Energy Reports 14 (2025) 3048–3061

Contents lists available at ScienceDirect

Energy Reports

journal homepage: www.elsevier.com/locate/egyr

Research paper

Isomorphic structured pruning of temporal CNNs for scalable NILM on edge devices

Sotirios Athanasoulias [a],*, Nikos Temenos [a], Ilias Kappos [b], Isidoros Kokos [b], Pedro Antonio Garcia-Abadillo Navaro [c], Nikolaos Ipiotis [d], Anastasios Doulamis [a], Nikolaos Doulamis [a]

[a] National Technical University of Athens, School of Rural, Surveying and Geoinformatics Engineering, Athens, 15773, Attica, Greece
[b] Intracom S.A. Telecom Solutions, Telco Software Dpt, Athens, 19002, Attica, Greece
[c] Barbara Tech, Technology Software Dpt, Madrid, 28040, Community of Madrid, Spain
[d] Plegma Labs, Technology Software Dpt, Athens, 28040, Attica, Greece

ARTICLE INFO

Keywords:
Isomorphic structured pruning
NILM
Deep learning
Temporal CNN
Energy disaggregation
Edge deployment

ABSTRACT

This work introduces isomorphic structured pruning (ISP) within the Non-Intrusive Load Monitoring (NILM) domain to support the deployment of optimized Deep Learning (DL) models in resource-constrained edge environments. By grouping functionally equivalent substructures and ranking them independently, it increases the tolerance to neural network units removal thereby offering improved pruning-computational performance trade-off, along with consistency and reliability in the pruning results. Experimental results using a temporal Convolutional Neural Network (CNN) architecture on two datasets, the Plegma, containing Mediterranean-based appliances, and UK Dale, containing widely-used domestic appliances, demonstrate up to 97% model size reduction and computational efficiency gains of up to 42× in terms of Multiply-and-Accumulate (MAC) operations, all with negligible classification performance losses. Comparisons with popular pruning techniques highlight ISP's ability in maximizing the trade-off between pruning threshold values and classification performance with the results showing 85% pruning of the temporal CNN, resulting in a value of approximately 0.5 MB in size, combined with up to 99% in accuracy and up 81.4% in F1-score across all devices considered. To showcase its real-world applicability, ISP was deployed on two industrial edge devices with diverse computational resources, an EI-52 and an LEC-7230M, achieving over 95% faster inference times, along with reductions exceeding 85% in both CPU usage and energy consumption compared to the baseline model. These findings underscore the advantages of the ISP as a practical and scalable solution for deploying high-performance NILM models in real-world edge computing scenarios.

1. Introduction

Improving energy efficiency and optimizing energy consumption in buildings has become an urgent priority in response to the global energy and climate crisis (Hosseini et al., 2023). Advanced metering infrastructure plays a pivotal role in this transition by enabling direct communication between both utilities and consumers, thereby supporting real-time energy management and grid optimization efforts (Hosseini et al., 2023). With smart meter deployments expected to reach over 80% of European households by 2025, the potential impact of detailed energy monitoring on both user behavior and system-level efficiency continues to grow significantly (Commission et al., 2020). A particularly effective strategy within this domain is electricity load monitoring at the appliance level, which allows for more granular insights into household energy usage (Kee et al., 2019). In this context, Non-Intrusive Load Monitoring (NILM), emerges as a critical enabler for smart energy management, offering actionable feedback without the need for extensive sub-metering infrastructure.

NILM is a technique that is used to estimate the energy usage of individual appliances by analyzing only the total consumption signal from a single central meter (Hart, 1992). By enabling detailed consumption insights without additional equipment, NILM provides a scalable, low-cost path to personalized energy feedback and behavior change, all critical components in addressing residential energy demand and enabling smarter, more efficient grids. In recent years, rapid advancements in Artificial Intelligence (AI) have advanced NILM research toward Deep Learning (DL) approaches (Kaselimi et al., 2022), where

* Corresponding author.
E-mail address: sotiriosathanasoulias@gmail.com (S. Athanasoulias).

https://doi.org/10.1016/j.egyr.2025.09.007
Received 23 May 2025; Received in revised form 7 August 2025; Accepted 18 September 2025
Available online 8 October 2025

these models have demonstrated significantly improved disaggregation performance compared to traditional signal processing techniques. However, the high computational complexity of deep neural networks (DNNs) requires centralized data processing infrastructures, which introduces challenges related to cost, scalability, and privacy, particularly when sensitive household data must be transferred to external servers (Athanasoulias et al., 2024b). To overcome these barriers and enable decentralized, edge-based NILM solutions, research has shifted towards reducing the computational footprint of DL models, achieved through various model compression techniques, aiming to make the deployment of DNN-based NILM systems feasible on resource-constrained devices.

An approach that has been extensively applied in NILM to reduce a model's computational demands is pruning; a technique that facilitates the creation of smaller, resource efficient yet computationally effective DNNs by removing less significant components from a pretrained model (Kukunuri et al., 2020). This can involve eliminating individual weights through unstructured pruning, or larger architectural elements such as neurons, filters, channels, or even entire layers through structured pruning. However, existing pruning methods in the NILM domain exhibit important limitations (Athanasoulias et al., 2024d). Specifically, in unstructured pruning, model deployment often depends on specialized AI accelerators or dedicated software to get the full benefits from weight elimination as their values are zeroed rather than removed, thus hindering its widespread applicability (Sun et al., 2022). On the other hand, although conventional structured pruning provides a more straightforward deployment, it often faces difficulties in evaluating the relative importance of diverse computing units such as self-attention layers, depth-wise convolutions, and residual connections as they differ not only in parameter scales but also in computational roles, making it difficult to apply a uniform pruning strategy without risking the removal of crucial parts of the network (Fang et al., 2024; Li et al., 2024).

Motivated by the applicability and model-agnosticity limitations of the above pruning techniques, this work introduces isomorphic structured pruning (ISP) to DNN-based NILM models to significantly improve their computational efficiency. ISP categorizes each one of the DNN model's sub-structures based on their functional types through network graph modeling, groups them according to the existence of isomorphic connections between them and finally prunes units across the network's architecture through an importance ranking function. By doing so, ISP allows for an improved trade-off between computational performance and model complexity compared to unstructured and structured pruning techniques used within the context of NILM. In summary the contributions of this work are presented below:

- **Isomorphic Structured Pruning for NILM:** ISP is introduced within the context of NILM, a technique that isolates and ranks network sub-structures based on their functional types enabling more precise and effective pruning without compromising disaggregation performance.
- **Evaluation on Two Distinct Real-World Datasets:** ISP is applied to a temporal, sequence-to-sequence (seq2seq) CNN architecture, and evaluated on two datasets: (1) the Plegma dataset, which captures region-specific appliance usage behaviors and contextual characteristics of the Mediterranean region, and (2) the UK Dale dataset, which has been widely used in NILM research.
- **Real-World Edge Deployment and Validation:** To assess real-world applicability, the DL models pruned through ISP are deployed and tested on two different edge devices with varying processing capabilities. This evaluation examines the impact of model compression on latency, resource consumption, and performance in practical scenarios, confirming the feasibility of deploying high-performing NILM models on low-resource, privacy-preserving edge platforms.

The remainder of the paper is organized as follows. Section 2 reviews NILM with DL and edge-computed approaches, focusing on pruning methods. Section 3 presents the operation principle of ISP along with the NILM problem formulation. Section 4 describes the experimental setup, covering datasets, training, evaluation metrics, while Section 5 presents computational performance results along with discussion on them. Section 6 advances the practicality of the ISP technique for NILM by applying it in two edge devices and demonstrates the results. Finally, Section 7 concludes this work.

2. Related work and contribution

In this section, we provide a brief overview of DL approaches for energy disaggregation, followed by an overview compression techniques frequently used, with a particular focus on unstructured pruning and conventional structured pruning, serving as the baseline methods for this work.

2.1. Deep learning models for NILM

With the advancement of AI, recent NILM approaches have transitioned from traditional signal processing and probabilistic approaches to DL models. Recurrent neural network (RNN) approaches, such Gated Recurrent Unit (GRU) Long Short-Term Memory (LSTM), leverage feedback connections to capture temporal dependencies in power signals, enabling fast convergence and improved disaggregation performance (Hwang and Kang, 2022; Wu et al., 2024; Xuan et al., 2024). However their performance can deteriorate when longer noisy sequences are provided due to the vanishing gradient problem and they also provide increased model complexity because of their complex gating mechanisms (Ribeiro et al., 2020). Denoising Autoencoders (dAEs) (García et al., 2024; Bao et al., 2018) and Generative Adversarial Networks (GANs) (Kaselimi et al., 2020) represent two additional effective methodologies in NILM, enhancing robustness to noise and enabling the modeling of complex appliance power usage distributions for more accurate disaggregation across varying conditions and appliance types. Training such models, though, can be computationally intensive, often requiring considerable time and a large amount of labeled data to achieve optimal performance (Huber et al., 2021). Finally, Transformer-based approaches have also gained attention in NILM due to their ability to capture long-range dependencies in energy time series data through attention mechanisms, leading to improved disaggregation performance (Yue et al., 2020a; Angelis et al., 2023; Rong et al., 2025). Despite these advantages, Transformer architectures face significant computational challenges which hinder their applicability in real world scenarios (Sykiotis et al., 2023). Specifically, the self-attention mechanism has a quadratic complexity with respect to input sequence length, resulting in high demands for both memory and processing power, especially when dealing with long sequences (Vaswani et al., 2017).

Another DL architecture widely adopted in NILM that effectively addresses many of the aforementioned challenges is the Temporal Convolutional Neural Network (1D CNN) (Zhang et al., 2018). Temporal CNN architectures enable high disaggregation performance primarily due to their capability to capture both local and long-range temporal dependencies in sequential energy consumption data, without encountering the vanishing gradient problem that is observed in RNN architectures. Additionally, although CNNs share the general drawback of increased computational complexity with the other DNN-based NILM approaches presented above, they have emerged as the main architectural choice for edge-based NILM research, being used in the majority of recent studies (Kukunuri et al., 2020; Ahmed and Bons, 2020; Athanasoulias et al., 2024c; Lu et al., 2022; Pan et al., 2024; Barber et al., 2020). This widespread adoption is primarily attributed to the architectural simplicity and modularity of convolutional and linear layers, which enhances compatibility with a wide range of model compression techniques. This compatibility enables significant simplification of CNN models, reducing computational overhead and facilitating their efficient deployment in resource-constrained environments.

2.2. NILM pruning techniques

To address the computational demands of DNN-based NILM models, pruning has been widely adopted. The two main approaches used in NILM include the unstructured and structured pruning which will serve as baselines in this work.

