^{-G(e)}$, where $G(e)$ is obtained by $G(e) = \log_{10}(L_{w}(e)) - \log_{10}(\sigma_{\text{softmax}}(e))$, $e$ is the training epoch, and $L_{w}(e)$ and $\sigma_{\text{softmax}}(e)$ are the total losses for epoch $e$. $\beta$ 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]. $\beta$ 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 $k_c = 5$, a batch normalisation 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 the 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:

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

(10)

with $F_h = F \cdot h$, and $n_{BN} = 4F_h$ that represents the number of parameters associated to the batch normalisation (2 trainable plus 2 non-trainable). $F_{h-1}$ is the number of feature maps in the input for the $h$-th 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, $N_{\text{CNN}}$ mainly depends on the number of convolutional blocks. The recurrent subpart has a number of parameters $N_{\text{RNN}}$ computed as [36]:

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

(11)

where the last term depends on the used framework and is $2U$ for Keras and PyTorch. $N_{\text{RNN}}$ 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, UK-DALE [30] and REFIT [31], have been used to evaluate the proposed method. UK-DALE contains data from 5 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 UK-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 pre-processed UK-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 pre-training sets $D_1$ for the Teacher: (i) Houses 5, 6, 7, 10, 12, 13, 16, 17, 18 and 19 of REFIT; (ii) Houses 1, 3, 4, and 5 of UK-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 favourable conditions when the pre-training domain is similar to the target data domain. The second allows us to evaluate the method performance when the pre-training and target data domains are statistically different [37]. The validation sets contain 20% of data from each training house. Input data are normalised using the mean and the standard deviation estimated on the pre-training sets.

**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 hours and 15 minutes (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 minutes) after having analysed activations in both pre-training datasets. We identified a total of four window lengths equally distributed from 55 minutes (540 samples) to 4 hours and 15 minutes (2550 samples). We chose a minimum time interval of 55 minutes as it is appropriate for weak labels annotation. Thus, the selected window sizes are 55 minutes (540 samples), 2 hours and 2 minutes (1210 samples), 3 hours and 8 minutes (1880 samples) and 4 hours and 15 minutes (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 \( \beta \) 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]. \( \beta \) and \( T \) have been optimised for each student network based on the validation set that has also been used to find the best threshold to quantise 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 over-fitting. 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 sub-parts. Firstly, 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 size 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.

### D. Benchmark Methods

In the experiments, we compared our method with two existing techniques in the literature that propose complexity reduction for NILM [14, 18] and with [7] that is a recent multi-label 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 multi-label appliance classification. Therefore, [14], [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 multi-label 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 labelled 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 post-processing applied for our method was applied to the raw predictions of benchmark methods, using a threshold for each network optimised 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 \( R = \frac{TP}{TP + FN} \), \( P = \frac{TP}{TP + FP} \), 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 \( F_1 = \frac{2 \cdot P \cdot R}{P + R} \). The load estimation is evaluated using the Total Energy Correctly Assigned (TECA) [38] defined as follows:

\[
\text{TECA} = 1 - \frac{\sum_k \sum_t |\hat{x}_k(t) - \hat{x}_k(t)|}{2 \sum_t \hat{y}(t)},
\]

with \( \hat{y}(t) = \sum_k \hat{x}_k(t), \hat{x}_k(t) \) and \( \hat{x}_k(t) \) 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. Figures 3 and 4 report the Teacher performance after fine-tuning on the target data for different window lengths for aforementioned appliances when pre-training is performed on REFIT and UK-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.

Both figures show that the reduced length window of 540 samples enables more effective detection of the appliances’ states. This is confirmed for both pre-training 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 pre-training 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 UK-DALE for pre-training (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 pre-trained 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 Recall. This is because
Knowledge distillation for scalable non-intrusive load monitoring

Washer Dryer activations in the test set are longer than the activations in REFIT pre-training set (approximately 82 minutes vs 30 minutes). 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, the F1-score decreases by 6.8% due to both Recall and Precision drop after the distillation from the Teacher pre-trained on UK-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 pre-trained on REFIT, the F1-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 pre-training scenarios. This behaviour shows that this architecture helps to improve Student generalisation ability independently of the pre-training set characteristics.

