

# Towards edge-computed NILM: Insights from a Mediterranean Use Case

Sotirios Athanasoulas

National Technical University of Athens  
Athens, Greece  
sotiriosathanasoulas@mail.ntua.gr

Nikos Temenos

National Technical University of Athens  
Athens, Greece  
ntemenos@gmail.com

Nikolaos Doulamis

National Technical University of Athens  
Athens, Greece  
ndoulam@cs.ntua.gr

Anastasios Doulamis

National Technical University of Athens  
Athens, Greece  
adoulam@cs.ntua.gr

Isidoros Kokos

Intracom Telecom  
Athens, Greece  
isik@intracom-telecom.com

Nikolaos Ipiotis

Plegma Labs  
Athens, Greece  
ni@pleg.ma

**Abstract**—An optimized structured pruning methodology within the context of Non-Intrusive Load Monitoring (NILM) is introduced. The proposed methodology exploits unstructured pruning to determine the optimal sparsity ratio for each layer in the deep neural network model. Subsequently, structured pruning is applied to remove entire units from each layer according to the sparsity values guided by the unstructured pruning. By doing so, important feature information is preserved, resulting in improved classification performance as the pruning threshold is not arbitrarily selected based on a random percentage ratio. Experimental results, evaluated on the Plegma dataset—one of the first datasets from the Mediterranean area capturing local devices and consumption patterns—demonstrate that the proposed methodology significantly optimizes inference performance. Specifically, the approach reduces the baseline model’s MFLOPs by up to 48.85% while keeping a satisfactory disaggregation performance, a stark contrast to the widely used unstructured pruning approach, which does not achieve FLOPs reduction. These findings underscore the potential of edge NILM for promoting flexibility and energy transition in the Mediterranean region, facilitating the broader adoption and implementation of NILM solutions in real-world scenarios.

**Index Terms**—Non-Intrusive Load Monitoring, Energy Disaggregation, Compression, Pruning, Resource Management

## I. INTRODUCTION

To address climate change, the European Commission has implemented policy initiatives aimed at achieving climate neutrality by 2050, with an intermediate target of reducing net greenhouse gas emissions by 55% by 2030 compared to 1990 levels [1]. Central to this objective is enhancing energy efficiency and optimizing energy consumption within buildings, which are responsible for approximately 37% of global energy-related CO<sub>2</sub> emissions [1]. Advanced metering infrastructure plays a pivotal role in this effort by enabling bidirectional communication between utilities and consumers,

This project has received funding from the European Union’s Horizon 2020 and Horizon Europe research and innovation program under the Marie Skłodowska-Curie and ODEON grant agreements, No 955422 and No 101136128, respectively

thereby facilitating the remote management of electricity usage [2]. With smart meter deployment anticipated to reach 80% of European consumers by 2025 [1], the implementation of detailed energy monitoring and techniques becomes essential. These techniques support real-time energy management, identification of malfunctioning appliances, and promote more efficient participation in sustainable energy practices, such as demand response schemes, targeted energy feedback, and tailored pricing policies.

An increasingly popular technique is non-intrusive-load-monitoring (NILM) or energy disaggregation, which identifies the ON-OFF states of appliances and estimates their power consumption based solely on the building’s total meter readings. Recently, NILM research has focused on using Deep Neural Networks (DNNs) [3]–[5], which have achieved impressive results compared to methods based on traditional signal processing and statistics [6], [7]. However, the complexity of DNNs necessitates a centralized data processing scheme due to their high computational demands for both the training and inference phases, leading to increased costs and privacy concerns due to the transfer of sensitive data to external servers. To address these issues and facilitate the shift from centralized data processing to decentralized approaches on edge, the NILM community is now focusing on reducing the computational complexity of deep learning models using various compression techniques. [8]–[12].

