







OPT-NILM: An Iterative Prior-to-Full-Training Pruning Approach For Cost-Effective User Side Energy Disaggregation

Sotirios Athanasoulas , Stavros Sykiotis , Maria Kaselimi , Anastasios Doulamis , *Member, IEEE*, Nikolaos Doulamis , *Member, IEEE*, Nikolaos Ipiotis 

Abstract—Non-Intrusive Load Monitoring describes the process of analyzing the aggregate household energy consumption to infer the individual energy consumption patterns of different appliances. Although NILM research has led to substantial progress as regards the performance of deep learning models, these models require exhaustive resources for the training phase and, due to their computational demand, are not well suited for deployment on edge devices with limited resources. NILM applications on low-resource devices enhance user adoption, opening up new energy market prospects. Although there has been some work toward edge-computed NILM, the proposed compression frameworks provide a solution only for the deployment phase since they are applied to the already trained models. This study presents OPT-NILM, a novel pruning strategy to discover sub-optimal NILM neural networks before full training, which reduces computing costs for both testing and training phase, and improves disaggregation performance compared to conventional after-training pruning. OPT-NILM proposes a metric to find the appropriate pruning threshold by evenly valuing model performance and computing cost, unlike other approaches that apply compression arbitrarily. Experimental results on the UK-Dale dataset show that the OPT-NILM approach may reduce model trainable parameters by up to 95% with minimal performance loss.

Index Terms—Edge computing, Non-Intrusive Load Monitoring, Pruning, Optimization, Resource Management

I. INTRODUCTION

Electricity load monitoring for appliances is a significant task in light of current economic and ecological trends. It complements home energy management systems (HEMSs) and ambient assisted living (AAL) technologies, contributing to efficient and cost-effective energy management [1], [2]. Additionally, electricity load monitoring serves as a tool for detecting malfunctioning appliances, such as identifying issues like frosting cycles in fridges with damaged seals, among other possibilities. Promoting sustainable living requires householders to adopt energy-related behavior changes. Energy monitoring plays a pivotal role in effective energy management by enabling the monitoring of power consumption of individual appliances, thus informing the planning of technical measures to minimize energy usage. Energy disaggregation techniques can be leveraged to enable granular monitoring of power consumption at the appliance level.

Non-Intrusive Load Monitoring (NILM) or energy disaggregation algorithms aim to infer the energy consumption patterns of domestic appliances by decomposing the aggregated household energy consumption signal into the individual power

signals of its corresponding appliances [3]. Recently, there is a significant number of publications for NILM using deep learning models ([4], [5], [6]). Due to many limitations, NILM approaches have not been widely used in households despite the interest from the industry. Specifically, the training process of such NILM models requires a lot of computational power and resources, so they cannot be deployed on the user side, i.e., on the edge. Instead, they require central servers or cloud computing infrastructures, which increase the cost and energy of running such a service. The current concept implies data transfer between the data source and a central server, which creates privacy problems and data storage costs [7]. Deploying deep learning algorithms on the edge - at consumers' homes equipped with smart meters and low-power devices - could be a viable solution. In order to make this transition from central data processing to user-side energy disaggregation, many different edge-NILM solutions have been proposed. The main goal of all these solutions is to compress and optimize the models' structures to be able to operate with limited computational resources. One of the most common techniques used for NILM model compression is pruning [8], [9].

Pruning is a technique in deep learning that aids in the development of smaller and more efficient neural networks by eliminating unnecessary values in the final trained models' weight tensors based on their contribution to their predictions. The weights and neurons contributions can be determined by local measures such as their magnitude and L1-norm [10]. However, the existing compression frameworks share the basic limitation that they are being applied to a fully trained model, and *they cannot be executed before full training*. Thus, the proposed edge-NILM solutions do not solve the core issues of central data processing in the sense that the network has to initially be fully trained centrally by allocating all the demands of the central server and cloud computing infrastructures. As a result, current compression schemes only provide a solution to the testing phase of NILM algorithms on the edge, which is the least computationally heavy task of the whole process. However, in the machine learning community, there is an increasing interest in a new training trend according to which we achieve training acceleration that embraces the promising training-on-the-edge paradigm.

Here, we propose a prior-to-full-training NILM compression scheme, which allows for the identification of optimal sub-deep NILM networks without first requiring full training of the selected model. Following such a scheme would aid in

dealing with central data processing issues and NILM real-world deployment since we are able to identify efficient sub-NILM models at their initialization stage, eliminating the training resources and creating efficient, lightweight models that would be able to run in limited resource devices. While the training phase of our framework necessitates data transmission to a central node in order to train the identified sub-network, it's essential to underscore that since the deployment is taking place in houses not included in the training set, the testing phase functions without any subsequent data transmission. All inferencing occurs directly on the edge side, bolstering data privacy and promoting user adoption, given that there's no necessity for users to dispatch their consumption data to an external entity.

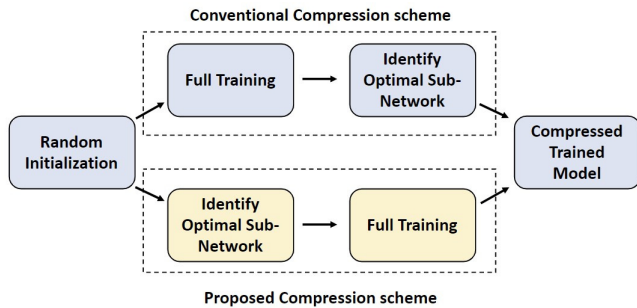


Fig. 1. The comparison of the conventional pruning process (upper) and the proposed OPT-NILM (lower).

These pruned models will be trained using fewer computational resources than the corresponding uncompressed ones and at the same time tested on the edge, utilizing the user's limited resource devices. The main goal of the OPT-NILM is to provide such a framework to optimally identify sub-deep networks before training for a cost-effective user-side NILM. The main contribution of this work is summarized below:

- **Proposing a computationally efficient before-full-training pruning scheme for edge computed NILM.** In contrast with the conventional pruning approaches, the proposed approach identifies optimal sub-deep NILM networks prior to full training. The proposed framework not only identifies sub-deep-neural-network structures that can be easily deployed in a limited resource device, but it also reduces the computational resources needed for the training phase of the NILM models promoting the real-world deployment and adoption of NILM applications.
- **OPT-NILM identifies optimal sub-networks that achieve better disaggregation performance compared to the conventional after-training pruning schemes.** Deep neural networks (DNN) are known to be over-parameterized. Thus, a trained DNN for NILM contains many ineffectual parameters that can be safely pruned or zeroed out with a small or no effect on its performance. In our scheme, where these parameters are pruned before the full training, our sub-deep neural network structures are less overparameterized during the full training, reducing the computational resources needed and preserving a better trade-off between disaggregation performance and reduction in the number of trainable parameters.

- **Proposing a model optimization metric to determine the ideal balance between the model's disaggregation performance and compression.** In NILM applications, the trade-off between accuracy and efficiency is critical. Assuming that we set a high pruning percentage, this results in a significant accuracy drop since the pruned model will not have enough representation power. OPT-NILM is both a resource-efficient and performance-effective technique and introduces an objective model optimization metric for NILM that describes the trade-off between the performance and the model complexity by equally weighting both these factors.

Although the proposed prior-to-full training pruning scheme was inspired by the [11], this work is a pioneering application within the NILM domain. Additionally, this study offers a comprehensive comparison to other compression methods and introduces a novel metric tailored to the unique needs of NILM. From a technical standpoint, the primary contribution of this paper is the introduction of a cost-effective and interoperable deployment strategy for the proposed OPT-NILM inference phase. Our solution is anchored on a Raspberry Pi device and leverages the Z-Wave communication protocol. Originally developed for use in connected home technology, this protocol ensures reliable and robust data transmission between monitored devices and their respective gateways [12]. The paper is structured as follows: Section 2 covers the background on low-frequency NILM with deep learning and compression methods. Section 3 delves into NILM problem formulation and its deep neural network modeling. The proposed solution is detailed in Section 4, while Sections 5 and 6 present and discuss results. Section 7 concludes and outlines future directions.

II. RELATED WORK

In this section, we provide a brief background on deep learning energy disaggregation approaches and a review of the compression approaches used in deep learning, as well as in deep-NILM models specifically.