Unstructured Pruning: It is a widely adopted technique in edge computing-based NILM, primarily due to its simplicity and ease of implementation. Unstructured pruning is based on the selective deactivation of individual weights in a NN based on their magnitude; by selecting a desired pruning threshold value, it removes the percentage of weights with the smallest absolute values based on L_1 norms. Although this method reduces the number of active parameters during inference, the weights are not physically removed from memory, and thus benefiting from computational advantages requires specialized hardware for sparse matrix operations, which increases deployment complexity and cost (Sun et al., 2022). Within the context of NILM, several works have adopted this technique. Barber et al. (2020) evaluated different weight unstructured pruning techniques on REFIT (Murray et al., 2017), Athanasoulias et al. (2024c,b) proposed a lottery ticket-inspired unstructured pruning methodology applied before full training to cut training computational costs, and Sykiotis et al. (2023) combined post-training unstructured pruning with quantization. Yet, they all share the core limitation of unstructured pruning, hindering real-world deployment.

Structured Pruning: In contrast to the unstructured pruning, structured pruning operates at a higher operational level in the sense that it removes entire units from the model such as neurons, filters and channels instead of zeroing the weights values (He and Xiao, 2023). As such, with a selected pruning threshold, defined here as the fraction of units pruned relative to the total, the resulting pruned model is not only lowered in parameter count but also lighter in terms of computational demands as entire floating-point operations are eliminated, making it more hardware-efficient and easier to deploy. Yet, considering its uniformity in unit removal, structured pruning is inherently more prone to performance degradation; pruning percentage should be selected with caution, as arbitrarily setting its value can potentially lead to aggressively elimination of important units, drastically affecting model's performance. Structured pruning has been adopted in recent edge-computed NILM studies, including Kukunuri et al. (2020), Wang et al. (2021), and Athanasoulias et al. (2024d). Although these works establish a valuable baseline for pruning-based model compression in NILM, they often suffer from greatly reduced performance at higher pruning ratios and offer limited evaluation on real-world edge devices.

2.3. Isomorphic structured pruning applications

ISP is a recently proposed pruning technique that extends traditional structured pruning by accounting for the architectural sub-structure of neural networks. Unlike standard structured pruning, which removes entire filters or layers globally, ISP delves deeper into the sub-structures that are formed within a network architecture; it models the networks' sub-structures as graphs to identify their inter-dependencies, groups them isomorphically based on their topology and parameters, and finally prunes units according to importance ranking functions. This allows ISP to significantly improve the trade-off between computational performance and model complexity. On top of these, ISP is inherently model-agnostic, attributed to its operation principle being based upon the networks sub-structures grouping, making it attractive for a wide variety of applications. One such is the image classification in the computer vision domain (Fang et al., 2024), where it is shown that DL models with diverse computing units (e.g., CNN-based, Vision Transformers) can effectively be sized down in terms of parameters with minimal computational performance loss with the use of ISP. Beyond the well-promising results, ISP development remains (currently) in a theoretical level in the sense that direct application of it is not yet explored.

2.4. ISP adaptation to the NILM application domain

Despite the use of unstructured and structured pruning techniques in NILM, their practical deployment on resource-constrained edge devices remains challenging. Unstructured pruning, while effective at reducing parameter count, relies on specialized hardware to process sparse representations, often unavailable in real-world embedded systems. Structured pruning, though more compatible with standard hardware, is highly sensitive to the selection of pruning thresholds, which can lead to substantial performance degradation if not carefully tuned. These limitations pose barriers to building efficient, generalizable NILM models that can run reliably on low-power edge platforms.

To address these challenges, this work introduces ISP into the NILM domain. It brings ISP from theory to practice, combining time-series energy data, CNN-based seq2seq architectures and real-world edge deployment with tailored use cases in the NILM domain. This integration demonstrates the practical viability of ISP for NILM, validating its use as a compression-aware pruning method suitable for edge-based energy disaggregation. These contributions along with differentiation of this work compared are summarized in Table 1.

3. Isomorphic pruning for NILM

This section introduces first to the NILM problem formulation with DL and then proceeds with explaining the ISP for NILM.

3.1. NILM problem formulation with DL

The task of NILM estimates the power consumption levels of an individual appliance by relying exclusively on the aggregated power consumption of all appliances. In formal terms, for each appliance $m = 1, \dots, M$ with corresponding power consumption levels $P_m(t)$ measured at a fixed time-window $t = 1, \dots, T$, the NILM problem is defined as

$$P_A(t) = \sum_{m=1}^{M} P_m(t) + \epsilon(t), \tag{1}$$

with $P_A(t)$ being the aggregated power consumption levels, while $\epsilon(t)$ is a signal denoting noise stemming from (a) instrumentation measurements, and (b) appliances that do not contribute to the data collection process but their presence is evident (Huber et al., 2021). Given $P_A(t)$, NILM tries to estimate the individual appliance power consumption levels instead of directly calculating it, i.e. $\hat{P}_m(t) \approx P_m(t)$, considering that NILM is inherently ill-posed problem (Schirmer and Mporas, 2022).

The process described above, also known as disaggregation process, is typically solved using diverse techniques including digital signal processing, optimization, ML and DL (Cruz-Rangel et al., 2024), with the latter gaining significant attention due to its effective function approximation capabilities (LeCun et al., 2015). Specifically, within the context of NILM, a DL model $F : \mathcal{X} \rightarrow \mathcal{Y}$ is used to map input sequences of aggregate power consumption $P_A(t) \in \mathcal{X}$ to estimated appliance-level consumption $\hat{P}_m(t) \in \mathcal{Y}$, parameterized by a set of learnable weights $\mathcal{W} = \{W_1, W_2, \dots W_L\}$ where L is the total number of layers, formally defined as $\hat{P}_m(t) = F(P_A(t); \mathcal{W})$. The seq2seq CNN architecture (Athanasoulias et al., 2024b) used in this work describing the DL function approximation process is shown in Fig. 1. It is composed of 7 computation layers where the first 5 are sequentially connected 1D Convolution layers followed by a ReLU activation function, while the last 2 are dense ones, with the final one having Sigmoid as its activation function. The use of the Sigmoid ensures that the output remains in the [0, 1] range, aligning with the normalized target values and improving prediction stability.

Table 1
Overview of existing and proposed applications of Isomorphic Structured Pruning (ISP).

Source	Domain	Architecture	Dataset	Use Case	Deployment
Zhang et al. (2023)	Computer Vision	CNNs, ViTs	ImageNet-1K	Image Classification	No
This Work	Smart Energy/NILM	Seq2Seq CNNs	Plegma, UK Dale	Time Series, NILM	**EI-52, LEC-7230M**

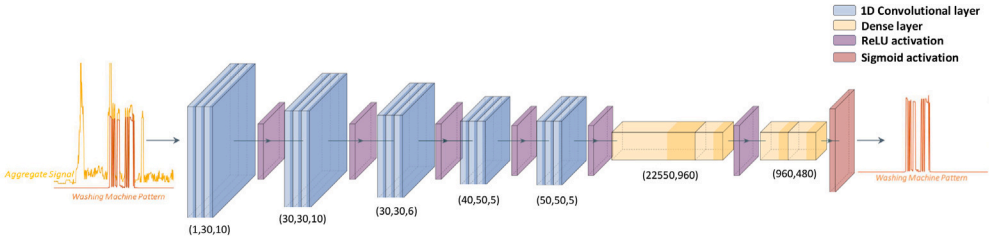

Fig. 1. The temporal seq2seq CNN architecture. It takes an aggregate consumption time series as input and predicts the corresponding appliance consumption for the same time window. The notation in the convolutional layers represents (in_channels, out_channels, kernel_size), while the notation in the dense layers represents (in_features, out_features).

3.2. Isomorphic structured pruning

The DL model of Fig. 1 used for NILM can be in fact prohibitively large for deployment in resource-constrained devices, attributed to the size of the learnable parameters $\mathcal{W}$. To address this issue, Isomorphic Structured Pruning (ISP) can be applied. Its process builds upon the following sequentially connected steps being (1) network sub-structures graph modeling and (2) isomorphic ranking and pruning. Below we delve deeper into each one of the ISP procedure phases and explain them in detail.

3.2.1. Network sub-structures graph modeling

To explain the graph modeling of network sub-structures, it is reasonable to proceed first with some definitions. Considering that the architecture of Fig. 1 contains both convolution and dense layers, they can be internally separated to $\mathcal{W} = \{\mathcal{W}^{\text{conv}}, \mathcal{W}^{\text{dense}}\}$ where

- if $W_l \in \mathcal{W}^{\text{conv}}$, then $W_l^{\text{conv}} \in \mathbb{R}^{C_l^{\text{conv}} \times N_l^{\text{conv}} \times K_l^{\text{conv}}}$ with $C_l^{\text{conv}}, N_l^{\text{conv}}, K_l^{\text{conv}}$ correspond to the number of input channels, output channels and kernel sizes,
- if $W_l \in \mathcal{W}^{\text{dense}}$, then $W_l^{\text{dense}} \in \mathbb{R}^{N_l^{\text{dense}} \times K_l^{\text{dense}}}$ with $N_l^{\text{dense}}, K_l^{\text{dense}}$ correspond to the input and output units of the dense layers.

Considering the above, a pruning function $q(W_l, d, s)$ can be applied to the input or output dimension d of any layer $W_l \in \mathcal{W}$. Specifically, if $W_l \in \mathcal{W}^{\text{conv}}$, the pruning function is

$$\hat{w_l} = q(W_l^{\text{conv}}, d, s) = \begin{cases} q(W_l^{\text{conv}}, 0, s) = W_l^{\text{conv}} e_s^T \in \mathbb{R}^{1 \times N_l^{\text{conv}}}, \\ q(W_l^{\text{conv}}, 1, s) = e_s W_l^{\text{conv}} \in \mathbb{R}^{C_l^{\text{conv}} \times 1}, \end{cases} \tag{2}$$

whereas if $W_l \in \mathcal{W}^{\text{dense}}$, the pruning function is

$$\hat{w_l} = q(W_l^{\text{dense}}, d, s) = \begin{cases} q(W_l^{\text{dense}}, 0, s) = W_l^{\text{dense}} e_s^T \in \mathbb{R}^{1 \times K_l^{\text{dense}}}, \\ q(W_l^{\text{dense}}, 1, s) = e_s W_l^{\text{dense}} \in \mathbb{R}^{N_l^{\text{dense}} \times 1}. \end{cases} \tag{3}$$

where $e_s = [0, \dots, 1, \dots, 0]$ is the standard basis vector having 1 in its sth element and dimension determined by i. the weights layer dimension, i.e. N_l, C_l (conv layer) or K_l, N_l (dense layer) and ii. the layer type, i.e. conv or dense; $d \in \{0, 1\}$ denotes if output ($d = 0$) or input ($d = 1$) is pruned; and $\hat{w_l}$ is the resulting pruned row or column vector. Extending the pruning process to the entire network architecture, necessitates taking into account *dependencies among layers* existing within it and hence their sub-structures. These can be modeled using a graph-based representation as follows. Let a graph $G = (V, E)$ consist of a set of vertices V, representing nodes and a set of edges $E \subseteq V \times V$, representing relationships between the vertices. Each pruning operation applied to a layer can be considered as a vertex $v \in V$ whereas the dependencies among the layers (e.g., pruning output channel in one layer affecting input channel of another layer) are modeled as edges $e \in E$. In this way, the graph is used to provide a principled coordination of pruning across layers while preserving the architecture's integrity.