Another important point about NILM is that it is inherently a context-aware problem, as the energy consumption patterns of appliances depend on various external and internal factors. These include environmental conditions like weather and seasonality and appliance-specific characteristics such as operational modes and technology types. Although extensive research has been conducted in regions like Northern Europe, the UK, and the USA, there is a lack of research focused on the Mediterranean region [13]–[17]. This region presents unique environmental conditions and appliance usage patterns, such as air conditioners for both cooling and heating and electric boilers, constituting significant and flexible loads [13]. This

gap highlights the need for NILM approaches tailored to the Mediterranean context to improve accuracy and unlock new opportunities for smart grid integration and flexibility.

To address these gaps, our work proposes an edge-based NILM approach that leverages both unstructured and structured pruning techniques. In addition, we utilize the Plegma dataset [13], a newly established dataset from the Mediterranean area, to develop and evaluate our methods. Our approach aims to reduce the computational complexity of deep learning models, making them suitable for deployment on resource-constrained edge devices while maintaining high accuracy. By focusing on the specific conditions and appliance usage patterns of the Mediterranean region, we aim to provide a robust solution for efficient energy management and enhanced smart grid integration. In summary, the contributions of this work are as follows: (i) Establishment of a comprehensive benchmark for the Plegma dataset and NILM within the Mediterranean context; (ii) Introduction of an optimized structured pruning methodology that exploits unstructured one to identify the optimal sparsity ratio per layer for removing whole units; (iii) Development and evaluation of an edge-based NILM approach utilizing both unstructured and optimized structured magnitude pruning, providing a comparative analysis of their respective strengths and limitations to assess the practical applicability of decentralized NILM approaches in real-world scenarios.

## II. OPTIMIZING STRUCTURED PRUNING FOR NILM

### A. NILM Problem Formulation

Non-intrusive load monitoring (NILM), initially introduced in [18], involves estimating the appliance level consumption by exclusively relying on the aggregate active power. Formally, for a number of  $m = 1, \dots, M$  appliances and a fixed time window  $t = 1, \dots, T$ , NILM is defined as

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

where  $y_m(t)$  denotes the power consumption of the  $m$ -th appliance,  $\epsilon(t)$  denotes the noise signal captured from measurement instruments and from appliances not individually metered during the data collection process [19], and  $x(t)$  is the aggregated signal. The aim of energy disaggregation is to estimate the power usage of each appliance  $y_m(t)$  at any given time  $t$  by relying solely on the aggregated power data  $x(t)$ . Solving NILM problems with traditional algorithms is often difficult, leading researchers to adopt deep neural networks (DNNs) for this task. While DNNs are effective universal function approximators for NILM, their complexity requires numerous neurons and substantial computational power, highlighting the need for compression.

### B. Pruning Strategies

DL models used to approximate eq. (1) can be complex, imposing challenges to the computational resources required for their real-time deployment on edge. To tackle such issues,

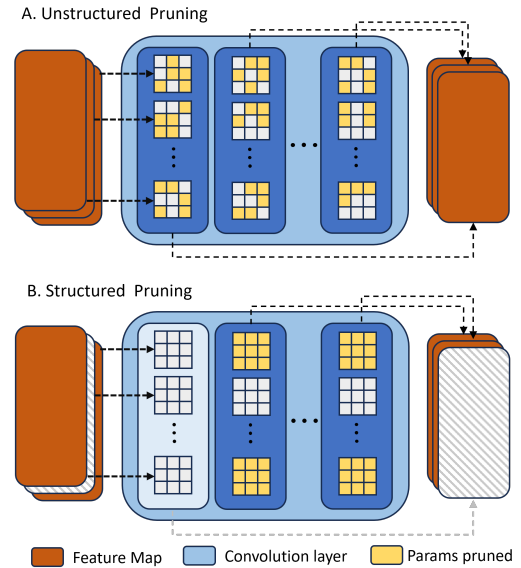


Fig. 1. Difference between unstructured (top) and structured (bottom) pruning. Unstructured pruning removes individual weights, whereas structured pruning removes whole units (filters and channels).