For the Student 1H-16U (N_{RNN} reduced by 93%), the F1-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 pre-trained on UK-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 pre-trained 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 pre-trained on UK-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 structures, with the smallest performance degradation (3% for UK-DALE pre-training) and highest performance improvement (6% for REFIT pre-training) with a reduction of 10x in number of parameters, coherently in both pre-training 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 Table VI and Table 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 pre-trained with UK-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 pre-training 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, 86x the FLOPs, and 10x the number of parameters.

### C. Comparison with Benchmark Methods

Table VIII and Table 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 pre-training 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 UK-DALE and REFIT datasets.

Pruning improved the performance of EdgeNILM on the Kettle and Washer Dryer appliances when pre-trained with

<table>
<thead>
<tr>
<th>Appliance</th>
<th>$F_1$</th>
<th>MM</th>
<th>WM</th>
<th>DW</th>
<th>TOA</th>
<th>WD</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kettle</td>
<td>0.68</td>
<td>0.61</td>
<td>0.62</td>
<td>0.64</td>
<td>0.68</td>
<td>0.62</td>
<td>0.66</td>
</tr>
<tr>
<td>Microwave</td>
<td>0.64</td>
<td>0.61</td>
<td>0.62</td>
<td>0.64</td>
<td>0.68</td>
<td>0.62</td>
<td>0.66</td>
</tr>
<tr>
<td>Dishwasher</td>
<td>0.63</td>
<td>0.61</td>
<td>0.62</td>
<td>0.64</td>
<td>0.68</td>
<td>0.62</td>
<td>0.66</td>
</tr>
<tr>
<td>Washing Machine</td>
<td>0.63</td>
<td>0.61</td>
<td>0.62</td>
<td>0.64</td>
<td>0.68</td>
<td>0.62</td>
<td>0.66</td>
</tr>
<tr>
<td>Microwave Dryer</td>
<td>0.63</td>
<td>0.61</td>
<td>0.62</td>
<td>0.64</td>
<td>0.68</td>
<td>0.62</td>
<td>0.66</td>
</tr>
<tr>
<td>Toaster</td>
<td>0.63</td>
<td>0.61</td>
<td>0.62</td>
<td>0.64</td>
<td>0.68</td>
<td>0.62</td>
<td>0.66</td>
</tr>
<tr>
<td>Student 1H-32U</td>
<td>0.63</td>
<td>0.61</td>
<td>0.62</td>
<td>0.64</td>
<td>0.68</td>
<td>0.62</td>
<td>0.66</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Size (MB)</th>
<th>FLOPS (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LightWeightCNN [18]</td>
<td>12.78</td>
<td>6.12</td>
</tr>
<tr>
<td>WL-NILM [7]</td>
<td>0.55</td>
<td>267.8</td>
</tr>
<tr>
<td>Teacher</td>
<td>1.10</td>
<td>324.2</td>
</tr>
<tr>
<td>Student 1H-32U</td>
<td>0.172</td>
<td>3.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Size (MB)</th>
<th>FLOPS (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>EdgeNILM Unpruned [14]</td>
<td>82.92</td>
<td>38.34</td>
</tr>
<tr>
<td>EdgeNILM Pruned 60% [14]</td>
<td>13.38</td>
<td>6.28</td>
</tr>
</tbody>
</table>

### C. Comparison with Benchmark Methods

Table VIII and Table IX report the results of the proposed approach and benchmark methods trained with $D_2$ = REFIT and tested on REFIT. Best results are reported in bold.