To form the sub-structures and hence the groups, it is essential to find their intra-connections, namely connections existing among the layers within a single group. To achieve this, a single weights layer W_l is selected initializing the graph G with $V = \{(W_l, 0, s)\}$ and $E = \{\emptyset\}$. Then, for the elimination of the sth dimension, all other potential dependencies $(W_l', d', s') \notin V$ with $W_l' = \mathcal{W} \setminus \{W_l\}$ are iteratively investigated to ascertain if dependencies with (W_l, d, s) exist or not. This is achieved by simultaneously satisfying two criteria: (1) layer adjacency, where W_l and W_l' are adjacent layers such that $d = d'$ and $s = s'$ and (2) pruning function sharing, where W_l and W_l' and $q(W_l', d', s') = q(W_l, d, s)$. Once both criteria are met, vertices are updated as

$$V = V \cup \{(W_l', d', s')\} \tag{4}$$

while edges are updated as

$$E = E \cup \{((W_l, d, s), (W_l', d', s'))\}. \tag{5}$$

Note that the resulting graph G corresponds to the sub-structure of the initial selected weights layer W_l.

3.2.2. Isomorphic grouping and ranking-based pruning

The graph formation process described in the previous subsection corresponds to the formation of a single graph G and refers to an initially selected weights layer W_l. Undoubtedly, there exist layers W_l' without connections to the initially selected one W_l, but, connections among them exist. Therefore, it is reasonable to repeat the graph formation process to exhaust all W_l' until no other groups can be identified, so as to make the network architecture a set of intra-connected groups, defined as $\mathcal{G} = \{G_1, \dots, G_B\}$.

With all groups identified and created, isomorphic grouping can be applied, a process that involves sub-structure clustering based on their topology and parameter configurations. Assuming two groups $G_i, G_j \in \mathcal{G}$ with $i \neq j$, they are defined as isomorphic if the following hold simultaneously: (a) their total number of edges are equal and (b) if their vertices match in the sense that their corresponding layers originate from the same layer type. Formally this is defined as

$$H(G_i, G_j) = \mathbb{1}\{|E_i| = |E_j|\} \cap \mathbb{1}\left\{\left(T(W_{G_i}), d\right) = \left(T(W_{G_j}), d'\right)\right\}, \tag{6}$$

where E_i, E_j and W_{G_i}, W_{G_j} are the edges and the weights layers of groups G_i, G_j respectively, $T : \mathcal{W} \to \mathcal{T}$ is a function mapping layer weights to the DL architecture's possible layer types, $\mathcal{T} =$

Table 2
Appliance characteristics for the Plegma and UK Dale datasets. The cutoff is the maximum power (W) an appliance can reach. The on threshold is the minimum power (W) to be considered "on".

Dataset	Appliance	Cutoff (W)	On Threshold (W)	Min. On (s)	Min. Off (s)
Plegma	Boiler	4500	2000	20	150
	Washing Machine	3000	50	10	120
	A/C	4000	100	20	60
UK Dale	Kettle	3100	2000	10	20
	Dishwasher	2500	10	10	180
	Washing Machine	3000	50	10	120

$\{\text{Convolution}, \text{Linear}, \text{BatchNorm}, \text{etc.}\}$ and $\mathbb{1}\{\cdot\}$ is the indicator function. Isomorphic grouping is then achieved by applying Eq. (6) to each sub-structure G_i as

$$\mathcal{R} = \{G_i\} \cup \{G_j \mid H(G_i, G_j) = 1, \quad j \in \{1, \dots, B\}, \quad j \neq i\}. \tag{7}$$

Once the isomorphic groups are formed, an importance function $\hat{A}(\cdot)$ is applied to each one of them as $\{\hat{A}(G_1), \dots, \hat{A}(G_B)\}$ and is defined as

$$\hat{A}(G_i(V, E)) = \sum_{(W_{G_i}, d, s) \in V} A(W_{G_i}, d, s), \tag{8}$$

where $A(W_{G_i}, d, s)$ is a pruning function applied to the isomorphic group. It should be noted that the importance function is different than that of the pruning function, both serving two distinct roles; the former is used to rank parameters based on importance and can follow strategies like L_1 norm (Kumar et al., 2021), Taylor (Molchanov et al., 2019) or random (Li et al., 2022), whereas the latter is used to remove the units and can behave as a Magnitude (Lee et al., 2020) or Unstructured (Liao et al., 2023), among others. In this work we have used the L_1 norm as the importance function and Meta Pruner as the pruning function (Liu et al., 2019). Finally, sub-structure elimination is applied by removing $p_t\%$ (pruning threshold) in the isomorphic groups considering the importance scores from Eq. (8).

4. Experimental setup

This section provides a comprehensive analysis of the experimental framework, including a detailed description of the dataset used, the evaluation metrics, the model training configuration, and the comparative methods utilized to benchmark the proposed compression approach.

4.1. Dataset description

The experiments conducted in this study leverage both the Plegma (Athanasoulias et al., 2024a) and UK Dale (Kelly and Knottenbelt, 2015) datasets to comprehensively evaluate the proposed approach across diverse household energy consumption scenarios. Plegma dataset is a recently established public dataset that provides whole-house aggregate energy consumption and appliance-level measurements at 10-second intervals from 13 households over a one-year period. It is among the first publicly available datasets of its kind in the Mediterranean region, capturing consumption patterns characteristic of the local climate and lifestyle, offering valuable insights into Mediterranean energy usage, particularly for appliances that are commonly found in this region but are often underrepresented in other NILM datasets. Notably, it includes energy consumption patterns of air conditioners, used for both heating and cooling, and electric water boilers, constituting a significant share of total household energy consumption and hold substantial flexibility potential.

The dataset was collected at a 10-second sampling rate to match the specifications of real-world smart meters, such as SMETS2 HAN, ensuring the practical relevance of any solutions developed using this data. This work specifically focuses on the electric water boiler, washing machine, and air conditioner appliance, with their characteristics presented in Table 2. This information was used for data processing and for accurately identifying the on–off states of each appliance.

In addition to the Plegma dataset, the UK-DALE dataset was employed to broaden the evaluation scope with data from a different regional and climatic context. UK-DALE is a widely used open-access NILM benchmark containing aggregate (1 Hz) and appliance-level (1/6 Hz) power consumption recordings from five UK households. Its high temporal resolution and detailed appliance annotations have made it a reference point for NILM algorithm benchmarking. This work focuses on three appliances from UK-DALE: kettle, dishwasher, washing machine which were selected for their high energy use, frequent operation, and representative diversity in consumption patterns. These include simple binary loads like the kettle, multi-phase cycles of the dishwasher and washing machine.

4.2. Evaluation metrics

To assess the model's disaggregation performance, the following metrics were employed. The Mean Absolute Error (MAE) was used to evaluate the model's regression performance, measuring its ability to reconstruct the appliance signal by comparing the predicted and actual appliance consumption.

$$\text{MAE}(\hat{P}_m, P_m) = \frac{1}{T} \sum_{t=1}^{T} |\hat{P}_m(t) - P_m(t)|. \tag{9}$$

Accuracy and F1-score were used to assess the model's classification performance in detecting appliance states (on/off). An appliance was classified as "on" when the model's predicted consumption exceeded the activation threshold and remained within the predefined minimum on/off durations, as specified in Table 2. Accuracy quantifies the proportion of correctly identified appliance states and is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

where TP and TN denote correctly predicted "on" and "off" states, respectively, while FP and FN indicate misclassified cases. The F1-score offers a more representative evaluation of the NILM problem, as it accounts for both precision and recall, making it well-suited for handling class imbalance it is defined as:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{11}$$

where

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \tag{12}$$

As shown in Eq. (12), precision quantifies the proportion of correctly classified "on" states among all instances predicted as "on", while recall measures the proportion of correctly identified "on" states among all actual "on" instances. Since NILM datasets often exhibit significant class imbalance, with "off" states being far more prevalent than "on" states, the F1-score provides a more balanced assessment than accuracy by penalizing both false positives and false negatives, ensuring a fairer evaluation of model classification performance. Regarding the computational complexity of the model, we considered the number of model parameters, storage size of the model in megabytes, and Multiply-Accumulate Operations (MACs).

An important note about these computational metrics is that in structured pruning approaches, the pruning threshold, p_t, has a *hyperbolic relationship* with them. This is because structured pruning reduces both input and output dimensions by $p_t\%$, preserving the structural coherence of the model and ensuring compatibility between adjacent layers. As a result, the actual parameter pruning ratio becomes approximately $1-(1-p_t\%)^2$. To give a better intuition behind this, for an effective 50% pruning of the parameters, one could set p_t to 0.30, which yields an actual parameter reduction of $1-(1-0.3)^2=0.51$. Note that in the experimental results shown in Section 5, the initial set p_t value is considered.

To determine the optimal pruning threshold, denoted as p_{opt}, we implement a two-stage methodology. Initially, for each of the specified pruning thresholds, ranging from $p_t=0\%$ to $p_t=95\%$ in increments of 5%, we compute the Euclidean distance from the 'ideal' point, defined at $p_t=100\%$ with an F1-score of 100% as

$$\text{dist}(F1,p_t)=\sqrt{(100-F_1)^2+(100-p_t)^2} \tag{13}$$

Subsequently, we designate p_{opt} as the pruning threshold p_t that minimizes this Euclidean distance, as

$$p_{opt}=\arg\min_{p_t\in(0,0.95)}\left(\text{dist}\left(F1,p_t\right)\right) \tag{14}$$

This approach effectively balances the trade-off between maintaining model performance and reducing parameter count.