deep neural network (DNN) compression methods are used, with pruning being a popular approach. Pruning entails removing individual weights or entire units, such as filters or channels, based on a criterion like the  $L_1$  norm. The removal of individual weights is referred to as *unstructured pruning*, whereas the removal of entire units is known as *structured pruning*. Each process is described as follows:

**Unstructured pruning:** For a DNN model with weights represented in an ordered set  $\mathbf{W} \in \mathbb{R}^N$ , and a sparsity ratio  $s$  denoting the proportion of pruned parameters over the total, unstructured pruning sets individual weight values  $w_{i,j}$ , with connection from  $i$  to  $j$ , to 0 if  $\|w_{i,j}\|_1 < s$ , where  $\|\cdot\|_1$  is the  $L_1$  norm. As a result, the pruned weights  $\mathbf{W}' \in \mathbb{R}^N$  lead to a reduced model trainable parameter size. However, since the weights are replaced with zeros rather than removed, to fully realize compression benefits, we need hardware specifically designed for sparse matrix operations, which increases the cost and complexity of the real-world deployment of such a solution [20]. The number of pruned weights is determined by the sparsity ratio and directly impacts the classification performance of the model; the higher the sparsity ratio, the larger the computational performance drop.

**Structured pruning:** In contrast to the above method, structure pruning defines the sparsity ratio  $s$  as the proportion of pruned units like filters, neurons, or channels over the total ones. Specifically, for a DNN with  $C \in \mathbb{N}^k$  channels, the structured pruning process sets  $C$  channels to 0 if  $\|C\|_1 < s$ . The resulting number of channels,  $C' \in \mathbb{N}^{k'}$  not only results in reduced model parameter size but also in reduced computational resources as floating point operations are removed entirely, constituting it a more hardware-efficient and straightforward deployment approach.

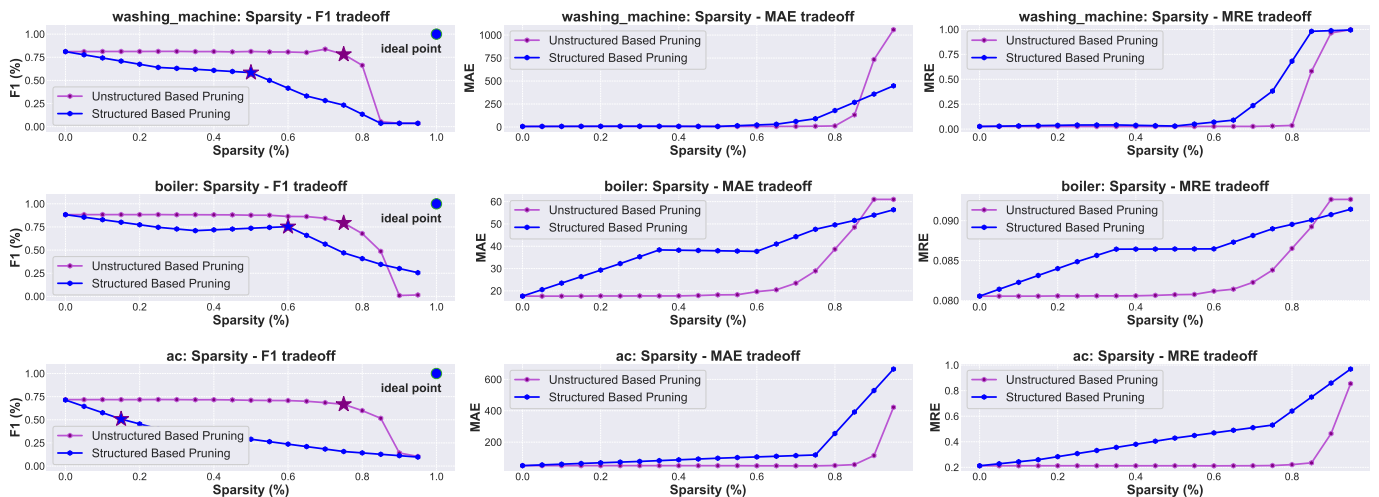


Fig. 2. Comparison of computational complexity (sparsity %) for different pruning thresholds (0-95%) to performance degradation (F1, MAE, MRE). Stars in the Sparsity-F1 plots mark the optimal pruning ratio  $\hat{s}$  for both optimized structured and unstructured magnitude-based pruning, while the blue dot represents the 'ideal' point.