<table>
<thead>
<tr>
<th>Model</th>
<th>$F_1$</th>
<th>MM</th>
<th>WM</th>
<th>DW</th>
<th>TOA</th>
<th>WD</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>EdgeNILM Unpruned [14]</td>
<td>0.81</td>
<td>0.41</td>
<td>0.19</td>
<td>0.13</td>
<td>0.21</td>
<td>0.14</td>
<td>0.39</td>
</tr>
<tr>
<td>EdgeNILM Pruned 60% [14]</td>
<td>0.82</td>
<td>0.29</td>
<td>0.19</td>
<td>0.13</td>
<td>0.11</td>
<td>0.15</td>
<td>0.37</td>
</tr>
<tr>
<td>LightweightCNN [18]</td>
<td>0.74</td>
<td>0.65</td>
<td>0.34</td>
<td>0.62</td>
<td>0.11</td>
<td>0.32</td>
<td>0.46</td>
</tr>
<tr>
<td>WL-NILM [7]</td>
<td>0.74</td>
<td>0.74</td>
<td>0.54</td>
<td>0.43</td>
<td>0.25</td>
<td>0.02</td>
<td>0.85</td>
</tr>
<tr>
<td>Teacher</td>
<td>0.81</td>
<td>0.93</td>
<td>0.67</td>
<td>0.70</td>
<td>0.51</td>
<td>0.67</td>
<td>0.71</td>
</tr>
<tr>
<td>Student 1H-32U</td>
<td>0.80</td>
<td>0.91</td>
<td>0.71</td>
<td>0.72</td>
<td>0.56</td>
<td>0.73</td>
<td>0.75</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>$F_1$</th>
<th>MM</th>
<th>WM</th>
<th>DW</th>
<th>TOA</th>
<th>WD</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>EdgeNILM Unpruned [14]</td>
<td>0.64</td>
<td>0.00</td>
<td>0.41</td>
<td>0.19</td>
<td>0.02</td>
<td>0.23</td>
<td>0.25</td>
</tr>
<tr>
<td>EdgeNILM Pruned 60% [14]</td>
<td>0.68</td>
<td>0.03</td>
<td>-</td>
<td>0.07</td>
<td>0.02</td>
<td>-</td>
<td>0.13</td>
</tr>
<tr>
<td>LightweightCNN [18]</td>
<td>0.75</td>
<td>0.33</td>
<td>0.53</td>
<td>0.53</td>
<td>0.06</td>
<td>0.42</td>
<td>0.43</td>
</tr>
<tr>
<td>WL-NILM [7]</td>
<td>0.72</td>
<td>0.72</td>
<td>0.48</td>
<td>0.48</td>
<td>0.14</td>
<td>0.14</td>
<td>0.24</td>
</tr>
<tr>
<td>Teacher</td>
<td>0.70</td>
<td>0.75</td>
<td>0.69</td>
<td>0.69</td>
<td>0.59</td>
<td>0.59</td>
<td>0.69</td>
</tr>
<tr>
<td>Student 1H-32U</td>
<td>0.71</td>
<td>0.76</td>
<td>0.60</td>
<td>0.63</td>
<td>0.55</td>
<td>0.78</td>
<td>0.57</td>
</tr>
</tbody>
</table>
Knowledge distillation for scalable non-intrusive load monitoring

D1=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 F1-score compared to EdgeNILM Pruned 60%, LightweightCNN and WL-NILM with an absolute increment of 0.38, 0.29 and 0.30, respectively.

When pre-trained with D1=UK-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 pre-training 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 pre-training 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 F1-score computed on both D1 pre-training 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.

VII. CONCLUSIONS AND FUTURE WORK

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 labelling effort. We analysed 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 knowledge distillation and transfer learning to reduce deep neural network models’ complexity and improve their performance on unseen domains for multi-label 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 10 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 addition, the method will be considered jointly with the active learning procedure to increase the efficacy of the network in the deployment environment.

VIII. ACKNOWLEDGMENTS

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