4.3. Model training setup

The model was trained using a learning rate of 1×10^{-4} with the Adam optimizer for 150 epochs. These hyperparameter choices follow prior NILM studies (Athanasoulias et al., 2024b; Zhang et al., 2018; Athanasoulias et al., 2024c; Sykiotis et al., 2022, 2023) using similar architectures, ensuring consistency and comparability. Training data was sourced from houses 1–13 (excluding house 2) for the Plegma dataset and from houses 1–5 (also excluding house 2) for the UK-DALE dataset. Both datasets were split into 80% training and 20% validation, while house 2 was entirely withheld from training and used exclusively for testing in both cases. This decision was made to prevent the introduction of bias by exposing the model to the unique consumption patterns of a specific household. Furthermore, this setup better simulates real-world deployment scenarios, where the model is applied to previously unseen houses. To improve model robustness, we employed 3-fold cross-validation and selected the best model based on the epoch with the highest average F1-score. The choice of F1-score as the model selection criterion is motivated by the inherent class imbalance in NILM, where the "on" state occurs significantly less frequently than the "off" state.

To account for both accurate predictions and device status classification, this work adopts a hybrid regression-classification loss function during training (Yue et al., 2020b). Specifically, apart from a Mean Squared Error (MSE) loss applied to $P_m(t)$, it incorporates 1. the probability distributions to evaluate the divergence between the predicted value and its corresponding label, 2. soft-margin loss to take into account the status predictions z with the purpose of introducing a penalty to inconsistent predictions and label status and 3. MAE of the $P_m(t)$ considering (9) to reduce the gap between predicted and ground truth energy values, tuned by a hyperparameter λ. The loss function is formally defined as

$$\begin{aligned}\mathcal{L}(P_m,z)=&\frac{1}{T}\sum_{t=1}^{T}(\hat{P}_m(t)-P_m(t))^2\\&+D_{KL}\left(\text{softmax}\left(\frac{\hat{P}_m(t)}{\tau}\right)\middle\|\text{softmax}\left(\frac{P_m(t)}{\tau}\right)\right)\\&+\frac{1}{T}\sum_{t=1}^{T}\log\left(1+\exp(-\hat{z}_t z_t)\right)+\frac{\lambda}{T}\sum_{t\in\mathcal{O}}|\hat{P}_m(t)-P_m(t)|,\end{aligned} \tag{15}$$

where $z,\hat{z}$ are the device status of the ground truth and the predicted values, τ is a hyperparameter used to tune the softmax function, $D_{KL}(\cdot)$ is the Kullback–Leibler divergence and $\mathcal{O}$ is a set used to define the set of time steps where the prediction or its label status is incorrect. For fine-tuning the compressed models, we retained the same training configuration, including the learning rate, optimizer, and data split. However, to adapt the pruned models efficiently while preventing overfitting, we limited the retraining process to only 5 epochs. This decision was based on experimental observations showing that 5 epochs were sufficient for the model to recover performance after pruning, while further retraining offered minimal improvement. This approach ensures that the benefits of pruning are not offset by high retraining costs and was applied consistently across all pruning methods to enable fair comparison.

4.4. Comparative compression methods

To assess the effectiveness of our isomorphic structured pruning approach in terms of performance and compression, we compare it with the baseline trained model, serving as the primary reference for performance degradation as well as with the unstructured and conventional structured pruning methods which have been described in Section 2.

Another comparative aspect we incorporated into our evaluation was applying post-training dynamic quantization to the pruned and fine-tuned model, which was obtained using the proposed ISP approach. Quantization reduces a model's precision by converting its weights and activations to a lower-bit integer representation. In our work, we applied post-training dynamic quantization, which focuses on quantizing only the model's weights at inference time, while activations remain in floating-point format (Choudhary et al., 2020). This approach is simple to apply and flexible, allowing developers to tailor quantization strategies to specific use cases. In our case, we quantized the model's weights from FP32, as used in the baseline model, to INT8.

5. Comparative experimental results

The experimental results presented in this section offer a comprehensive comparison of the proposed ISP technique with the baseline and the selected compression methods. The evaluation focuses on both disaggregation performance and computational complexity.

A comparative analysis of the selected pruning approaches, tested across various pruning thresholds, is presented in Fig. 2. The figure provides a comprehensive evaluation of the unstructured magnitude pruning, conventional structured pruning, isomorphic structured pruning and isomorphic structured pruning combined with dynamic quantization. These methods are assessed across a range of pruning thresholds from 0% to 95%, with respect to performance metrics, including F1-score and MAE, as well as model size measured in megabytes (MB) across both evaluated datasets. Regarding the 'pruning thresholds vs. performance' diagrams (F1,MAE), it is observed that isomorphic structured pruning approaches generally exhibit comparable performance to unstructured pruning for the majority of tested pruning thresholds, while conventional structured pruning tends to exhibit significantly poorer trade-offs. When comparing ISP and ISP combined with dynamic quantization, it becomes evident that dynamic quantization has no impact on performance, as shown by both the F1 and MAE diagrams. On the other hand, the 'pruning threshold vs model size' diagrams reveal some fundamental distinctions among the tested approaches. Specifically, unstructured pruning appears to preserve the original model size across all pruning thresholds since it does not physically remove any component from the model. In contrast, all the structured pruning approaches exhibit a hyperbolic relationship between the pruning threshold and model size as described in Section 4. Finally, the key difference between conventional structured pruning, isomorphic pruning, and isomorphic pruning with quantization lies in the resulting model size. In particular, the latter achieves a 4× reduction in size by quantizing model parameters from FP32 to INT8.

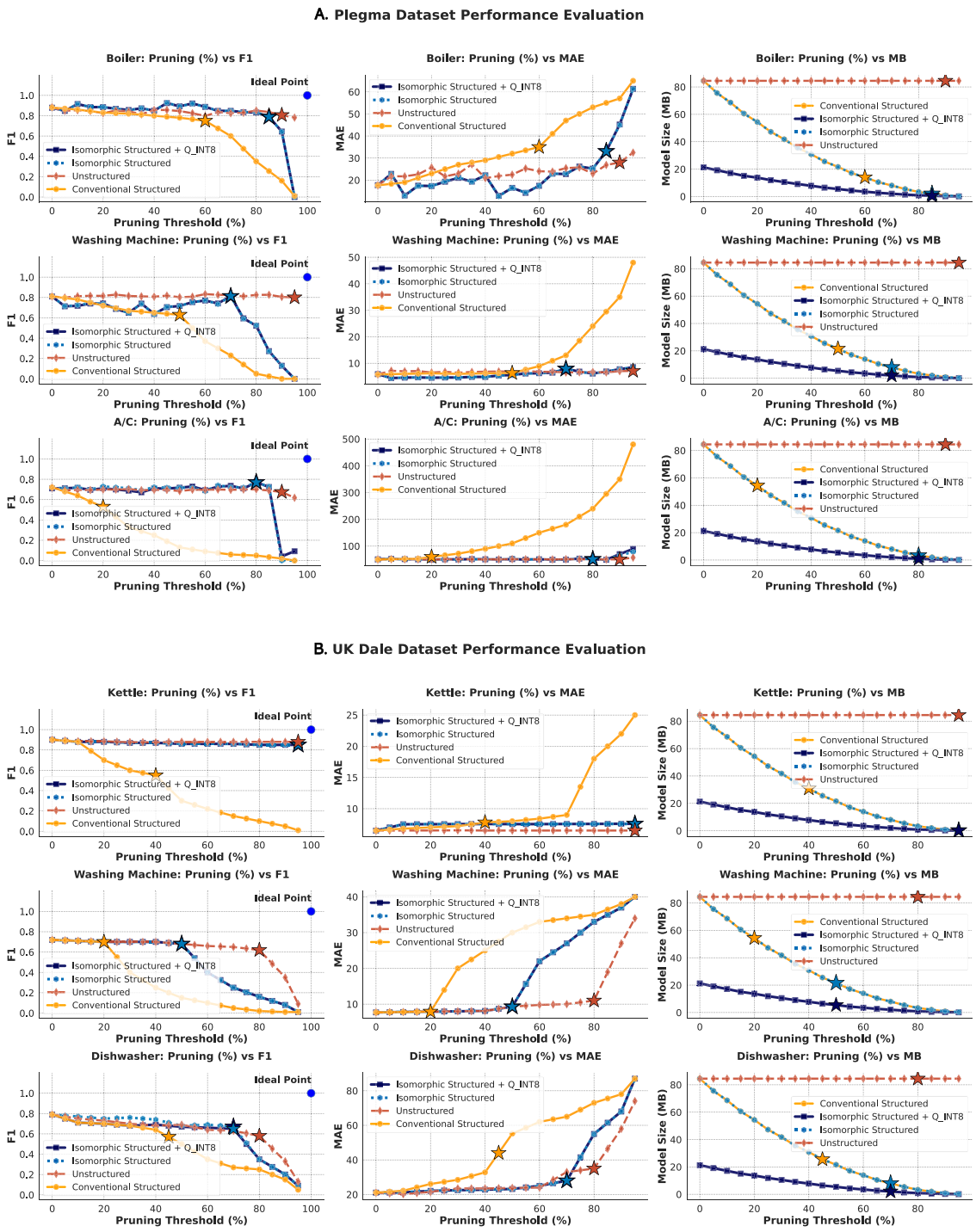

Fig. 2. The comparison of computational complexity (in terms of pruning percentage, 0%–95%) is analyzed against performance degradation metrics (F1-score, MAE) and model size (MB) for both the Plegma (top-A) and UK Dale (bottom-B) datasets. Stars indicate the optimal pruning ratio (p_{opt}) for Isomorphic Structured Pruning, Isomorphic Structured Pruning with Dynamic Quantization, Unstructured Magnitude-Based Pruning, and Conventional Structured Pruning. In the 'Pruning Threshold vs F1' diagrams, the black dot represents the theoretical "ideal" point for reference.

5.1. Plegma dataset: Performance evaluation results and discussion

Focusing on the results derived from the Plegma dataset, we arrive at several noteworthy observations regarding the performance of different pruning strategies and their performance on the different appliances. For the boiler, which shows a simple and consistent consumption pattern, unstructured pruning and ISP perform similarly, with optimal pruning at 90% and 85%, respectively. Conventional structured pruning performs slightly worse, with optimal pruning around 60%. A similar pattern is observed in the A/C appliance, where the optimal pruning thresholds are identified at 90% for unstructured pruning and 80% for isomorphic pruning approaches and 20% for the convetional structured pruning. In contrast, the washing machine's diverse operational states and highly variable consumption patterns result in a more pronounced performance gap between unstructured and structured pruning approaches. Specifically, unstructured pruning finds the optimal threshold at 95%, ISP (with and without quantization) at 70%, and conventional structured at 50%. This highlights the increased difficulty

Table 3
Comparative evaluation of performance and compression metrics across the baseline, unstructured, conventional structured, ISP, and ISP + INT8 quantization. For the structured pruning approaches, compression metrics show a hyperbolic compression pattern with respect to pruning percentage, as reducing both input and output dimensions by p_t% yields an actual pruning ratio of $1-(1-p_t\%)^2$.