### C. Proposed Optimized Structured Pruning

Both methods discussed in the previous subsections have their own advantages and disadvantages when real-time edge deployment is considered. Specifically, unstructured pruning makes the weight matrix a sparse one by zeroing individual weights but keeping the floating point operations number the same. On the contrary, structured pruning results in a computationally efficient structure by removing entire units, which can result in significant performance degradation if not applied judiciously.

This work combines the strengths of both approaches by determining the optimal sparsity ratio distribution across different layers in structured pruning of DL models derived from unstructured approach. The motivation for this work stems from the arbitrary selection of the sparsity ratio of each layer when structured pruning is applied, which often leads to the removal of filters or channels containing crucial features, consequently causing unwanted performance degradation. To tackle this issue, we leverage unstructured pruning to identify the percentage of parameters that should be pruned in each layer based on the desired sparsity ratio. Then, based on this percentage value, we apply structured pruning in each layer to remove an equal number of units, i.e., filters or channels. To give a better intuition behind this, consider the following example. Assume a simple 1D CNN model containing 3 layers, 2 convolutional ones, and 1 linear, along with a desired sparsity ratio of 50%. Using unstructured pruning, an optimal distribution of weights that should be removed can be 15%, 15%, 30% for the 2 convolutional and the linear layers, respectively. Now that the optimal sparsity ratio per layer is *known*, structured pruning is employed to remove whole units in each layer equal to the percentages identified from the unstructured pruning, i.e., 15%, 15%, 30% for the 2 convolutional and the linear layers respectively. As a result, structured pruning is guided in a sense by the unstructured one

to keep filters and channels with the most important features instead of removing them arbitrarily.

## III. EXPERIMENTAL RESULTS

### A. Experimental setup: Dataset & Model architecture

The experiments for both the baseline, which is the unpruned trained model and the pruned models were conducted using the Plegma dataset. [13], which provides total and appliance-level electricity consumption data at 10-second intervals. This study focuses on devices commonly used in the Mediterranean region, such as boilers and air conditioners, which are not typically included in other NILM datasets, and thus, their results are presented. The models were trained on data from houses 1-13, excluding house 2, which was used as unseen test data.

The chosen model was a sequence-to-sequence 1D Convolutional Neural Network known for its capabilities in edge-NILM [21]. This model offers better computational efficiency than the traditional seq2point CNN architecture, as it generates predictions for entire windows rather than individual time points, thus requiring fewer forward passes.

### B. Evaluation metrics

To evaluate the model's performance, the following three metrics were adopted: Mean Absolute Error (MAE), Mean Relational Error (MRE), and the F1 score. MAE and MRE were used to assess the regression performance of the model and its ability to predict the consumption signature of each of the tested appliances, while the F1 score was used to evaluate the model's performance in correctly classifying their 'on/off' states. Additionally, the computational complexity of the model was evaluated using the number of trainable parameters and Floating Point Operations (FLOPs). They measure the number of arithmetic operations required to execute the model, providing insight into its computational efficiency.