A. Deep Learning Models for NILM

Deep learning has achieved enormous success in domains such as natural language processing, time-series analysis, and computer vision [13]. Over the last few years, numerous deep learning approaches have been proposed for NILM as it has been proved that they achieve a superior performance [14], including Convolutional neural networks (CNN), recurrent neural networks (RNN), long short-term memory (LSTM), bidirectional (bi)LSTM, gated recurrent unit (GRU) - (bi)GRU, and Transformer models [3]. RNN approaches, such as LSTM and GRU, use feedback connections to capture temporal dependencies within the power signals [15]. Both LSTM and GRU architectures have been widely proposed in NILM [16], [17] since they converge fast and provide a good disaggregation performance. CNN-based architectures capture long-range temporal dependencies in time-series data, making them a successful NILM technique [18]. This strategy requires large model depth and extensive filters, which increases computational complexity. NILM techniques like [19] propose

hybrid recurrent-convolutional architectures, benefiting from the advantages of both types of layers. Transformer-based architectures have become another widespread approach for NILM [15], [20], [21] due to their ability to adopt self-attention mechanism and process data in an order-invariant way. However, all the aforementioned deep learning NILM approaches suffer from computational complexity issues, which increase the training cost and limit their applicability in a real-world deployment on edge.

B. DNN Compression Methods for NILM

Recent developments have driven the adoption of NILM and related energy applications on edge devices. The basic reason for that is that deploying such applications on the edge eliminates the need to transfer data between the users and a central data source, addressing the challenges tied to central data processing and privacy. The landscape of research in edge computed NILM is broad and includes different approaches, from deep learning models on edge devices [8], [9], [22]–[24] to feature extraction [25]–[27], federated learning [28] and hardware-specific optimizations such as Field-Programmable Gate Arrays (FPGAs) [29] and e-Sense device [30]. Since NILM research has mainly been traversed to deep learning techniques, there is a growing interest in works that deal with NILM inference on edge devices to be deployed as part of Home Energy Management Systems [31]. This trend significantly influenced our decision in this paper to delve deeper into the realm of deploying deep learning architectures on resource-constrained devices and explore the existing and new compression methodologies. However, research on compression methodologies on edge-computed NILM models remains limited. In [9], multiple pruning techniques, including magnitude, relative threshold, and entropy-based pruning, are being investigated and applied on NILM CNN sequence-to-point (seq2point) proposed in [23]. These methods are tested on the kettle and dishwasher appliances from the Refit dataset [32]. The application of a quantization approach has also been proposed in [24]. In this work, the same seq2point CNN architecture is being modified from 32-bit float to 8-bit integer model weights. [8] proposes a model compression scheme of a multi-class seq2point CNN using pruning and tensor decomposition. This approach is evaluated on 3 different appliances from UK-DALE, and REDD datasets [33], [34]. In [35], a performance-aware NILM compression technique is proposed, incorporating an after-pruning approach (PAOP) and an after-pruning approach combined with quantization (PAOPQ) tested across four different architectures. Lastly, in [28], the authors introduced a cloud model compression technique suitable for edge implementation of FedNILM. This was achieved by employing filter pruning within the convolutional layers of the chosen deep-learning model. Although the aforementioned works lead the way toward edge inference in NILM, they provide some significant limitations. The basic drawback of these approaches is that the existing compression schemes are applied to already trained models. Thus, the proposed approaches do not overcome the issues raised by the high computational demand of the training phase and provide a

solution only for the testing phase of the NILM models. Another limitation is that [8],[24] and [9] are being employed in a seq2point CNN architecture, which is a computationally inefficient approach since it provides only a midpoint prediction for each window. Since seq2point models are trained to predict the output signal only at the midpoint of the window, they employ a sliding window approach to construct the entire consumption signal, which increases the number of forward passes and, consequently, the computational resources required for inference compared to seq2seq models that predict the entire sequence at once [23]. Finally, in both [9], and [8], compression is applied in an arbitrary way, and there is no framework that evaluates the trade-off between model complexity and performance degradation to define the optimal pruning level.

III. PROBLEM FORMULATION

This section presents NILM problem formulation as well as its modeling using deep neural networks. It also discusses some deep learning-related issues that hinder the real word edge deployment of such an application.

A. NILM Problem Formulation

The concept of non-intrusive load monitoring was first introduced by George W. Hart in 1992 [36]. According to their proposed problem formulation, the aggregate active power of a number of measured appliances $m = 1, \dots, M$ at time $t = 1, \dots, T$ can be formally defined as:

$$x(t) = \sum_{m=1}^M y_m(t) + \epsilon_{noise}(t), \quad (1)$$

where $y_m(t)$ expresses the power consumption of the m -th appliance and $\epsilon_{noise}(t)$ describes the noise originating from the measurement equipment and the appliances that are not sub-metered during the measurement campaign. [14]. The goal of energy disaggregation is to solve the inverse problem in (1) and determine the individual consumption $y_m(t)$ of a selected appliance m at time t based exclusively on the measurement of the aggregate signal $x(t)$.

NILM is considered as a very challenging problem, as power signals do not present any linearity, and the use of each appliance depends on the contextual characteristics of each household. The diverse energy consumption patterns make the implementation of robust NILM algorithms with good generalization behavior even more challenging. Finally, another challenge that NILM models should deal with is the dataset imbalance since every appliance is used with different frequencies and duration.

B. Deep Learning Modelling of NILM

Deep learning for NILM was first introduced in 2015 by Jack Kelly, with major progress on disaggregation performance and generalization capability compared to conventional approaches such as [4], [5]. Solving energy disaggregation using deep neural networks is translated into a non-convex

optimization problem. Specifically, learning in deep neural networks describes the process of calculating the weights of the parameters associated with the various regressions throughout the network. In order to find the parameters that give the best approximation, an objective is needed. Assuming a training set of $v = 1, \dots, V$ values, the objective function $J(\cdot)$ quantifies the distance between the ground truth consumption values, y_n , and the predicted ones, \hat{y}_n , as:

$$J(\theta) = \frac{1}{2} \sum_{v=1}^V \mathcal{L}(\hat{y}_v, y_v) \quad (2)$$

where θ are the model parameters (or weights) and $\mathcal{L}(\cdot)$ is the cost function. Note that in (2) we omit the subscript m as we describe the optimization function of a single device. The minimization process of $J(\cdot)$ takes place through the back-propagation step [37], where gradient descent is applied to update the parameters of the model. Deep neural networks are universal function approximators that are capable of approximating very complicated functions. However, the trade-off of this capability is the number of neurons needed. Specifically, in order to approximate a non-convex function, as it is needed to do in NILM, which is considered a very challenging problem, it requires to use of high-complexity deep learning models with many parameters [38]. Although these models are considered as state-of-the-art approaches toward NILM, they increase the computational complexity and resources required to tackle this problem.

IV. METHODOLOGY

In this part, we describe the suggested OPT-NILM compression strategy as well as the standard after-training magnitude pruning, which has already been employed in [8] [9] as a way to reduce the complexity of NILM deep learning models towards edge inference. In addition, we discuss the methods and benefits of the suggested scheme, highlighting its key contributions to the acceptance and implementation of an edge NILM application in the real world. Lastly, we define a trade-off metric for approximating the optimal pruning threshold in relation to the model's performance.

A. Magnitude Pruning

One of the most common methodologies for optimizing DNN structures is magnitude pruning. The origin of idea of pruning in artificial neural networks derives from synaptic pruning in the human brain, where axons and dendrites decay and die off, resulting in synapse elimination that occurs between early childhood and the onset of puberty [39]. In analogy, deep learning pruning removes redundant parameters or neurons that do not significantly contribute to the model's predictions. Subsequently, model pruning is a technique that reduces the number of the model's weights, $\theta \in \mathbb{R}^K$, to a lower dimensional representation, $\hat{\theta} \in \mathbb{R}^{\hat{K}}$ in which $\hat{K} < K$, by removing non-informative model connections.

Many deep-learning pruning variations have been proposed. Specifically, pruning can either be applied after training or iteratively during the training process [40] [41]. The removal

of connections is performed either in an unstructured way by eliminating specific weights from each layer or in a structured one by removing larger structures such as neurons or convolutional filters [42]–[45]. Finally, pruning approaches remove weights based on different metrics such as weights magnitude, gradients magnitude, layer-wise mutual information, or learned threshold via gradient descent [44], [46], [47].

In this work, we implement a post-training pruning based on L_1 -norm metric as a baseline approach since it has also been used for edge computed NILM in [8] [9]. This approach removes the model's connections with the smallest contribution to its output according to a specified threshold p_{thrs} . Given a dataset $\mathcal{D} = \{(x(t), y_m(t))\}_{t=1}^T$ corresponding to a time window $t = 1, \dots, T$ of measured signal powers and a desired sparsity level p_{thrs} (i.e. the percentage of removed parameters) neural network structural pruning can be formulated as the following constrained optimization problem:

$$\begin{aligned} \min_{\theta_0 \in \mathbb{R}^N} \mathcal{L}(\mathcal{D}; \theta_0) \\ \text{s.t.} \quad \|\theta_0\|_1 \leq p_{thrs} \end{aligned} \quad (3)$$

Here, $\mathcal{L}(\cdot)$ is the defined loss function, θ_0 are the initial weight values and $\|\cdot\|_1$ is the standard L_1 -norm. Thus, after magnitude pruning, the pruned model would only keep the weights with the highest $(1 - p_{thrs})\%$ while the rest will be discarded.