Dataset	Appliance	Compression method	Pruning (%) $p_t = p_{opt}$	Performance metrics			Compression metrics		
				F1 Score	MAE	Acc.	Params ($M \times 10^6$)	Size (MB)	MACs ($G \times 10^9$)
Plegma	Boiler	Baseline Model	0 (%)	0.880	17.642	0.996	22.148	84.491	39.268
		Unstructured Pruning	90 (%)	0.850	27.980	0.994	22.148	84.491	39.268
		Conventional Structured Pruning	60 (%)	0.753	37.710	0.992	3.543	13.920	6.570
		ISP	85 (%)	0.778	33.077	0.991	0.525	2.022	0.928
		ISP + Q_INT8	**85 (%)**	**0.778**	**33.111**	**0.991**	**0.525**	**0.526**	**0.928**
	Washing Machine	Baseline Model	0 (%)	0.810	5.792	0.993	22.148	84.491	39.268
		Unstructured Pruning	95 (%)	0.801	7.127	0.992	22.148	84.491	39.268
		Conventional Structured Pruning	50 (%)	0.623	6.521	0.992	5.587	22.120	9.847
		ISP	70 (%)	0.811	7.895	0.993	2.091	7.994	3.689
		ISP + Q_INT8	**70 (%)**	**0.814**	**7.927**	**0.993**	**2.091**	**2.027**	**3.689**
	A/C	Baseline Model	0 (%)	0.717	50.199	0.964	22.148	84.491	39.268
		Unstructured Pruning	90 (%)	0.674	50.501	0.956	22.148	84.491	39.268
		Conventional Structured Pruning	20 (%)	0.534	68.702	0.969	14.174	54.148	25.243
		ISP	80 (%)	0.770	51.094	0.971	0.869	3.333	1.522
		ISP + Q_INT8	**80 (%)**	**0.769**	**50.995**	**0.971**	**0.869**	**0.855**	**1.522**
UK Dale	Kettle	Baseline Model	0 (%)	0.901	6.490	0.997	22.148	84.491	39.268
		Unstructured Pruning	95 (%)	0.889	6.560	0.995	22.148	84.491	39.268
		Conventional Structured Pruning	40 (%)	0.581	7.910	0.994	7.943	30.920	14.570
		ISP	95 (%)	0.864	6.870	0.995	0.055	0.211	0.098
		ISP + Q_INT8	**95 (%)**	**0.862**	**6.870**	**0.995**	**0.055**	**0.052**	**0.098**
	Washing Machine	Baseline Model	0 (%)	0.725	7.462	0.992	22.148	84.491	39.268
		Unstructured Pruning	80 (%)	0.682	11.456	0.990	22.148	84.491	39.268
		Conventional Structured Pruning	20 (%)	0.711	7.780	0.990	14.587	54.120	25.131
		ISP	50 (%)	0.698	8.523	0.991	5.591	21.194	9.817
		ISP + Q_INT8	**50 (%)**	**0.698**	**8.526**	**0.991**	**5.591**	**5.094**	**9.817**
	Dishwasher	Baseline Model	0 (%)	0.791	21.190	0.990	22.148	84.491	39.268
		Unstructured Pruning	80 (%)	0.782	36.220	0.988	22.148	84.491	39.268
		Conventional Structured Pruning	45 (%)	0.593	43.280	0.987	6.695	25.543	11.872
		ISP	70 (%)	0.638	28.710	0.988	2.058	7.980	3.652
		ISP + Q_INT8	**70 (%)**	**0.636**	**28.739**	**0.988**	**2.058**	**2.056**	**3.652**

of effectively structured pruning models when handling complex and dynamic consumption signals.

Referring to the recorded metrics in Table 3, all optimal pruned models demonstrate solid performance, with isomorphic structured approaches offering the best trade-offs between accuracy and compression. Specifically focusing on performance metrics, for the boiler, unstructured pruning results in a 3.4% decrease in F1 score and a MAE increase of 10 units. Isomorphic structured approaches cause a larger F1 drop of 11% and increase MAE by about 15 units, while conventional structured pruning performs worse, with a 14.4% F1 drop and a 20 units increase in MAE. For the washing machine, unstructured pruning slightly reduces F1 by 1.2%, with MAE increasing from 5.79 to 7.13. Notably, the isomorphic pruning approaches slightly improves the F1 score from the baseline by up to 0.49% and keeps MAE below 8.0. Conventional structured pruning, however, leads again to worse performance leading to a 23% decrease in F1 score. Finally For the A/C, unstructured pruning lowers F1 from 0.71 to 0.67, with MAE increasing marginally. ISP improves F1 to 0.770, with MAE rising only slightly to 51.1. Conventional structured pruning again shows the weakest performance, reducing F1 to 0.53.

The comparative performance evaluation results analyzed above can be visually verified through the consumption prediction diagrams presented in Fig. 3. These diagrams illustrate the predicted appliance consumption obtained from unstructured pruning, ISP, and ISP with dynamic quantization, compared against the baseline and ground truth data. From these curves, it can be observed that all approaches successfully reconstruct the consumption patterns of the appliances and accurately identify their activation and deactivation (on–off) status. Additionally, it is evident that dynamic quantization does not degrade the performance of ISP, as both approaches yield nearly identical predictions.

Analyzing the compression metrics presented in Table 3 reveals distinct trade-offs between the evaluated methods. Specifically, although unstructured pruning achieves the highest compression levels this is not translated in computational efficiency as pruned weights are simply replaced with zeros, leaving the model's parameters, size, and MACs effectively unchanged. In contrast, all structured pruning approaches exhibit substantial improvements across all evaluated metrics. Unlike unstructured pruning, these methods eliminate entire units from the model, resulting in a more compact and computationally efficient architecture. Notably, both isomorphic structured pruning and its quantized variant achieve a substantial 97% reduction in model parameters and MACs for the boiler, 90% for the washing machine, and 96% for the air conditioner (AC). In contrast, conventional structured pruning yields significantly lower compression metrics, as it prunes units based on comparisons across globally heterogeneous components of the model, often resulting in less optimal pruning decisions than those achieved by the proposed isomorphic approach.

5.2. UK Dale dataset: Performance evaluation results and discussion

Focusing on the results derived from the UK Dale dataset, a similar evaluation is conducted across the three appliances: kettle, washing machine, and dishwasher. The kettle, which exhibits a highly repetitive and short-duration power signature, enables the proposed pruning strategies to retain strong performance even at aggressive compression levels, similar to the behavior observed for the boiler in the Plegma dataset. This similarity can be attributed to the comparable consumption profiles and operational characteristics of the two appliances. Specifically, the optimal pruning threshold for unstructured pruning was experimentally identified at 95%, resulting in a minor F1-score decrease from 0.901 to 0.889. ISP achieves its best performance at the

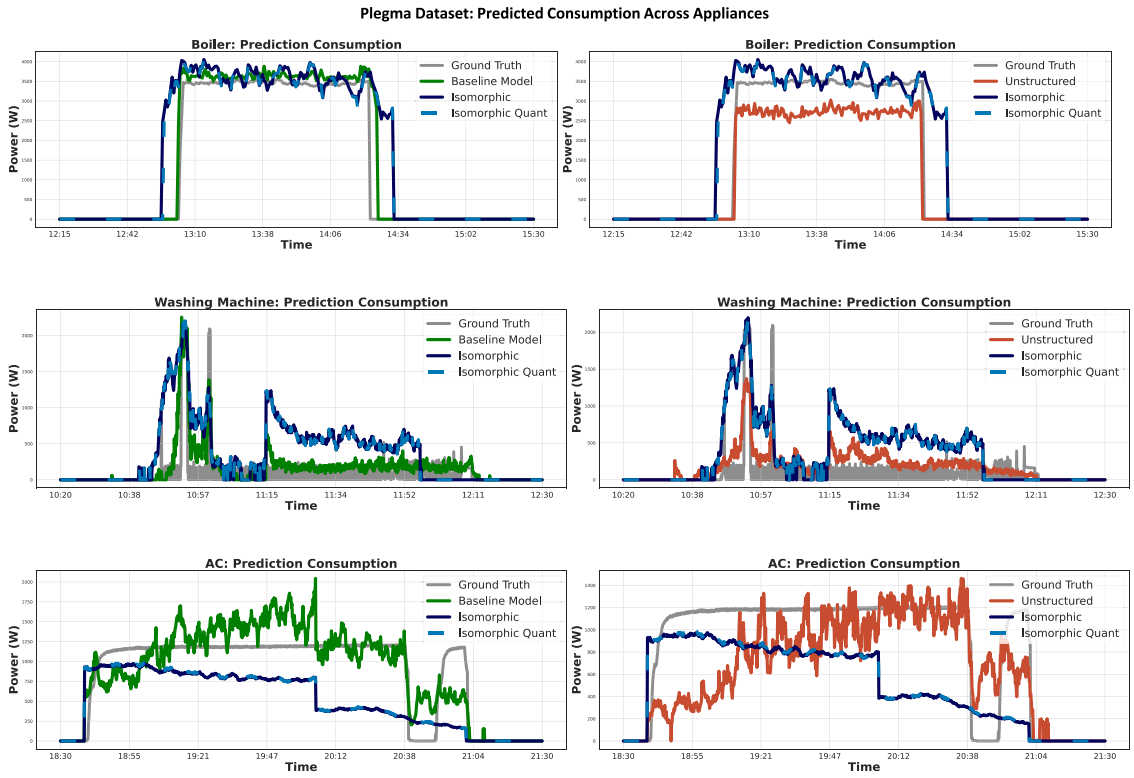

Fig. 3. Comparison of predicted consumption diagrams on Plegma Dataset using the proposed ISP and ISP with Dynamic Quantization against the Unstructured Pruning and the Baseline Model, with pruning thresholds set to $P_{\text{thr}} = P_{\text{opt}}$.

same pruning threshold (95%), maintaining an F1-score of 0.864, and demonstrates no degradation after applying quantization. In contrast, conventional structured pruning reaches its optimal point at 40%, yet this still leads to a substantial drop in performance, with the F1-score falling to 0.581. This reflects a poorer trade-off between pruning ratio and model accuracy, consistent with the observations from the Plegma dataset experiments.