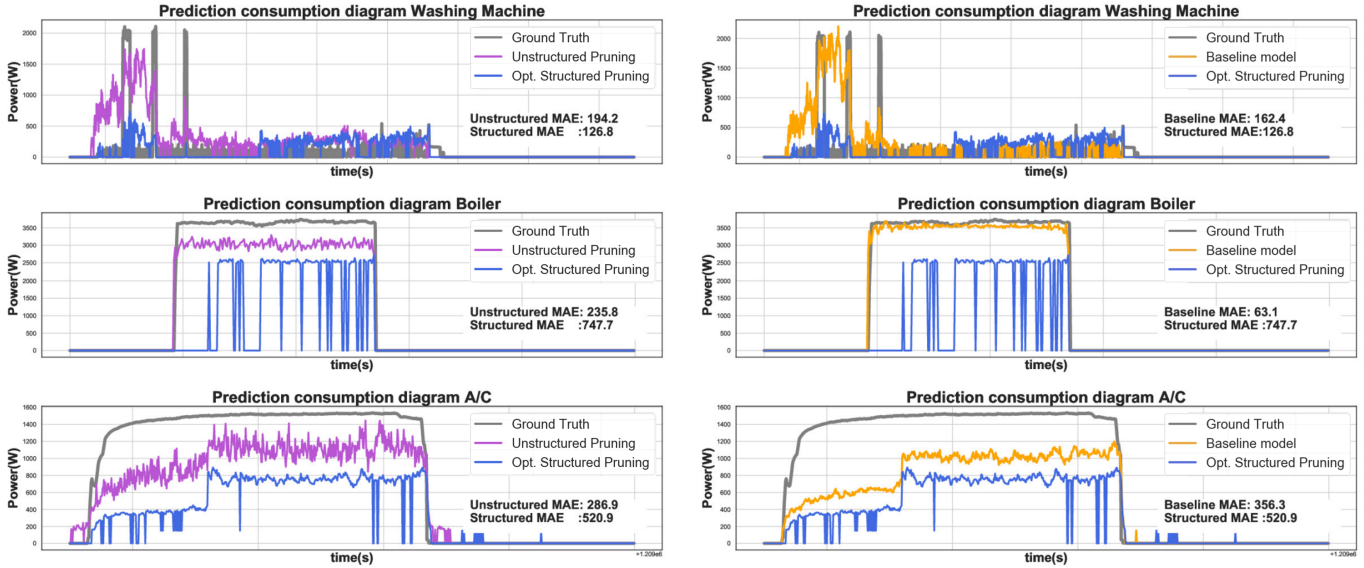


Fig. 3. Prediction consumption using the optimized structured pruning vs. the unstructured pruning scheme and the optimized structured pruning vs. the baseline model. The pruning thresholds were set equal to the  $\hat{s}$  as determined by Eq. (3).

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

Appliance	Approach	Compression metrics			Performance metrics		
		Pruning Percentage (%) $s = \hat{s}$	Number of Trainable Parameters	MFLOPs	F1	MAE	MRE
Washer	Baseline	0	22147640	430.26	0.81	5.79	0.026
	Unstructured Pruning	75	5538140	430.26	0.78	8.42	0.031
	Opt. Structured Pruning	50	11085760	259.01	0.58	6.05	0.032
Boiler	Baseline	0	22147640	430.26	0.88	17.64	0.081
	Unstructured Pruning	75	5538140	430.26	0.79	28.92	0.082
	Opt. Structured Pruning	60	8868520	220.12	0.75	37.71	0.086
A/C	Baseline	0	22147640	430.26	0.71	50.19	0.212
	Unstructured Pruning	75	5538140	430.26	0.66	48.79	0.228
	Opt. Structured Pruning	15	18827590	379.66	0.51	64.08	0.259

To identify the optimal pruning sparsity ratio  $\hat{s}$  for each model, we consider a two-step procedure [21]. First, we calculate the Euclidean distance for each one of the selected sparsity ratio percentages with values  $s = 0\%, 5\%, \dots, 90\%, 95\%$  with respect to the 'ideal point' which is that of  $s = 100\%$  and  $F1 = 100\%$  (Eq. 2). Then, we select  $\hat{s}$  to be equal to the  $s$ , resulting in the minimum Euclidean distance as shown in Eq. 3. Note that the above metric used accounts for both performance loss in pruned models and savings in parameter reduction.