B. OPT-NILM approach

Magnitude pruning removes a percentage of a model's lowest L_1 -norm connections according to a specified p_{thrs} . However, the whole pruning procedure is being applied to an already trained model, meaning that excessive computational resources and data transmission to a central server are required for the training process. The proposed OPT-NILM pruning approach, which deals with the aforementioned limitations, is mainly inspired by the Lottery Ticket Hypothesis paper [11]. According to this work, a randomly-initialized neural network contains a sub-network that is initialized such that - when trained in isolation - it can match the test set accuracy of the original network after training for at most the same number of iterations. The key characteristic of this approach is that pruning is being performed before full training rather than after training, as it is proposed in the existing edge NILM frameworks. Based on this idea, our proposed prior-to-full training pruning technique prunes the NILM networks at the initialization stage. The first step of the proposed approach is to initialize the NILM neural network and train it for a couple of iterations while also keeping track of its initial weights parameters θ_0 . In contrast with full training, where the model should become as accurate as possible, in this stage, we are trying to determine which of the initialized parameters lends themselves to the task. In order to achieve this, the model should only be trained for a couple of iterations, which are significantly less compared to the full training. Subsequently, this slightly trained model is pruned using the same techniques that are used to prune a fully trained model. In this work, the L_1 -norm pruning technique is used to remove the parameters which are not helpful to the task. Since the model is not trained

for a long time this technique gives an indication of not only the current parameters but also of their initialization. Thus, if a parameter is currently ineffective, its initialization is probably not part of the optimal sub-network. The final step is to reset the parameters that were not pruned back to their initialization θ_0 .

The process of training, pruning, and resetting is repeated for $\hat{N} \ll N$, where \hat{N} stands for the epochs of the pretraining cycle and N stands for the epochs of the full training till the desired pruning level has been achieved. Once the optimal sub-network has been found, this network can be trained fully. Figure 2 provides a visual illustration of the process described above.

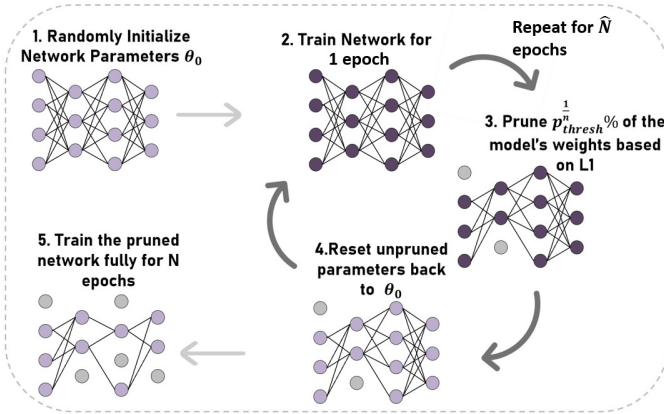


Fig. 2. Overview of the OPT-NILM pruning scheme. Steps 2,3 and 4 consist the pre-training process of finding the optimal sub-networks, and they are repeated till the desired pruning level has been achieved.

From a more mathematical perspective, let $f(x(t)_{t=1}^T; \theta_0)$ be a deep neural network with initial parameters θ_0 . The procedure of the pretraining process is as follows: initially, the network is trained for $\hat{n} = 1, \dots, \hat{N}$ iterations until the first desired θ_1^{Tr} is obtained, where the superscript Tr denotes the training state. This can be described as:

$$\text{Train} : \theta_{\hat{n}}^{\text{Tr}} = F_{\text{train}}(\theta_{\hat{n}-1}^{\text{Rst}}), \quad (4)$$

where $F_{\text{train}}(\cdot)$ is a function describing the training procedure of the network and $\theta_{\hat{n}-1}^{\text{Rst}}$ are weights obtained from the network after the reset state. Afterwards, $p_{\text{thresh}}^{1/\hat{n}}$ % of smallest magnitude weights are being pruned by applying a binary mask $\mu \in \{0, 1\}^K$ such that its initialization is $\theta_1^{\text{Pr}} = \mu_1 \odot \theta_1^{\text{Tr}}$, where \odot denotes the Hadamard (point-wise) multiplication. This is described as:

$$\text{Prune} : \theta_{\hat{n}}^{\text{Pr}} = \mu_{\hat{n}} \odot \theta_{\hat{n}}^{\text{Tr}}, \quad (5)$$

where $\theta_{\hat{n}}^{\text{Pr}}$ are weights obtained from the network after pruning. Then, the remaining weights are reset back to θ_0 as

$$\text{Reset} : \theta_{\hat{n}}^{\text{Rst}} = F_{\text{rst}}(\theta_0, \theta_{\hat{n}}^{\text{Pr}}), \quad (6)$$

where $F_{\text{rst}}(\cdot)$ is a function that replaces the non-zero index values of the pruned network with those of θ_0 . Note that the above described process is repeated for all the \hat{N} epochs. The identified optimal sub-network $f(x(t)_{t=1}^T; \hat{\theta})$ could then be fully trained, employing much fewer computational resources

compared to the original uncompressed model. The proposed OPT-NILM pruning scheme is compactly described in the Algorithm (1).

Algorithm 1 OPT-NILM Compression Scheme

```

Initialize a neural network  $f(x(t)_{t=1}^T; \theta_0)$ 
while  $\hat{n} \leq \hat{N}$  do
    • Train the network for 1 epoch to obtain  $\theta_{\hat{n}}^{\text{Tr}}$ 
    • Prune  $p_{\text{thresh}}^{1/\hat{n}}$  % of the  $\theta_{\hat{n}}^{\text{Tr}}$  by creating a binary mask  $\mu_{\hat{n}}$ 
    • Reset the remaining weights back to  $\theta_0$ ,  $F_{\text{rst}}(\theta_0, \theta_{\hat{n}}^{\text{Pr}})$ 
end while
Fully train the obtained sub-network  $f(x(t)_{t=1}^T; \hat{\theta})$ 
    
```

The proposed pre-training process is able to find optimal computational light sub-networks that could be deployed on a limited resource device and trained using much fewer computational resources, providing a cost-effective embedded NILM solution for the consumers. Furthermore, experimental results show that the proposed OPT-NILM scheme manages to achieve better performance by identifying even smaller sub-deep NILM networks than the conventional pruning scheme. Last but not least, this approach could increase the efficiency and enhance the design of the network by providing information about what an optimal sub-network architecture would look like in terms of layers' importance and the number of initial parameters.

C. Optimal pruning threshold estimation

A basic limitation of the aforementioned works on NILM compression is that $\hat{p}_{\text{thresh}}^{\text{opt}}$ is selected in an arbitrary way without taking into account the performance of the models. This paper proposes a metric that fills this gap and identifies the optimal pruning threshold $\hat{p}_{\text{thresh}}^{\text{opt}}$ for NILM models by equally weighting the trade-off between model complexity and disaggregation performance. This metric incorporates both the performance degradation of the pruned model as well as the gain in terms of parameter reduction. The metric that is being used to find the $\hat{p}_{\text{thresh}}^{\text{opt}}$ is the F1-score as presented in (7).

$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \quad (7)$$

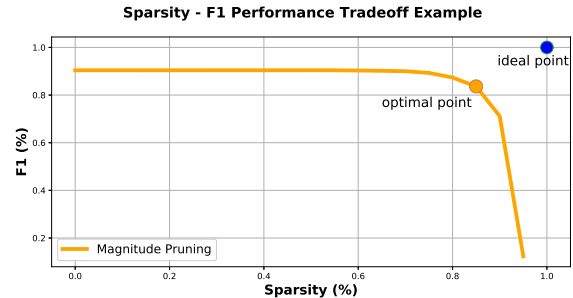


Fig. 3. Example of the proposed trade-off metric. The blue dot denotes the ideal point (sparsity=1.0, $F_1 = 1$ while the orange dot denotes the optimal point of the performance-sparsity curve.

where TP, FP and FN stand for the True Positive, False Positive and False Negative classified time instances in the predicted signature. The reason that F1-score was the selected measure for evaluating the disaggregation performance of the pruning-performance trade-off metric is its ability to assess if the model can properly identify the appliances' activations and address the class imbalance problem of NILM.