For the washing machine, which exhibits longer operational cycles and more complex temporal patterns, unstructured pruning achieves optimal performance at a pruning threshold of 80%. ISP pruning and its quantized variant perform best at 50%, showing only minimal degradation in both F1 score and MAE. In contrast, conventional structured pruning again offers the weakest trade-off, with its optimal threshold identified at just 20%.

Finally, for the dishwasher, unstructured pruning achieves its optimal performance at 80%, resulting in a minimal F1-score drop from 0.791 to 0.782, while the MAE increases from 21.19 (baseline) to 36.22. ISP pruning and its quantized variant both achieve the best trade-off between pruning threshold and performance degradation at 70%, with an F1-score of 0.638 and a MAE of 28.71—indicating a moderate decline in accuracy but with significant compression gains. Conventional structured pruning provides the weakest performance–compression trade-off, with its optimal pruning threshold identified at 45%, resulting in an F1-score of just 0.593 and a substantial MAE increase to 43.28.

According to Fig. 4, all pruning approaches successfully reconstruct the appliance consumption signals and identify on–off transitions. The predictions from ISP pruning and its quantized variant remain virtually identical and closely align with the baseline model, highlighting the minimal impact these methods have on disaggregation performance and confirming their consistency across datasets.

6. Real world edge deployment & evaluation

In order to validate the practical feasibility of the ISP technique, we extended our evaluation by deploying the NILM models on two distinct industrial edge devices with varying computational capabilities. This section details the experimental setup, deployment workflow, and evaluation strategy used to benchmark the models in terms of inference latency, CPU utilization, and energy consumption, analyzing their performance across two different edge deployment scenarios.

6.1. Experimental system setup

The evaluation was carried out on two industrial edge devices with noticeably different computational capabilities, enabling a comparative analysis of the proposed ISP methodology under varying hardware constraints.

The first device used was the EI-52, a compact and high-performance edge intelligence system designed for IoT connectivity and edge analytics. It is powered by Intel's 11th Gen Core processors (i5), offering up to 4 cores and 8 threads with clock speeds reaching up to 4.4 GHz. This level of processing power makes the EI-52 suitable for computationally intensive tasks such as real-time data processing and machine learning inference. In contrast, the second device used in the experiments was the LEC-7230M, which, unlike the EI-52, has significantly lower computational power. The LEC-7230M features low-power Intel Atom processors, offering fewer cores and lower clock speeds, making it better suited for basic automation tasks and monitoring applications rather than complex data processing. Its processing capabilities are limited compared to the EI-52, which restricts its performance in more computationally demanding scenarios. The above are summarized in Table 4.

By including two devices with distinct computational power capabilities, we aimed to test our proposed methodology under different conditions. This approach allows us to assess how the proposed pruning technique performs across varying levels of processing power, providing valuable insights into its adaptability and efficiency.

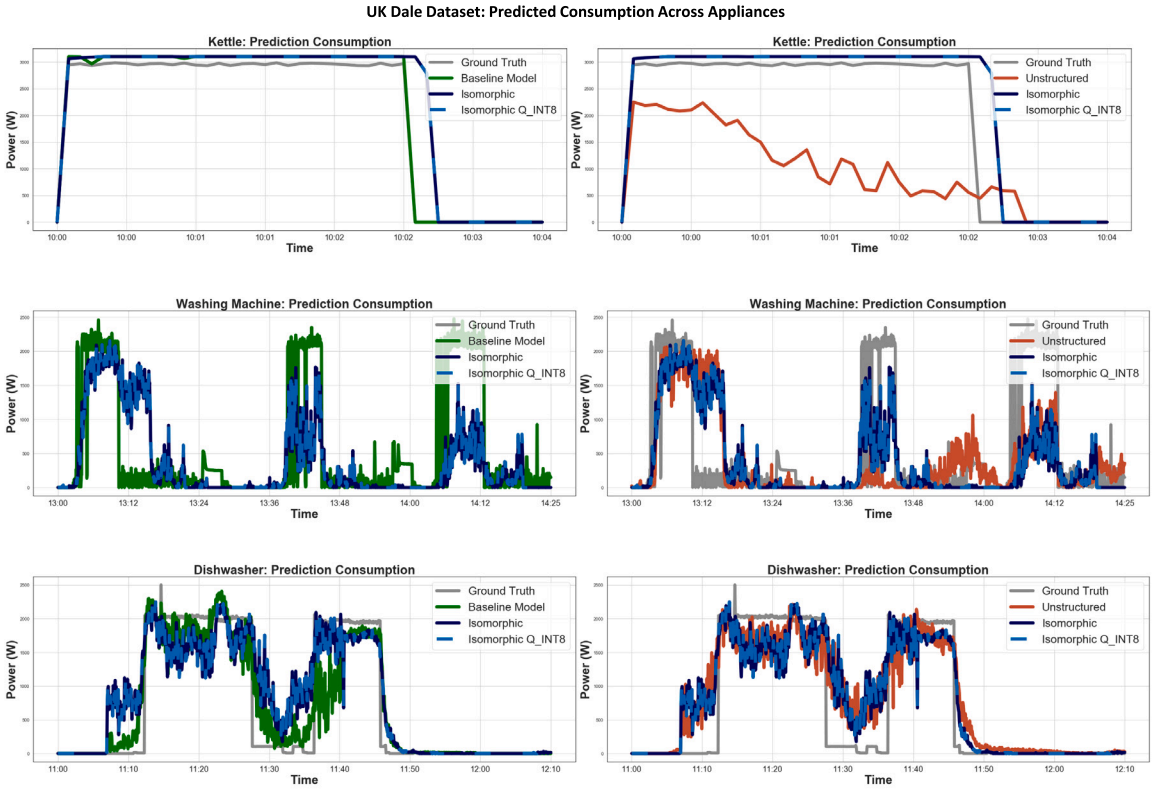

Fig. 4. Comparison of predicted consumption diagrams on UK Dale Dataset using the proposed Isomorphic Structured Pruning and Isomorphic Structured Pruning with Dynamic Quantization against the Unstructured Pruning and the Baseline Model, with pruning thresholds set to $P_{\text{thr}} = P_{\text{opt}}$.

Table 4
Comparison of EI-52 and LEC-7230M Industrial Edge Devices.

Feature	EI-52	LEC-7230M
Purpose	High Performance Edge Computing	Industrial automation/POS
Processing Power	High-performance (Intel 11th Gen Core)	Low to moderate (Intel Atom)
CPU	Intel Core i5-1145G7E	Intel Celeron J1900 quad-core
RAM	16GB	8GB
CPU Cores/Threads	4 cores/8 threads	4 cores/4 threads
Clock Speed	Up to 4.4 GHz (Turbo)	Up to 2.42 GHz
GPU/Graphics	Integrated Intel UHD Graphics	Basic graphics support
Usage Scenario	High computational tasks	Low computational tasks

6.2. Edge deployment workflow

In order to deploy the models on the selected edge devices, the application deployment was containerized using Docker, leveraging the Python 3.10-slim base image to minimize resource consumption while maintaining compatibility with the required libraries. Deployment and execution were systematically managed using Docker Compose, which ensured a consistent and reproducible setup across all devices. The containerized application was first uploaded to the Barbara library, a central repository for edge applications, from which it was subsequently deployed to the designated edge nodes using the Node Manager component. Upon successful deployment, the application was executed locally on each edge device, facilitating AI inference workloads using various model versions. During runtime, comprehensive system performance metrics, including computational efficiency, latency, and resource utilization, were collected alongside power consumption data.

6.3. Experimental design for inference evaluation

To evaluate the efficiency and resource demands of the deployed models on edge devices, we recorded key inference-related metrics, including inference time, memory usage, CPU utilization, and energy consumption. These metrics provide comprehensive insights into the performance and computational efficiency of each model.

Inference time was measured by performing a forward pass through the selected model using random inputs to generate an output. To eliminate initialization effects, a warm-up phase was conducted, running the model for five iterations. After this warm-up, the model executed 50 forward passes with random inputs, and the inference time for each pass was recorded in milliseconds. The final inference time was calculated as the mean of the measured times from these 50 runs. This approach ensured accurate and consistent timing data while minimizing the impact of initialization and caching.

CPU utilization was recorded during both the idle phase (after loading the model into memory) and the inference phase (during forward passes). This dual-phase measurement provided insights into the baseline computational load and the additional processing demands during inference.

Additionally, energy consumption was recorded on the EI-52 edge device to assess the impact of different compression methodologies on the model's energy footprint—an essential consideration for edge device deployments. In order to accurately capture power metrics, measurements were continuously recorded during three distinct operational phases:

Table 5
Telemetry metrics collected from edge devices during idle and inference phases for Plegma Dataset experiments.

Edge Device	Appliance	Model	Inference Time (ms)	CPU (%) Idle	CPU (%) Inference
El-52	Boiler	Baseline	27.22 ± 0.80	1.33	4.62
		Unstructured (90%)	27.16 ± 0.45	1.18	4.77
		ISP (85%)	1.58 ± 0.27	0.92	3.21
		ISP (85%) + Q_INT8	1.20 ± 0.10	1.07	3.05
	Washing Machine	Baseline	27.24 ± 0.40	1.48	4.62
		Unstructured (95%)	27.19 ± 0.41	1.48	4.53
		ISP (70%)	4.37 ± 0.14	0.92	3.06
		ISP (70%) + Q_INT8	3.78 ± 0.08	1.22	3.20
	AC	Baseline	27.36 ± 0.46	1.33	4.77
		Unstructured (90%)	26.65 ± 0.40	1.48	4.48
		ISP (80%)	1.92 ± 0.15	0.92	2.77
		ISP (80%) + Q_INT8	1.85 ± 0.07	1.07	3.06
LEC-7230M	Boiler	Baseline	294.22 ± 20.24	4.43	26.45
		Unstructured (90%)	292.43 ± 21.77	4.42	26.55
		ISP (85%)	20.31 ± 5.36	2.32	4.67
		ISP (85%) + Q_INT8	18.22 ± 4.58	2.01	4.22
	Washing Machine	Baseline	293.62 ± 23.65	4.44	26.05
		Unstructured (95%)	292.13 ± 17.13	4.45	26.55
		ISP (70%)	45.61 ± 4.91	2.55	5.75
		ISP (70%) + Q_INT8	41.58 ± 4.75	2.34	5.17
	AC	Baseline	301.48 ± 27.65	4.67	25.05
		Unstructured (90%)	291.01 ± 22.42	4.42	26.44
		ISP (80%)	25.98 ± 7.33	2.65	4.62
		ISP (80%) + Q_INT8	23.98 ± 6.87	2.38	4.25

- **Idle State:** Capturing the baseline energy consumption prior to inference.
- **Inference Execution State:** Measuring active power usage during the inference process.
- **Post-Execution State:** Quantifying residual energy usage after the inference has completed.