$$\text{dist}(F1, s) = \sqrt{(100 - F1)^2 + (100 - s)^2} \quad (2)$$

$$\hat{s} = \arg \min_{s \in (0, 0.95)} (\text{dist}(F1, s)) \quad (3)$$

### C. Evaluation and Insights

A comparative analysis of unstructured and optimized structured magnitude-based pruning methods is shown in Fig. 2. Specifically, it evaluates both pruning approaches using the chosen performance metrics across various compression levels

ranging from 0-95% sparsity. The 'sparsity vs performance' curves suggest that unstructured pruning achieves higher performance compared to optimized structured pruning when large sparsity levels are selected for all tested appliances. For the boiler, which exhibits a simple consumption signal and low variation across different boiler types, the performance gap between the two pruning methods is smaller. Both methods effectively identify sparser networks, achieving trainable parameter reductions of 75% and 60%, respectively. In contrast, for the air conditioner (A/C), which has a highly variable consumption signal due to the diversity in device types, wattages, and operational modes, the performance gap is more pronounced. Specifically, optimized structured pruning is only able to reduce model parameters by 15%, highlighting its limitations compared to unstructured pruning, which achieves a reduction of 75% in handling devices with complex consumption patterns.

Similar insights are derived from the analysis of the consumption prediction diagrams presented in Figure 3. The results indicate that unstructured pruning provides better disag-

gregation performance compared to optimized structured pruning across all tested appliances. Nevertheless, both pruning methods effectively identify device activations and accurately classify their on-off states.

An analysis of the compression metrics reveals significant differences between the pruning methods. Although unstructured pruning achieves a better trade-off between pruning percentage and performance, this advantage does not translate into improved MFLOPs, showing no improvement (0%) for all tested appliances. This is due to the inherent execution method of unstructured pruning, where parameters are replaced with zeroes, maintaining the same model dimensions. In contrast, the optimized structured pruning, although not delivering as robust performance, significantly reduces the MFLOPs by removing entire units and filters from the model. This method directly enhances the efficiency of the model's inference phase, resulting in reductions of 39.8%, 48.85%, and 11.75% in MFLOPs for the washing machine, boiler, and air conditioner, respectively. These findings are particularly relevant for the real-world deployment of edge NILM (Non-Intrusive Load Monitoring) solutions. Optimized structured pruning's ability to reduce the computational load and improve inference efficiency makes it a more practical and deployable option for edge devices, where resource constraints are a critical factor. By lowering the MFLOPs, optimized structured pruning enables more efficient and effective NILM solutions on edge devices, facilitating wider adoption in real-world applications.

#### IV. CONCLUSIONS

This work addresses the critical challenges posed by the resource-intensive nature of deep learning models in NILM, as well as their context-aware characteristics. Unlike existing studies, this research benchmarks devices such as air conditioners and boilers from the Plegma dataset, which are prevalent in the Mediterranean region. It compares the performance between the baseline CNN model, unstructured magnitude-based pruning, and optimized structured pruning. Experimental results demonstrate that all tested methods provided satisfactory results for the selected appliances, offering valuable insights into the applicability of NILM in this geographical area. Furthermore, the proposed structured pruning technique, in contrast to the unstructured approach, presents a viable solution for edge-computed NILM by significantly reducing the MFLOPs of the model by up to 48.85%. Unlike unstructured pruning, which does not achieve a reduction in MFLOPs, structured pruning decreases the computational load, thereby enhancing the practicality of NILM systems for real-world deployment on resource-constrained edge devices. These findings underscore the potential of edge NILM towards flexibility and energy transition in the Mediterranean region, facilitating the broader adoption and implementation of NILM solutions in real-world scenarios.