Pruning results are presented as an achieved performance against the pruning percentage with values $p_{thrs} \in (0, 0.95)$. The optimal point of such a curve is computed as the point that has the minimal Euclidean distance from the 'ideal' points and whose coordinates are F1-score equal to 1 and pruning percentage equal to 1. This metric using:

$$\hat{p}_{thrs}^{opt} = \arg \min_{p_{thrs} \in (0, 0.95)} (\text{dist}(F1, p_{thrs})) \quad (8)$$

and

$$\text{dist}(F1, p_i) = \sqrt{(1 - F1)^2 + (1 - p_i)^2} \quad (9)$$

where $p_i \in (0, 0.95)$. A visual representation of the proposed trade-off metric is depicted in Figure 3.

Utilizing the performance-sparsity trade-off metric, we are now able to identify the optimal pruning threshold of each pruning technique and use it as a baseline to compare the conventional NILM magnitude pruning with the proposed before-full training NILM pruning scheme. Although different trade-off metrics, such as the performance-sparsity rate of change, could have been used to select the optimal pruning level, the major advantage of the proposed metric is that since the performance and the sparsity axis are in the same scale, it equally weights the performance and the model complexity factors concluding to a fair trade-off metric.

D. Deployment of OPT-NILM to consumer's side

The objective of this paper is to introduce a cutting-edge and cost-effective framework for NILM compression. However, to ensure practical usability and consumer benefits, a deployment scenario is essential. In this regard, we propose a decentralized solution that eliminates data transmission requirements for the inference phase and addresses privacy concerns of the consumers. The developed solution is based on the Z-wave communication protocol, which is ideal for smart home solutions due to its ability to create a mesh network topology, which allows devices to communicate with each other ensuring the reliability and stability of the network as well as better coverage and communication range [7], [48].

To implement our solution, several integral components are employed. We utilize a Z-Wave energy meter, specifically the Aeotec Home Energy Meter Gen 5 [49], which is capable of recording up to 200 amps with an impressive 99% accuracy, in order to monitor and transmit the aggregated consumption data to the OPT-NILM inference service. As a gateway to collect this data and execute the OPT-NILM inference service, the Raspberry Pi Model 4 [50] was used, due to its cost-efficiency, compact design for easy installation, and competency in facilitating Z-Wave communication using the Z-Wave daughter card [51]. To ensure users can conveniently access the appliance-level consumption predictions while safeguarding

data security, we've set up a local host web service, negating the need for transmitting data externally. A comprehensive visual layout of the proposed OPT-NILM inference deployment strategy is depicted in Figure 4. The proposed solution comprises four distinct services all developed and deployed on the edge side. These services are tasked with gathering the aggregate consumption data and producing the disaggregated results.

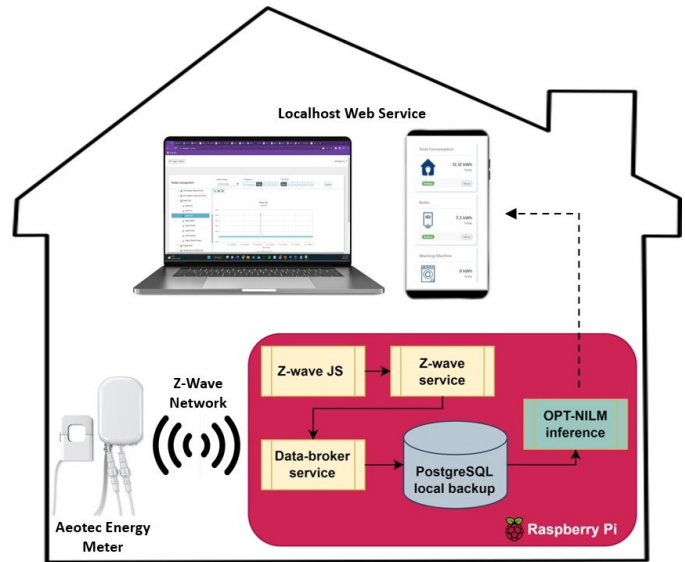


Fig. 4. Proposed deployment architecture based on Z-wave communication protocol.

- **Z-Wave JS:** This is an open-source dockerized service that interfaces with the aggregate consumption smart meter via the Z-Wave protocol. It then transmits the collected data to the Z-Wave service through the MQTT (Message Queuing Telemetry Transport) protocol.
- **Z-Wave service:** This custom service receives the collected data from the Z-Wave JS UI via the MQTT protocol, and subsequently forwards it to the DataBroker service through an API (Application Programming Interface).
- **Data-broker service:** This service is responsible for receiving the data collected by the Z-Wave service and communicates with a local PostgreSQL database. Additionally, the Data-Broker service is tasked with updating (saving and deleting) the collected data in the existing database.
- **OPT NILM inference service:** This service is deployed in a Docker container that runs continuously on the edge device. This service communicates directly with the PostgreSQL database at specified intervals to generate the disaggregation results that they will be visualised through the developed localhost web service.

The demonstrated deployment scenario underscores the practical applicability of our OPT-NILM approach, illustrating its real-world operation. This strategy addresses privacy concerns by keeping all data transmissions confined to the user's side, eliminating the need for external exchanges during the whole inference phase.

V. EXPERIMENTAL SETUP

In this section, we give details related to the experimental setup. Specifically, we give a brief description of the dataset, the selected evaluation metrics as well as the seq2seq model architecture that was used to run our experiments and assess the performance of the proposed pruning scheme.

A. Dataset

A publicly available electrical load measurement dataset - UK-DALE [33] was used to showcase the proposed pruning methodology. UK-Dale consists of aggregate consumption and appliance-level energy consumption measurements from five different houses in the United Kingdom. The dataset was built at the sample rate of 1 Hz or one measurement per second for whole-house and 1/6 Hz or one measurement every six seconds for individual appliance consumption. UK-Dale has been widely used for bench-marking NILM algorithms as it is one of the first open-access datasets at this temporal resolution. In this paper, the appliances used to evaluate and test our algorithms include the kettle, the dishwasher, the washing machine, and the fridge due to their high frequency of use, high consumption, and presence in most houses. Furthermore, another reason for selecting these devices is their different consumption patterns, as the kettle provides an on-off consumption signal, the dishwasher and washing machine have different operational states, leading to a more complicated consumption pattern, and the fridge operates continuously. The aggregate signal was resampled to match the frequency of the appliance-level signals at 1/6 Hz. The models were trained using the data from houses 1,3,4 and 5, and they were tested on unseen data from house 2.

B. Model architecture

To evaluate and test the proposed prior-to-full training pruning scheme, we conducted experiments using a seq2seq CNN model. The model's architecture was inspired by the seq2point CNN, which was proposed in [23], and it was also used by the aforementioned NILM compression approaches. The basic reason that we decided to modify this architecture and use a seq2seq model is that seq2point models are less computationally efficient since they produce only one time-point prediction instead of whole windows requiring much more forwards-pass iterations. The proposed model architecture employs 5 1-D convolutional layers with rectified linear activation functions (ReLU) followed by two linear layers with

ReLU and Sigmoid activations correspondingly. The CNN architecture is shown in Figure 5. The foundational model outlined possesses 22,146,000 trainable parameters and takes up 84 MB of memory. While each model in this study was tailored for a particular appliance, the model's minimal memory footprint posed no issues, especially since it was deployed on a Raspberry Pi 4 with 4GB RAM and a storage capacity of 16 GB. The parameters of the model that were adjusted for optimal training cost include the weights of the convolutional and linear layers of the model architecture described above. Although the proposed pruning technique is designed to be agnostic to specific model architecture, its practical implementation might necessitate some modifications depending on the specific architecture. Our choice of a CNN structure for this work was motivated by the robust compatibility of PyTorch's pruning module with the layers present in our proposed model.

C. Evaluation Metrics

We record three widely used metrics to evaluate model performance. Mean Absolute Error (MAE), Symmetric Mean Absolute Percentage Error (SMAPE) equations (10) and (11), were calculated using the ground truth, y_t , and estimated appliance signature, \hat{y}_t , providing an evaluation of the NILM model regression performance under a specific time window $t = 1, \dots, T$ as

$$\text{MAE} = \frac{1}{T} \sum_{t=1}^T |\hat{y}_t - y_t| \quad (10)$$

and

$$\text{SMAPE} = \frac{2}{T} \sum_{t=1}^T \max \left(\frac{|y_t - \hat{y}_t|}{|y_t| + |\hat{y}_t|}, \epsilon \right) \quad (11)$$

Moreover, F_1 score (7) was also used to assess the model's classification performance. The on-off activations of the appliances were computed by comparing the appliance consumption pattern with the requirements of Table 1. In this study, F_1 score is considered the most important metric, as it captures the model's ability to address the class imbalance, identify the appliances' activations and minimize the false positives. This was also the reason that F_1 score was selected to be used for the \hat{p}_{thrs}^{opt} calculation.