To ensure statistically significant results, each model was subjected to 10,000 inference iterations. This high iteration count was necessary because pruned and quantized models complete inference significantly faster than baseline models, requiring a larger sample size to obtain stable and reliable energy estimates.

6.4. Computational performance results across edge devices

To comprehensively assess the impact of the proposed compression strategies, this study extended the evaluation to real-world conditions by deploying the models in the two scenarios described in above. For the structured pruning approaches, we focused solely on the isomorphic structured and isomorphic structured with quantization variants, excelling performance, as demonstrated in Section 5. Overall, the telemetry metrics recorded from the edge devices as shown in Table 5, exhibited trends consistent with the earlier compression evaluation results. In both scenarios, the isomorphic approaches consistently enhanced computational performance across all recorded metrics compared to the baseline model, while unstructured pruning showed no improvement.

In the high-performance edge device scenario (EL-52), the proposed ISP and its combination with quantization achieved a remarkable reduction in inference time — up to 17 times faster for the boiler appliance — compared to both the baseline and unstructured pruning models. The analysis of CPU metrics in idle and inference phase, reveals that the proposed ISP and ISP with quantization exhibit a significantly lower increase in CPU usage compared to both the baseline and unstructured pruning methods. Specifically, the proposed methods show an increase in CPU usage of approximately 2.8 times, while both the baseline and unstructured pruning methods result in similar increases in CPU usage, averaging 3.54 times across all devices.

Focusing on the deployment scenario involving the moderate-to-low performance edge device (LEC-7230M), we observe that although the impact of the proposed compression methodologies is similar to that of the high-performance edge device (EL-52), the absolute values for inference time and CPU usage are significantly higher. This difference is primarily attributed to the reduced computational performance of the LEC-7230M device. Specifically, for inference time, the baseline and unstructured pruned model on LEC-7230M fluctuate around 300 ms, compared to 27 ms on the high-performance edge device.

Continuing with the analysis of CPU usage, the disparity between the high-performance edge device (EL-52) and the moderate-to-low performance device (LEC-7230M) is also evident in terms of computational efficiency. On the LEC-7230M, the baseline model exhibits a significant increase in CPU utilization during inference, spiking from approximately 4% after model loading (idle phase) to around 26.5% during inference. This increase is notably higher than that observed on the EL-52 device, where the CPU usage during inference rises from about 1.4% to 4.7%. While CPU utilization remains higher overall on the LEC-7230M, both ISP and its quantized variant reduce peak usage compared to the baseline and unstructured pruning. Notably, ISP with quantization brings peak CPU usage down from 26% to ≈ 5%—a significant improvement, despite still exceeding the EL-52's levels. Finally, the last analysis focused on evaluating the impact of different compression methodologies on the energy consumption during each model's inference phase. Energy consumption is a critical metric in edge device deployment, as it directly affects the operational cost, battery life, and overall sustainability of the system. The experimental results of the 4 different models (baseline, unstructured pruned, isomorphic structured pruned and isomorphic structured pruned with dynamic quantization) for the AC appliance on EL-52 edge device derived using the experiment described in Section 6.3 and they are presented in Fig. 5.

According to the results presented in this figure, clear differences are observed during the inference phase, which evaluates the energy consumption of each deployed model. Notably, the baseline and unstructured pruned models exhibit similar behavior — consistent with earlier metrics — since their structural configuration, including parameter count and computational complexity, remains effectively unchanged. On the other hand, significant changes are observed in the isomorphic structured pruned and quantized models. Although their peak energy consumption remains comparable to that of the baseline and unstructured pruned models—fluctuating around 34 Watts—the proposed

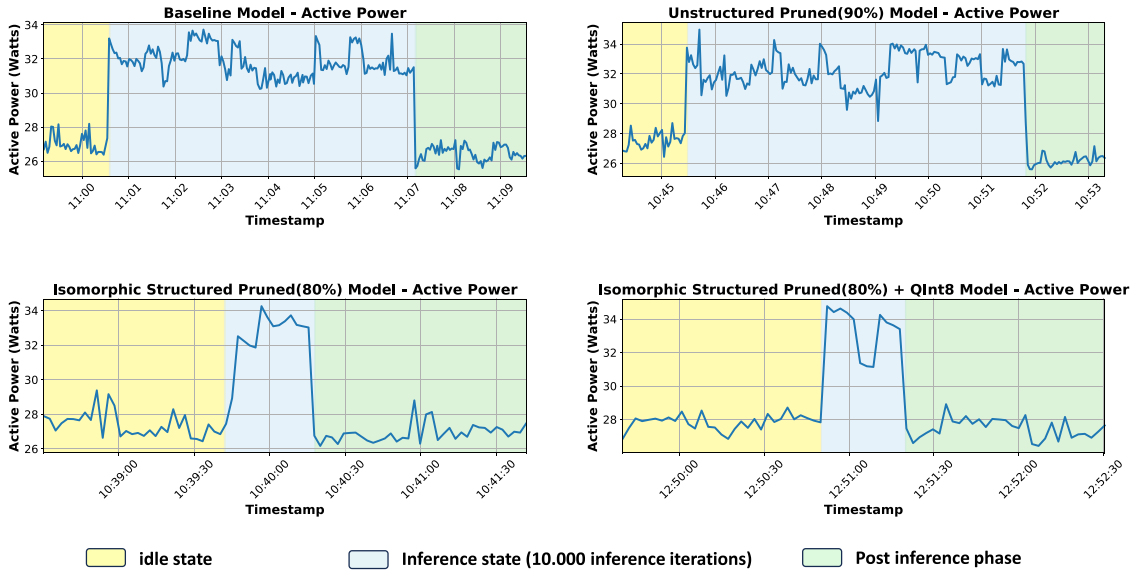

Fig. 5. Comparison of energy consumption among the baseline, unstructured pruned, isomorphic structured pruned, and isomorphic structured pruned & quantized model for the A/C appliance as deployed in the El-52 device. To effectively assess energy consumption, we continuously recorded power usage in three distinct operational states: idle state (yellow), inference execution state (black), and post-inference state (green). To ensure statistically significant results, each model underwent 10,000 inference iterations, providing a robust evaluation of energy efficiency across configurations.

models exhibit significantly lower energy consumption overall due to their substantially reduced inference time. Specifically, the baseline and unstructured pruned models take approximately 6.45 min to complete the inference phase experiment, which consists of 10,000 inference iterations while the proposed ISP and the ISP with quantization complete the same task in just around 1 min. This difference translates to approximately 34.8×10^{-4} kWh for the baseline and unstructured pruned models, compared to just 5.55×10^{-4} kWh for the isomorphic structured pruned model and 5.15×10^{-4} kWh for the isomorphic structured pruned model with dynamic quantization. Therefore, the proposed techniques achieve an overall energy consumption reduction of up to 85% during the inference phase, making them highly suitable for low-power edge deployments where energy efficiency is a critical concern.

7. Conclusions

This work introduced the ISP technique within the energy disaggregation domain, aiming to improve the computational efficiency and energy consumption of DL models used for NILM that are deployed on edge devices. Its evaluation was done through extensive experimentation considering two datasets; the Plegma and UK Dale. The former focused on Mediterranean region-based devices such as air conditioners and boilers, while the latter focused on widely-used machines such as dishwasher and washing machine, all being workload heavy and hence energy consuming. From the experimental results on both datasets, it was demonstrated that ISP allows for even better trade-off between pruning ratio and computational performance resulting in up to than x11 times reduced parameters for the appliances tested. In addition, it was shown that when coupled with INT8 quantization, ISP technique can reduce model size to approximately 5MB for all appliances considered, making the technique attractive for edge-constrained devices with limited computational resources. Further, to fully demonstrate the applicability of ISP, this work extended its scope by evaluating ISP in real-world deployment scenarios using two distinct industrial edge devices. The results demonstrated notable improvements in inference time, CPU utilization, and energy consumption, highlighting the strong practical viability of in real-world settings.

The results and findings presented in this work while focused on NILM and the energy disaggregation domain, they can, in fact, capture a wider application scenario that falls within the IoT domain. Industrial IoT (IIoT) devices, such as smart meters, IoT gateways, and grid-edge controllers, are increasingly integrated into infrastructures that require continuous communication, often over limited-bandwidth networks like LTE, RF mesh, or NB-IoT. On top of that, these devices face both hardware and power constraints and also demand real-time, on-device intelligence to process the vast amount of data generated at the edge. Therefore, ISP can stand out as a technique to effectively make DL models compact and computationally efficient, making it well-suited for such scenarios, setting a scalable path towards deployable DL models in resource-limited environments.

CRediT authorship contribution statement

Sotirios Athanasoulias: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Nikos Temenos:** Writing – review & editing, Supervision. **Ilias Kappos:** Software. **Isidoros Kokos:** Supervision. **Pedro Antonio Garcia-Abadillo Navaro:** Software. **Nikolaos Ipiotis:** Supervision. **Anastasios Doulamis:** Supervision. **Nikolaos Doulamis:** Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Sotrios Athanasoulias reports financial support was provided by European Commission. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was funded by the European Union's Horizon Europe programme (HORIZON-MISS-2023-CIT-01) EXPEDITE – 'Enabling Positive Energy Districts through a Planning and Management Digital Twin', Grant Agreement No 101139527, and ODEON, Grant Agreement No 101136128.

Data availability

No data was used for the research described in the article.

References

Ahmed, S., Bons, M., 2020. Edge computed NILM: A phone-based implementation using MobileNet compressed by tensorflow lite. In: Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring. Association for Computing Machinery, New York, NY, USA, pp. 44–48.

Angelis, G.F., Timplalexis, C., Salamanis, A.I., Krinidis, S., Ioannidis, D., Kehagias, D., Tzovaras, D., 2023. Energformer: A new transformer model for energy disaggregation. IEEE Trans. Consum. Electron. 69 (3), 308–320.

Athanasoulias, S., Guasselli, F., Doulamis, N., Doulamis, A., Ipiotis, N., Katsari, A., Stankovic, L., Stankovic, V., 2024a. The plegma dataset: Domestic appliance-level and aggregate electricity demand with metadata from Greece. Sci. Data 11 (1), 376. http://dx.doi.org/10.1038/s41597-024-03208-0.