#### REFERENCES

- [1] E. Commission, D.-G. for Energy, C. Alaton, and F. Tounquet, *Benchmarking smart metering deployment in the EU-28 – Final report*. Publications Office, 2020.
- [2] S. Athanasoulas, 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 PETRA International Conference*, ser. PETRA '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 589–595. [Online]. Available: <https://doi.org/10.1145/3594806.3596541>
- [3] 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.
- [4] S. Athanasoulas, 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. New York, NY, USA: Association for Computing Machinery, 2022, p. 601–607.
- [5] S. Sykiotis, M. Kaselimi, A. Doulamis, and N. Doulamis, "Electricity: An efficient transformer for nilm," *Sensors*, vol. 22, no. 8, 2022.
- [6] K. Srinivasarengan, Y. Goutam, M. G. Chandra, and S. Kadhe, "A framework for non intrusive load monitoring using bayesian inference," in *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. IEEE, 2013, pp. 427–432.
- [7] Z. Wu, C. Wang, W. Peng, W. Liu, and H. Zhang, "Nilm using factorial hidden markov model based on adaptive density peak clustering," *Energy and Buildings*, vol. 244, p. 111025, 2021.
- [8] D. Batic, G. Tanoni, L. Stankovic, V. Stankovic, and E. Principi, "Improving knowledge distillation for non-intrusive load monitoring through explainability guided learning," in *ICASSP 2023*, 2023, pp. 1–5.
- [9] R. Kukunuri, A. Aglawe, J. Chauhan, K. Bhagtani, R. Patil, S. Wallia, and N. Batra, "EdgeNILM: Towards NILM on Edge Devices," in *Proceedings of the 7th ACM Energy-Efficient Buildings, Cities, and Transportation*, ser. BuildSys '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 90–99.
- [10] S. Ahmed and M. Bons, *Edge Computed NILM: A Phone-Based Implementation Using MobileNet Compressed by Tensorflow Lite*. New York, NY, USA: ACM, 2020, p. 44–48.
- [11] J. Barber, H. Cuayáhuil, M. Zhong, and W. Luan, *Lightweight NILM Employing Pruned Sequence-to-Point Learning*. New York, NY, USA: Association for Computing Machinery, 2020, vol. 1, p. 11–15.
- [12] S. Athanasoulas, S. Sykiotis, N. Temenos, A. Doulamis, and N. Doulamis, "A pre-training pruning strategy for enabling lightweight non-intrusive load monitoring on edge devices," in *2024 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, 2024, pp. 249–253.
- [13] S. Athanasoulas, F. Guasselli, N. Doulamis, A. Doulamis, N. Ipiotis, A. Katsari, L. Stankovic, and V. Stankovic, "The plegma dataset: Domestic appliance-level and aggregate electricity demand with metadata from greece," *Scientific Data*, vol. 11, no. 1, p. 376, April 2024. [Online]. Available: <https://doi.org/10.1038/s41597-024-03208-0>
- [14] 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, 03 2015.
- [15] J. Kolter and M. Johnson, "REDD," in *IN SUSTKDD*, vol. 25, 01 2011.
- [16] C. Beckel, W. Kleiminger, R. Cicchetti, T. Staake, and S. Santini, "The eco data set and the performance of non-intrusive load monitoring algorithms," in *Proceedings of the 1st ACM conference on embedded systems for energy-efficient buildings*, 2014, pp. 80–89.
- [17] 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, no. 1, pp. 1–12, 2017.
- [18] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.
- [19] 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.
- [20] Y. Sun, L. Zheng, Q. Wang, X. Ye, Y. Huang, P. Yao, X. Liao, and H. Jin, "Accelerating sparse deep neural network inference using gpu tensor cores," in *2022 IEEE High Performance Extreme Computing Conference (HPEC)*, 2022, pp. 1–7.
- [21] S. Athanasoulas, S. Sykiotis, M. Kaselimi, A. Doulamis, N. Doulamis, and N. Ipiotis, "Opt-nilm: An iterative prior-to-full-training pruning approach for cost-effective user side energy disaggregation," *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 4435–4446, 2024.