VI. RESULTS

The conducted experiments presented in this section compare the after-training pruning, which has been used in the previous compression NILM frameworks [8], [9], [15] with the proposed OPT-NILM scheme. The results focus on the performance of each technique as well as on the reduction of the model's trainable parameters. It is worth noting that the OPT-NILM approach requires multiple iterations in order to identify the optimal sub-network, which may seem to extend the cumulative training duration. To delve into details, for the conducted experiments, the identification of the sub-networks took 10 cycles of a single epoch each, amounting to 10% of the full training duration that consisted of 100 epochs. Although this might seem a significant time commitment, the results are compelling. Namely, given that the proposed

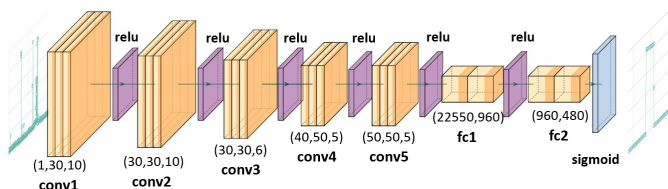


Fig. 5. The proposed CNN seq2seq architecture. The values in CNN layers represents (in_channels,out_channels,kernel_size) while the values in linear layers represents (in_features,out_features).

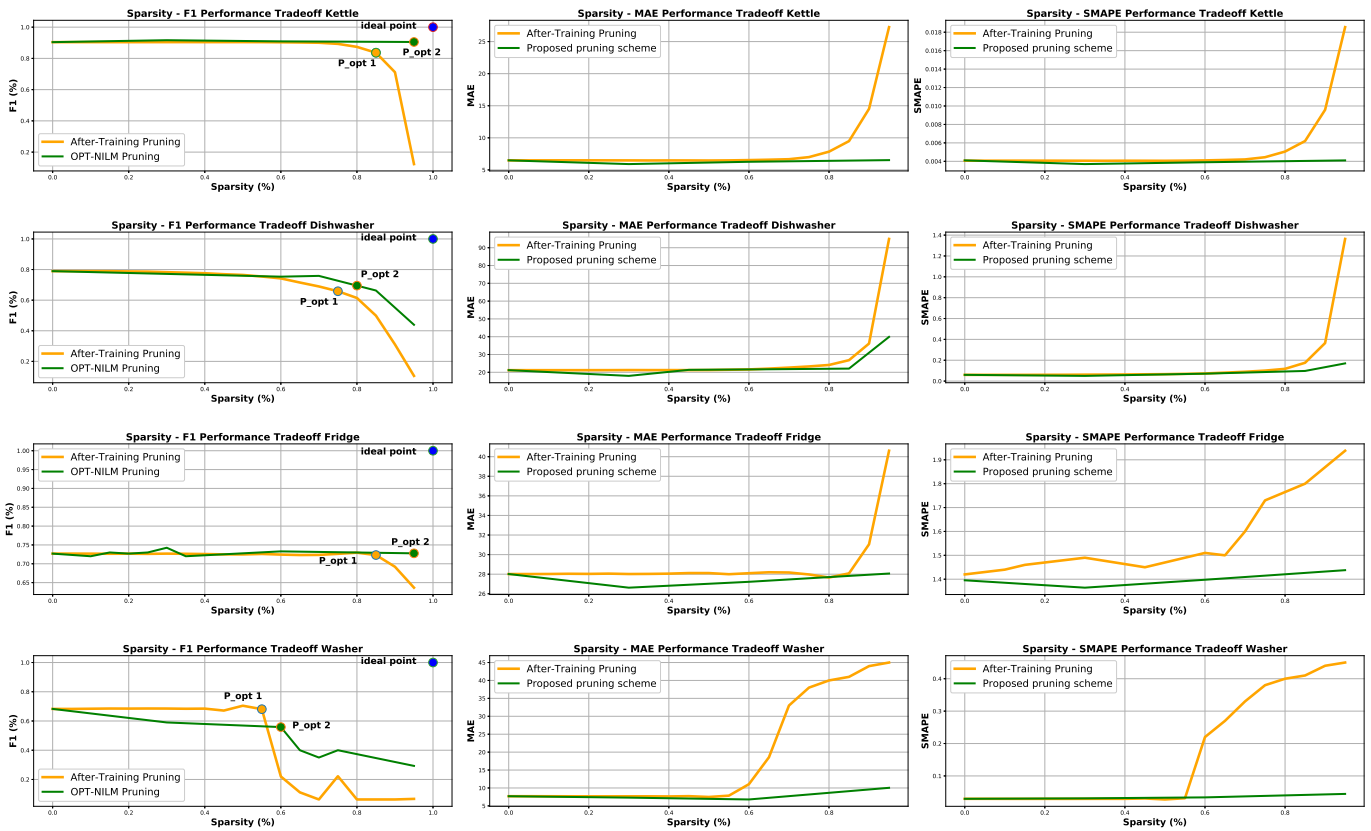


Fig. 6. Pruning threshold vs Performance degradation diagrams. The blue dot indicates the ideal point, while the green and orange dots represent the optimal points based on the proposed trade-off metric.

compression scheme prunes the model’s parameters before training, it establishes itself as an efficient NILM compression framework. This is attributed to its dual benefit: it not only produces optimized models tailored for seamless deployment on edge devices with limited resources, but it also mitigates the computational burden during the initial training phase, given that training is executed on the identified sub-optimal model, thereby diminishing computational expenses.

As can be seen in Figure 6, the performance-pruning level curves indicate that the proposed prior-to-full training pruning can achieve a significantly better disaggregation performance with much fewer trainable parameters than the conventional approach. Specifically, for kettle appliance that only presents an ‘on’ and an ‘off’ state, the performance degradation, when using the proposed pruning scheme, is indiscernible even when the model only presents 5% of the initial weights. On the other side, the impact of parameter pruning is more severe on the dishwasher and washing machine, which have a more complicated consumption signal with more operational states. Finally, OPT-NILM showcases its superiority in fridge appliance where it also manages to sustain a better performance-compression trade-off for all the selected evaluation metrics. To sum up, in all of the tested cases, the proposed pruning technique seems to perform significantly better than the conventional after-training pruning since, for the same pruning levels, it manages to achieve significantly higher performance. This assumption could also be confirmed

by looking at the consumption prediction diagrams in Figure 7, which present the inferred consumption pattern of each appliance for a pruning threshold set to \hat{p}_{thresh}^{opt} of the OPT-NILM approach and compares them with the baseline and after-training pruning approach.

For the kettle appliance, our proposed pruning scheme showcases a superior disaggregation capability even with the pruning level set to 95%, as it manages to infer the corresponding consumption pattern. On the other hand; conventional magnitude pruning does not manage to detect the kettle’s activation function at all, providing a very poor disaggregation performance for the same pruning threshold. Comparing the results of the proposed pruning scheme and the baseline model, we could observe that both prediction curves are very similar to each other even though the pruned model uses only 5% of the parameters of the baseline one. Specifically, the OPT-NILM method surpasses the baseline model, yielding a MAE error of 140 compared to the baseline’s MAE error of 153. For the dishwasher appliance, both techniques manage to infer the appliance’s consumption pattern. However, the conventional after-full training pruning provides many false positive activations contrary to the proposed technique, which successfully predicts both ‘on’ and ‘off’ states.