Athanasoulias, S., Sykiotis, S., Kaselimi, M., Doulamis, A., Doulamis, N., Ipiotis, N., 2024b. OPT-NILM: An iterative prior-to-full-training pruning approach for cost-effective user side energy disaggregation. IEEE Trans. Consum. Electron. 70 (1), 4435–4446. http://dx.doi.org/10.1109/TCE.2023.3324493.

Athanasoulias, S., Sykiotis, S., Temenos, N., Doulamis, A., Doulamis, N., 2024c. 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, pp. 249–253. http://dx.doi.org/10.1109/ICASSPW62465.2024.10626463.

Athanasoulias, S., Temenos, N., Doulamis, N., Doulamis, A., Kokos, I., Ipiotis, N., 2024d. Towards edge-computed NILM: Insights from a mediterranean use case. In: 2024 3rd International Conference on Energy Transition in the Mediterranean Area. SyNERGY MED, pp. 1–5. http://dx.doi.org/10.1109/SyNERGYMED62435.2024.10799297.

Bao, K., Ibrahimov, K., Wagner, M., Schmeck, H., 2018. Enhancing neural non-intrusive load monitoring with generative adversarial networks. Energy Informatics 1, 295–302.

Barber, J., Cuayáhuitl, H., Zhong, M., Luan, W., 2020. Lightweight non-intrusive load monitoring employing pruned sequence-to-point learning. In: Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring. 1, Association for Computing Machinery, New York, NY, USA, pp. 11–15. http://dx.doi.org/10.1145/3427771.3427845.

Choudhary, T., Mishra, V., Goswami, A., Sarangapani, J., 2020. A comprehensive survey on model compression and acceleration. Artif. Intell. Rev. 53 (7), 5113–5155. http://dx.doi.org/10.1007/s10462-020-09816-7.

Commission, E., for Energy, D.-G., Alaton, C., Tounquet, F., 2020. Benchmarking smart metering deployment in the EU-28 – Final report. Publications Office, http://dx.doi.org/10.2833/492070.

Cruz-Rangel, D., Ocampo-Martinez, C., Diaz-Rozo, J., 2024. Online non-intrusive load monitoring: A review. Energy Nexus 100348.

Fang, G., Ma, X., Mi, M.B., Wang, X., 2024. Isomorphic pruning for vision models. In: European Conference on Computer Vision. Springer, pp. 232–250.

García, D., Pérez, D., Papapetrou, P., Díaz, I., Cuadrado, A.A., Enguita, J.M., Domínguez, M., 2024. Conditioned fully convolutional denoising autoencoder for multi-target NILM. Neural Comput. Appl. 1–15.

Hart, G.W., 1992. Nonintrusive appliance load monitoring. Proc. IEEE 80 (12), 1870–1891.

He, Y., Xiao, L., 2023. Structured pruning for deep convolutional neural networks: A survey. IEEE Trans. Pattern Anal. Mach. Intell. 46 (5), 2900–2919.

Hosseini, S.H., Tsolakis, A., Alagumalai, A., Mahian, O., Lam, S.S., Pan, J., Peng, W., Tabatabaei, M., Aghbashlo, M., 2023. Use of hydrogen in dual-fuel diesel engines. Prog. Energy Combust. Sci. 98, 101100. http://dx.doi.org/10.1016/j.pecs.2023.101100, URL https://www.sciencedirect.com/science/article/pii/S0360128523000308.

Huber, P., Calatroni, A., Rumsch, A., Paice, A., 2021. Review on deep neural networks applied to low-frequency nilm. Energies 14 (9), 2390.

Hwang, H., Kang, S., 2022. Nonintrusive load monitoring using an LSTM with feedback structure. IEEE Trans. Instrum. Meas. 71, 1–11.

Kaselimi, M., Protopapadakis, E., Voulodimos, A., Doulamis, N., Doulamis, A., 2022. Towards trustworthy energy disaggregation: A review of challenges, methods, and perspectives for non-intrusive load monitoring. Sensors 22 (15), http://dx.doi.org/10.3390/s22155872, URL https://www.mdpi.com/1424-8220/22/15/5872.

Kaselimi, M., Voulodimos, A., Protopapadakis, E., Doulamis, N., Doulamis, A., 2020. Energan: A generative adversarial network for energy disaggregation. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 1578–1582.

Kee, K.-K., Lim, Y.S., Wong, J., Chua, K.H., 2019. Non-intrusive load monitoring (NILM) – a recent review with cloud computing. In: 2019 IEEE International Conference on Smart Instrumentation, Measurement and Application. ICSIMA, pp. 1–6. http://dx.doi.org/10.1109/ICSIMA47653.2019.9057316.

Kelly, J., Knottenbelt, W., 2015. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. Sci. Data 2, http://dx.doi.org/10.1038/sdata.2015.7.

Kukunuri, R., Aglawe, A., Chauhan, J., Bhagtani, K., Patil, R., Walia, S., Batra, N., 2020. EdgeNILM: Towards NILM on edge devices. In: Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation. BuildSys '20, Association for Computing Machinery, New York, NY, USA, pp. 90–99. http://dx.doi.org/10.1145/3408308.3427977.

Kumar, A., Shaikh, A.M., Li, Y., Bilal, H., Yin, B., 2021. Pruning filters with L1-norm and capped L1-norm for CNN compression. Appl. Intell. 51 (2), 1152–1160.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.

Lee, J., Park, S., Mo, S., Ahn, S., Shin, J., 2020. Layer-adaptive sparsity for the magnitude-based pruning. arXiv preprint arXiv:2010.07611.

Li, Y., Adamczewski, K., Li, W., Gu, S., Timofte, R., Van Gool, L., 2022. Revisiting random channel pruning for neural network compression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 191–201.

Li, H., Li, C., Xue, M., Fang, G., Zhou, S., Feng, Z., Wang, H., Wang, Y., Cheng, L., Song, M., et al., 2024. PruningBench: A comprehensive benchmark of structural pruning. arXiv preprint arXiv:2406.12315.

Liao, Z., Quétu, V., Nguyen, V.-T., Tartaglione, E., 2023. Can unstructured pruning reduce the depth in deep neural networks? In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1402–1406.

Liu, Z., Mu, H., Zhang, X., Guo, Z., Yang, X., Cheng, K.-T., Sun, J., 2019. Metapruning: Meta learning for automatic neural network channel pruning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3296–3305.

Lu, Z., Cheng, Y., Zhong, M., Luan, W., Ye, Y., Wang, G., 2022. Lightnilm: lightweight neural network methods for non-intrusive load monitoring. In: Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation. BuildSys '22, Association for Computing Machinery, New York, NY, USA, pp. 383–387. http://dx.doi.org/10.1145/3563357.3566152.

Molchanov, P., Mallya, A., Tyree, S., Frosio, I., Kautz, J., 2019. Importance estimation for neural network pruning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11264–11272.

Murray, D., Stankovic, L., Stankovic, V., 2017. An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study. Sci. Data 4, 160122. http://dx.doi.org/10.1038/sdata.2016.122.

Pan, Z., Wang, H., Li, C., Wang, H., Zhao, J., 2024. Perfednilm: a practical personalized federated learning-based non-intrusive load monitoring. Ind. Artif. Intell. 2 (1), 4.

Ribeiro, A.H., Tiels, K., Aguirre, L.A., Schön, T., 2020. Beyond exploding and vanishing gradients: analysing RNN training using attractors and smoothness. In: International Conference on Artificial Intelligence and Statistics. PMLR, pp. 2370–2380.

Rong, J., Wang, C., Zhou, Q., He, Y., Wu, H., 2025. Enhancing non-intrusive load monitoring through transfer learning with transformer models. Energy Build. 115334.

Schirmer, P.A., Mporas, I., 2022. Non-intrusive load monitoring: A review. IEEE Trans. Smart Grid 14 (1), 769–784.

Sun, Y., Zheng, L., Wang, Q., Ye, X., Huang, Y., Yao, P., Liao, X., Jin, H., 2022. Accelerating sparse deep neural network inference using GPU tensor cores. In: 2022 IEEE High Performance Extreme Computing Conference. HPEC, pp. 1–7. http://dx.doi.org/10.1109/HPEC55821.2022.9926300.

Sykiotis, S., Athanasoulias, S., Kaselimi, M., Doulamis, A., Doulamis, N., Stankovic, L., Stankovic, V., 2023. Performance-aware NILM model optimization for edge deployment. IEEE Trans. Green Commun. Netw. 7 (3), 1434–1446. http://dx.doi.org/10.1109/TGCN.2023.3244278.

Sykiotis, S., Kaselimi, M., Doulamis, A., Doulamis, N., 2022. Electricity: An efficient transformer for non-intrusive load monitoring. Sensors 22 (8), http://dx.doi.org/10.3390/s22082926, URL https://www.mdpi.com/1424-8220/22/8/2926.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Adv. Neural Inf. Process. Syst. 30.

Wang, X., Zhou, H., Freris, N.M., Zhou, W., Guo, X., Liu, Z., Ji, Y., Li, X.-Y., 2021. LCL: Light contactless low-delay load monitoring via compressive attentional multi-label learning. In: 2021 IEEE/ACM 29th International Symposium on Quality of Service. IWQOS, pp. 1–6. http://dx.doi.org/10.1109/IWQOS52092.2021.9521262.

Wu, J., Tang, X., Zhou, D., Deng, W., Cai, Q., 2024. Application of improved DBN and GRU based on intelligent optimization algorithm in power load identification and prediction. Energy Informatics 7 (1), 36.

Xuan, Y., Pang, C., Yu, H., Zeng, X., Chen, Y., 2024. An enhanced bidirectional transformer model with temporal-aware self-attention for short-term load forecasting. IEEE Access.

Yue, Z., Witzig, C.R., Jorde, D., Jacobsen, H.-A., 2020a. BERT4NILM: A bidirectional transformer model for non-intrusive load monitoring. In: Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring. NILM '20, Association for Computing Machinery, New York, NY, USA, pp. 89–93. http://dx.doi.org/10.1145/3427771.3429390.

Yue, Z., Witzig, C.R., Jorde, D., Jacobsen, H.-A., 2020b. Bert4nilm: A bidirectional transformer model for non-intrusive load monitoring. In: Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring. pp. 89–93.

Zhang, C., Zhong, M., Wang, Z., Goddard, N., Sutton, C., 2018. Sequence-to-point learning with neural networks for non-intrusive load monitoring. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence. In: AAAI'18/IAAI'18/EAAI'18, AAAI Press, New Orleans, Louisiana, USA.