Comparing the identified sub-network for the dishwasher appliances between the proposed pruning approach and the baseline model, we observe a similar pattern with the kettle appliance, with prediction curves being very similar to each

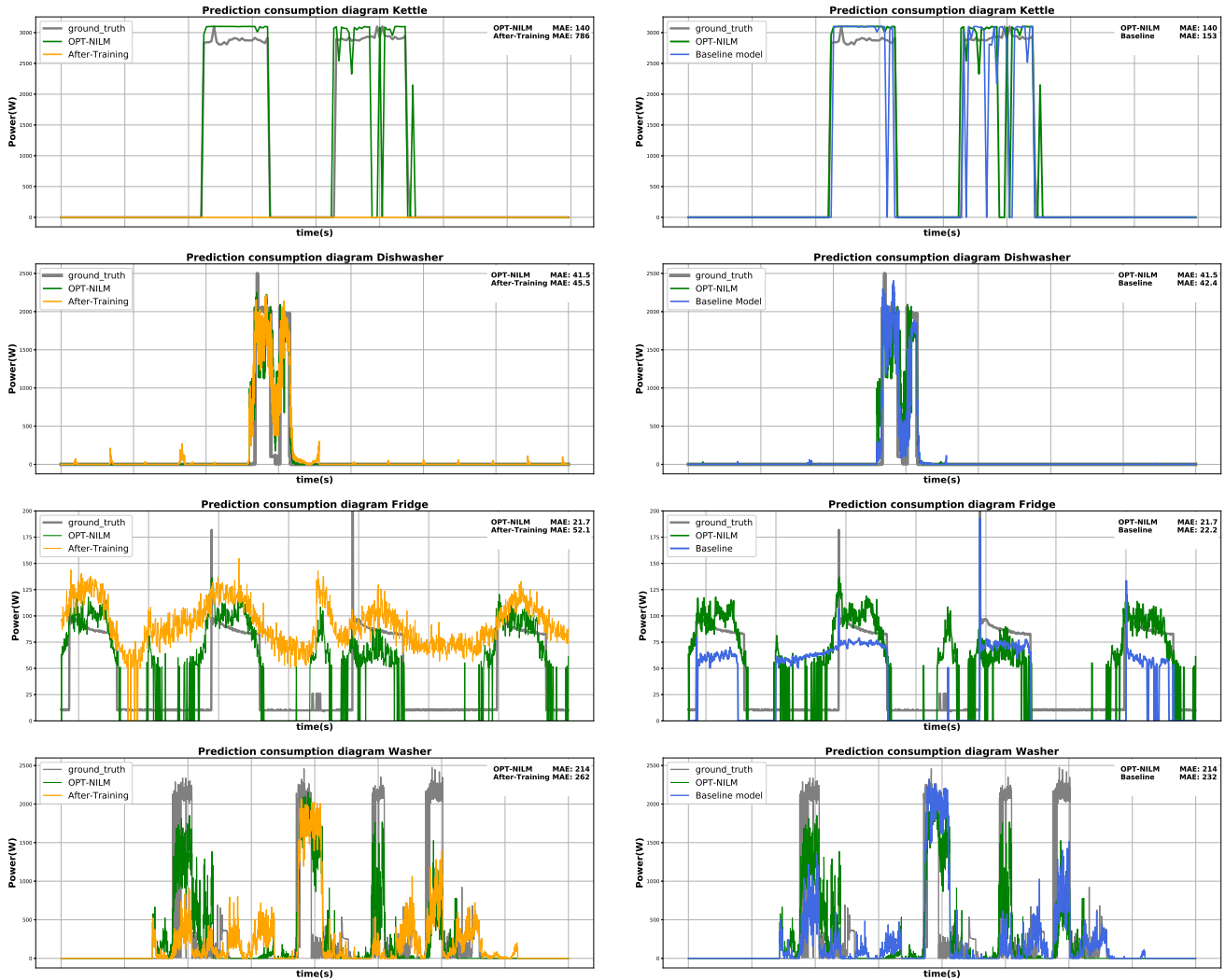


Fig. 7. Prediction consumption diagrams using the OPT-NILM vs the after-training pruning scheme and the OPT-NILM vs the baseline model. The pruning thresholds were set equal to the \hat{p}_{thrs}^{opt} of the OPT-NILM approach both for the OPT-NILM and after-training approaches, 95% for the kettle, 80% for the dishwasher, 85% for the fridge and %60 for the washing machine.

TABLE I
COMPARATIVE EVALUATION RESULTS - DISAGGREGATION PERFORMANCE WITH RESPECT TO COMPRESSION THRESHOLD

Appliance	Approach	Compression metrics			Performance metrics		
		Pruning Percentage (%) $p = \hat{p}_{thrs}^{opt}$	Number of Trainable Parameters	MFLOPs	F1	MAE	SMAPE
Kettle	Baseline	0	22146000	39.27	0.90	6.49	0.004
	After-Training	85	3321900	19.07	0.84	9.81	0.006
	OPT-NILM	95	1107300	15.86	0.90	6.58	0.004
Dishasher	Baseline	0	22146000	39.27	0.79	21.19	0.059
	After-Training	75	5536500	23.86	0.62	25.65	0.127
	OPT-NILM	80	4429200	21.13	0.66	22.07	0.097
Fridge	Baseline	0	2214600	39.27	0.72	28.01	1.39
	After-Training	85	3321900	19.09	0.68	28.31	1.78
	OPT-NILM	95	1107300	15.86	0.71	28.30	1.46
Washer	Baseline	0	22146000	39.27	0.68	7.69	0.029
	After-Training	55	9965700	25.01	0.67	7.85	0.031
	OPT-NILM	60	8858400	25.89	0.60	7.51	0.030

other even though the pruned network uses only 20% of the baseline's parameters. Notably, the OPT-NILM model achieves a MAE of 41.5, whereas the post-training pruning

yields a MAE of 42.4, further demonstrating the former's superior performance. Similar behavior is also observed in the fridge and washing machine appliances, with the OPT-NILM

TABLE II
PERCENTAGE IMPROVEMENT IN COMPRESSION METRICS DURING TRAINING AND INFERENCE PHASE OF THE FINAL PRUNED MODEL.

Phase	Appliance	Approach	Improvement (%) Compression metrics	
			Num of Train Params	MFLOPs
Training	Kettle	After-Training Pruning	0 %	0 %
		OPT-NILM	95 %	60%
	Dishwasher	After-Training	0 %	0 %
		OPT-NILM	80 %	46%
	Fridge	After-Training	0 %	0 %
		OPT-NILM	95 %	60 %
	Washer	After-Training	0 %	0 %
		OPT-NILM	60 %	34%
Inference	Kettle	After-Training	85 %	51 %
		OPT-NILM	95 %	60 %
	Dishwasher	After-Training	75 %	39%
		OPT-NILM	80 %	46%
	Fridge	After-Training	85 %	51%
		OPT-NILM	95 %	60 %
	Washer	After-Training	55 %	36%
		OPT-NILM	60 %	34%

approach managing to perform significantly better than the after-training approach and inferring a consumption pattern very similar to the baseline model for a pruning threshold set to 85% and 60% correspondingly. Based on the prediction consumption diagrams for the washing machine, the OPT-NILM achieved a MAE of 21.7, markedly better than the after-training’s 542.1 and the baseline’s 22.2. A similar trend was observed for the washer appliance, where the OPT-NILM registered a MAE of 214, surpassing the baseline’s 232 and the after-training’s 262. The hypothesis that the suggested pruning approach could result in enhanced disaggregation effectiveness identifying more computationally efficient NILM models compared to traditional after-training pruning is also confirmed by looking at the Table I, which presents the disaggregation performance in regard to the model’s compression. Specifically, according to this table, the proposed technique achieves a better performance-compression trade-off (i.e. high pruning threshold and low-performance degradation) for all the tested appliances. Overall, the proposed OPT-NILM methodology consistently outperforms the traditional after-training pruning techniques and frequently produces comparable or even better disaggregation results than the baseline model. This enhanced performance is attributed to the fact that the proposed pruning approach identifies an optimal sub-structure within the initial network before the training stage, manifesting an augmented generalization capability on unseen data. This stands in contrast to the baseline model, which, due to potential overparameterization, may incorporate extraneous noise that undermines its performance. Conventional after-training pruning, on the other hand, operates under the assumption that low-magnitude weights are inconsequential and, therefore, dispensable. This assumption, however, is not always correct. Some of these low-magnitude weights remain pivotal to the model’s core functionality. Their removal can, hence, significantly impair performance, rendering OPT-NILM a more efficacious alternative.

However, despite the improvement in disaggregation performance, the main contribution of the proposed pruning scheme is the fact the model’s parameters are removed before the full training of the model. This concludes with a more efficient model initialization since the identified sub-network would need much fewer computational resources to be fully trained. Thus, the model’s training cost and computational resources will be dramatically reduced, promoting the real-world deployment and adoption of such a system. The reduction in the model’s complexity is evaluated using the number of trainable parameters as well as the number of floating point operations (FLOPs) required to perform a forward pass. In order to highlight the contribution of the proposed technique, we evaluate the complexity of the pruned model both before the full training and testing phase.

Table II indicates that the proposed pruning method leads to a noteworthy enhancement in computational efficiency. Specifically, the optimal sub-network for the kettle appliance retains just 5% of the initial number of trainable parameters, while the one for the dishwasher appliance retains 20% of the initial number of trainable parameters. For the fridge appliance, the optimal sub-network retains 5% of the initial parameters, and for the washer appliance, it retains 40% of the original model parameters. Similar behavior is also observed in FLOPs parameters, where they also present a significant drop. On the contrary, the conventional magnitude pruning approach does not improve the computational efficiency of the model during the training phase nor on FLOPs or model parameters.

Evaluating the computational complexity of the pruned NILM models for the testing phase, we observe that the proposed pruning technique is also superior in comparison to the standard after-training pruning. In terms of both the number of trainable parameters and FLOPs, the proposed pruning scheme seems to identify more computationally efficient networks that would be able to be deployed in a limited resource device and produce better disaggregation performance.

TABLE III
COMPARATIVE RESULTS WITH OTHER WORKS AMONG ALL TESTED APPLIANCES

Approach	Percentage Change Compression		Percentage Change Performance		
	Trainable parameters	MFLOPS	F1	MAE	SMAPE
Edge-NILM[8]	-75%	-44%	-8.8%	18%	27%
PAOP[35]	-37.5%	-25%	-7.01%	8.3%	15.2%
PAOPQ[35]	-27.5%	-18.1%	-17.7%	6.5%	16.2%
OPT-NILM	-82.5%	-49.8%	-7.4%	1.05%	18%

The superiority of our approach in terms of both computational complexity and disaggregation performance is also demonstrated by comparing OPT-NILM’s overall performance across all tested appliances against two other works,[8] and [35] which employ after-training compression techniques on the same model architecture. The comparative results presented in Table III indicate that our approach achieves a better trade-off between compression and disaggregation performance, surpassing the capabilities of current edge NILM

solutions and offering a more dependable and computationally effective framework for potential consumers.

VII. CONCLUSIONS AND FUTURE WORK

In this work, we have proposed an efficient prior-to-full training pruning scheme for edge deployment of NILM that produces significantly better results than the conventional after-training pruning approach and reduces the computational resources for both the training and testing phase. The proposed pruning scheme not only identifies sub-optimal networks with better disaggregation performance but also assumes a cost-effective NILM deployment since the sub-network structures are identified before the training phase. Finally, we also introduced a trade-off metric to identify the optimal pruning threshold of a NILM model and use it to define a comparable ground between the proposed pruning scheme and the ones that have been used in past edge NILM research works. The experimental findings confirm that the proposed methodology outperforms conventional after-training pruning techniques, not only in terms of disaggregation performance but also in eliminating the computational costs of both training and testing phases, providing a framework for a cost-effective, secure and reliable embedded solution with high potential for the consumer's side. Additionally, OPT-NILM demonstrates an overall superior trade-off between disaggregation performance and compression when compared to other works, further underscoring the effectiveness of the approach. Therefore, the proposed solution presents a cutting-edge approach to edge-based NILM area that holds significant promise for real-world deployment and provides numerous advantages for consumers.

In our future research, we plan to explore additional pruning techniques, such as gradient-based magnitude pruning and information-based pruning, along with evaluating the efficacy of structured pruning. Additionally, we also plan to utilize the versatility of the developed pruning scheme and extend it to other architectures prominent in the NILM domain, like Transformers, LSTM and GRU. Finally, we aim to deploy our solution in real-world settings at a larger scale to assess the replicability of our simulation experiments under real-world conditions.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 955422.

REFERENCES

- [1] B. Buddhahai, W. Wongseree, and P. Rakkwamsuk, "An energy prediction approach for a nonintrusive load monitoring in home appliances," *IEEE Transactions on Consumer Electronics*, vol. 66, no. 1, pp. 96–105, 2020. DOI: 10.1109/TCE.2019.2956638.
- [2] R. Jiao, C. Li, G. Xun, T. Zhang, B. B. Gupta, and G. Yan, "A context-aware multi-event identification method for non-intrusive load monitoring," *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2023. DOI: 10.1109/TCE.2023.3236452.
- [3] M. Kaselimi, E. Protopapadakis, A. Voulodimos, N. Doulamis, and A. Doulamis, "Towards trustworthy energy disaggregation," *Sensors*, vol. 22, no. 15, 2022, ISSN: 1424-8220. DOI: 10.3390/s22155872. [Online]. Available: <https://www.mdpi.com/1424-8220/22/15/5872>.
- [4] K. Srinivasarengan, Y. Goutam, M. G. Chandra, and S. Kadhe, "A framework for nilm using bayesian inference," in *2013 Conference on Innovative Mobile and Internet Services*, IEEE, 2013, pp. 427–432.
- [5] S. Makonin, F. Popowich, I. V. Bajić, B. Gill, and L. Bartram, "Exploiting hmm sparsity to perform online real-time nilm," *IEEE Transactions on smart grid*, vol. 7, no. 6, pp. 2575–2585, 2015.
- [6] S. Athanasoulis, S. Sykiotis, M. Kaselimi, E. Protopapadakis, and N. Ipiotis, "A First Approach Using Graph Neural Networks on Non-Intrusive-Load-Monitoring," in *Proceedings of the 15th PETRA International Conference*, ser. PETRA '22, Corfu, Greece: Association for Computing Machinery, 2022, pp. 601–607, ISBN: 9781450396318. DOI: 10.1145/3529190.3534722.
- [7] Y.-L. Lee, P.-K. Tsung, and M. Wu, "Technology trend of edge ai," in *2018 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, Ambassador Hotel, Hsinchu, Taiwan: IEEE, 2018, pp. 1–2. DOI: 10.1109/VLSI-DAT.2018.8373244.
- [8] R. Kukuluri, A. Aglawe, J. Chauhan, *et al.*, "EdgeNILM: Towards NILM on Edge Devices," in *Proceedings of the 7th ACM Energy-Efficient Buildings, Cities, and Transportation*, ser. BuildSys '20, Virtual Event, Japan: Association for Computing Machinery, 2020, pp. 90–99, ISBN: 9781450380614. DOI: 10.1145/3408308.3427977.
- [9] J. Barber, H. Cuayáhuitl, M. Zhong, and W. Luan, "Lightweight NILM Employing Pruned Sequence-to-Point Learning," in *Proceedings of the 5th International Workshop on NILM*. New York, NY, USA: Association for Computing Machinery, 2020, vol. 1, pp. 11–15, ISBN: 9781450381918. DOI: <https://doi.org/10.1145/3427771.3427845>.
- [10] S. Vadera and S. Ameen, "Methods for pruning deep neural networks," *IEEE Access*, vol. 10, pp. 63 280–63 300, 2022. DOI: 10.1109/ACCESS.2022.3182659.
- [11] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," *arXiv preprint arXiv:1803.03635*, 2018.
- [12] S. Athanasoulis, A. Katsari, M. Savvakis, S. Kalogridis, and N. Ipiotis, "An interoperable and cost-effective iot-based framework for household energy monitoring and analysis," in *Proceedings of the 16th International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '23, Corfu, Greece: Association for Computing Machinery, 2023, pp. 589–595. DOI: 10.1145/3594806.3596541. [Online]. Available: <https://doi.org/10.1145/3594806.3596541>.
- [13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

- [14] P. Huber, A. Calatroni, A. Rumsch, and A. Paice, "Review on deep neural networks applied to low-frequency nilm," *Energies*, vol. 14, no. 9, p. 2390, 2021.
- [15] S. Sykiotis, M. Kaselimi, A. Doulamis, and N. Doulamis, "Electricity: An efficient transformer for nilm," *Sensors*, vol. 22, no. 8, 2022, ISSN: 1424-8220. DOI: 10.3390/s22082926. [Online]. Available: <https://www.mdpi.com/1424-8220/22/8/2926>.
- [16] M. Kaselimi, N. Doulamis, A. Doulamis, A. Voulodimos, and E. Protopapadakis, "Bayesian-optimized bidirectional lstm regression model for nilm," in *ICASSP 2019 - 2019 IEEE (ICASSP)*, 2019, pp. 2747–2751. DOI: 10.1109/ICASSP.2019.8683110.
- [17] N. Batra, R. Kukunuri, A. Pandey, *et al.*, "Towards reproducible state-of-the-art energy disaggregation," in *Proceedings of the 6th ACM Int Conf on Systems for Energy-Efficient Buildings, Cities, and Transportation*, ser. BuildSys '19, New York, NY, USA: Association for Computing Machinery, 2019, pp. 193–202, ISBN: 9781450370059. DOI: 10.1145/3360322.3360844. [Online]. Available: <https://doi.org/10.1145/3360322.3360844>.
- [18] Y. Zhang, G. Yang, and S. Ma, "Non-intrusive load monitoring based on convolutional neural network with differential input," *Procedia CIRP*, vol. 83, pp. 670–674, 2019, 11th CIRP, ISSN: 2212-8271. DOI: <https://doi.org/10.1016/j.procir.2019.04.110>.
- [19] İ. H. Çavdar and V. Faryad, "New design of a supervised energy disaggregation model based on the deep neural network for a smart grid," *Energies*, vol. 12, no. 7, p. 1217, 2019.
- [20] Z. Yue, C. R. Witzig, D. Jorde, and H.-A. Jacobsen, "Bert4nilm," in *5th Workshop on NILM*, ser. NILM'20, Virtual Event, Japan: ACM, 2020, pp. 89–93, ISBN: 9781450381918. DOI: 10.1145/3427771.3429390. [Online]. Available: <https://doi.org/10.1145/3427771.3429390>.
- [21] L. Wang, S. Mao, and R. M. Nelms, "Transformer for nilm: Complexity reduction and transferability," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 18987–18997, 2022. DOI: 10.1109/JIOT.2022.3163347.
- [22] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A Survey of Model Compression and Acceleration for Deep Neural Networks," vol. 35, 2017. DOI: 10.48550/ARXIV.1710.09282.
- [23] Z. et. al., "Sequence-to-point learning with neural networks for nilm," in *Proceedings of the 32nd AAAI Conference on AI*, ser. AAAI'18/IAAI'18/EAAI'18, New Orleans, Louisiana, USA: AAAI Press, 2018, ISBN: 978-1-57735-800-8.
- [24] S. Ahmed and M. Bons, "Edge computed nilm: A phone-based implementation using mobilenet compressed by tensorflow lite," in *5th Workshop on NILM*. New York, NY, USA: ACM, 2020, pp. 44–48, ISBN: 9781450381918.
- [25] E. Tabanelli, D. Brunelli, and L. Benini, "A feature reduction strategy for enabling lightweight non-intrusive load monitoring on edge devices," in *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*, IEEE, 2020, pp. 805–810.
- [26] E. Tabanelli, D. Brunelli, A. Acquaviva, and L. Benini, "Trimming Feature Extraction and Inference for MCU-based Edge NILM: A Systematic Approach," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 943–952, 2022.
- [27] Q. Xu, Y. Liu, and K. Luan, "Edge-Based NILM System with MDMR Filter-Based Feature Selection," in *2022 IEEE 5th International Electrical and Energy Conference (CIEEC)*, IEEE, 2022, pp. 5015–5020.
- [28] Y. Zhang, G. Tang, Q. Huang, *et al.*, "Fednilm: Applying federated learning to nilm applications at the edge," *IEEE Transactions on Green Communications and Networking*, vol. 7, no. 2, pp. 857–868, 2023. DOI: 10.1109/TGCN.2022.3167392.
- [29] A. Hernandez, R. Nieto, D. Fuentes, and J. Urena, "Design of a SoC Architecture for the Edge Computing of NILM Techniques," in *2020 XXXV Conference on Design of Circuits and Integrated Systems (DCIS)*, IEEE, 2020, pp. 1–6.
- [30] R. Gopinath and M. Kumar, "Deepedge-nilm: A case study of non-intrusive load monitoring edge device in commercial building," *Energy and Buildings*, vol. 294, p. 113226, 2023, ISSN: 0378-7788. DOI: <https://doi.org/10.1016/j.enbuild.2023.113226>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0378778823004565>.
- [31] C. Xia, W. Li, X. Chang, F. C. Delicato, T. Yang, and A. Y. Zomaya, "Edge-based energy management for smart homes," in *IEEE*, IEEE, 2018, pp. 849–856.
- [32] D. Murray, L. Stankovic, and V. Stankovic, "An electrical load measurements dataset of united kingdom households from a two-year longitudinal study," *Scientific Data*, vol. 4, p. 160122, Jan. 2017. DOI: 10.1038/sdata.2016.122.
- [33] J. Kelly and W. Knottenbelt, "The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes," *Scientific Data*, vol. 2, Mar. 2015. DOI: 10.1038/sdata.2015.7.
- [34] J. Kolter and M. Johnson, "REDD," in *IN SUSTKDD*, vol. 25, Jan. 2011.
- [35] S. Sykiotis, S. Athanasoulis, M. Kaselimi, *et al.*, "Performance-aware nilm model optimization for edge deployment," *IEEE Transactions on Green Communications and Networking*, pp. 1–1, 2023. DOI: 10.1109/TGCN.2023.3244278.
- [36] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.
- [37] Y. Bengio, Y. Lecun, and G. Hinton, "Deep learning for ai," *Communications of the ACM*, vol. 64, no. 7, pp. 58–65, 2021.
- [38] M. Kaselimi, N. Doulamis, A. Voulodimos, E. Protopapadakis, and A. Doulamis, "Context aware nilm using adaptive bidirectional lstm models," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3054–3067, 2020. DOI: 10.1109/TSG.2020.2974347.

- [39] J. Hawkins, "Special report : Can we copy the brain?" *IEEE Spectrum*, vol. 54, no. 6, pp. 34–71, 2017. DOI: 10.1109/MSPEC.2017.7934229.
- [40] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'15, Cambridge, MA, USA: MIT Press, 2015, pp. 1135–1143.
- [41] G. Castellano, A. M. Fanelli, and M. Pelillo, "An iterative pruning algorithm for feedforward neural networks," *IEEE transactions on Neural networks*, vol. 8, no. 3, pp. 519–531, 1997.
- [42] J. Kruschke and J. Movellan, "Benefits of gain: Speeded learning and minimal hidden layers in back-propagation networks," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, no. 1, pp. 273–280, 1991. DOI: 10.1109/21.101159.
- [43] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," *arXiv preprint arXiv:1608.08710*, 2016.
- [44] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Learning efficient convolutional networks through network slimming," 2017, pp. 2736–2744.
- [45] S. Srinivas, A. Subramanya, and R. V. Babu, "Training sparse neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 455–462. DOI: 10.1109/CVPRW.2017.61.
- [46] K. Azarian, Y. Bhalgat, J. Lee, and T. Blankevoort, "Learned threshold pruning," *arXiv preprint arXiv:2003.00075*, vol. 1, 2020.
- [47] C. Fan, J. Li, T. Zhang, *et al.*, "Layer-wise Model Pruning based on Mutual Information," Nov. 2021, pp. 3079–3090. DOI: 10.18653/v1/2021.emnlp-main.246.
- [48] M. B. Yassein, W. Mardini, and A. Khalil, "Smart homes automation using z-wave protocol," in *2016 International Conference on Engineering MIS (ICEMIS)*, 2016, pp. 1–6. DOI: 10.1109/ICEMIS.2016.7745306.
- [49] *Aeotec home energy meter*, <https://aeotec.com/products/aeotec-home-energy-meter>, 2023.
- [50] *Raspberry pi 4*, <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>, 2023.
- [51] *Z-wave daughter card*, <https://z-wave.me/products/raspberry/slide-2>, 2023.



Stavros Sykiotis received his B.Sc. and M.Sc. from the Technical University of Berlin and is currently an MSCA-ITN fellow and Ph.D. candidate at the National Technical University of Athens. His research interest focuses on machine learning, signal processing techniques, data analysis, and modeling in energy-related applications.



Maria Kaselimi received the Diploma, MSc, and Ph.D. degrees from the National Technical University of Athens (NTUA). Her research interest focuses on machine learning, signal processing techniques, data analysis, and modeling with applications in earth monitoring and the environment. She has 25 papers in international journals and conferences and more than 160 citations. She has been involved in several European research projects as a researcher.



Anastasios Doulamis (Member, IEEE) received the Diploma and Ph.D. degree in Electrical and Computer Engineering from the National Technical University of Athens (NTUA) with the highest honor. Until January 2014, he was an Associate Professor at the Technical The University of Crete is now an Associate Professor at NTUA. He has received several awards in his studies, including the Best Greek Student Engineer, Best Graduate Thesis Award. He has also served as a program committee in several major conferences of IEEE and ACM.



Nikolaos Doulamis (Member, IEEE) received the Diploma and Ph.D. degree in electrical and computer engineering from the National Technical University of Athens (NTUA), both with the highest honor. He is currently a full Professor with the NTUA. He has received many awards (e.g., Best Student among all Engineers, Best Paper Awards). He was an Organizer and TPC at major IEEE conferences. He has authored more than 340 papers in the field of signal processing and machine learning.



Nikolaos Ipiotis holds an MSc in International Business Administration from Westminster University and a BSc in Business Administration from The American College of Greece. Some career highlights include BizDev, and @Velti, a leading global technology provider of mobile marketing and advertising solutions, from startup to Nasdaq. He is currently the CEO of Plegma Labs, an AI - IoT company based in Greece.



Sotirios Athanasoulis received a BSc degree in Computer Science & Artificial Intelligence and MSc degree in Artificial Intelligence & Adaptive Systems from the University of Sussex, Brighton, UK, in 2019 and 2020, respectively. He is currently pursuing his Ph.D. degree at NTUA and Plegma Labs on Machine Learning and Energy Applications as part of the GECKO Marie Curie Innovative Training Network.