Sequence-to-Segments-to-Sequence Learning with Neural Networks for Non-Intrusive Load Monitoring

Yechun Ruan
Sun Yat-Sen University
Guangzhou, Guangdong, China

Qianyi Huang
Sun Yat-Sen University
Guangzhou, Guangdong, China

Abstract

In order to better understand the power consumption of each appliance at home, Non-Invasive Load Monitoring (NILM) aims to extract per-appliance power consumption from aggregated whole-home power readings. In recent years, deep learning-based approaches have modeled the problem as a sequence-to-sequence (Seq2Seq) learning task, showing state-of-the-art performance for NILM. However, existing Seq2Seq models perform prediction on short sequences and treat each point in the sequence with equal importance. This is inefficient as only some representative samples in the sequence are more informative than the rest. Furthermore, these short sequences lack global information such as appliances' ON/OFF moments and duty cycle, leading to sub-optimal model performance. To address this issue, this paper introduces the Sequence-to-Segments-to-Sequence (Seq2Seg2Seq) scheme, which conducts segment-wise feature extraction. Specifically, the input signal is divided into multiple non-overlapping segments, followed by intra-segment feature extraction and inter-segment feature interaction. The Seq2Seg2Seq scheme can handle sequences that are an order of magnitude longer and enables long-time context awareness, with affordable resources on an IoT device. Experimental results on two real-world datasets, REDD and UK-DALE, demonstrate that our model exhibits better generalization capability, achieving 11% – 18% MAE gain and 11% – 24% SAE_δ gain over the state-of-the-art models. Furthermore, our model is more efficient and has a lower latency.

CCS Concepts

• **Computing methodologies** → **Neural networks**; • **Human-centered computing** → **Ambient intelligence**.

Keywords

Non-intrusive load monitoring, Source separation, Deep learning

ACM Reference Format:
Yechun Ruan and Qianyi Huang. 2025. Sequence-to-Segments-to-Sequence Learning with Neural Networks for Non-Intrusive Load Monitoring. In *Proceedings of The 12th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '25).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3736425.3770095

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
BuildSys '25, Golden, CO, USA
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-1945-5/25/11
https://doi.org/10.1145/3736425.3770095

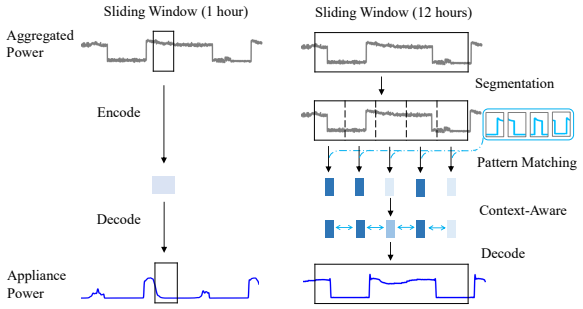
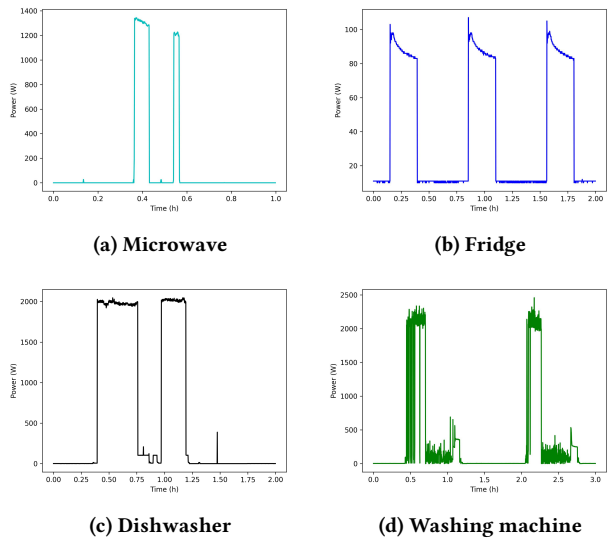


Figure 1: Pipeline for Seq2Seq (left) and Seq2Seg2Seq (right). Seq2Seg2Seq uses a larger sliding window and performs feature extraction for each segment.

1 Introduction

With the increasing demand for electricity and the growing awareness of energy conservation, there has been a surge of interest in smart grids. Studies indicate that accurate and fine-grained energy consumption monitoring and analysis can help detect faulty devices and optimize load scheduling, leading to about 20% potential energy savings [14, 22]. Energy disaggregation [6], also known as Non-Intrusive Load Monitoring (NILM), has emerged as an essential research topic. Through energy disaggregation, the entire household energy consumption, reported through a single smart meter, can be broken down into the energy usage of individual appliances. This avoids deploying individual sensors for each appliance. Due to data privacy and security concerns, there is a growing preference for processing data locally on edge devices. However, energy disaggregation is a very challenging task due to its inherent single-channel blind source separation (BSS) nature, where multiple sources need to be extracted from a single observation. In NILM, uncertain factors, such as power line noise, the diversity of appliances, and the overlapping active periods of multiple appliances, make the prediction problem more complex.

Deep learning techniques have significantly advanced the performance of NILM, which is far superior to traditional methods. Existing deep learning methods model the NILM problem as a sequence-to-sequence (Seq2Seq) learning task, mapping the whole home electricity consumption into the electricity consumption of individual appliances. However, existing Seq2Seq models accept short input sequences (usually with a duration not exceeding one hour), and treat each time sample as equally important. Although this approach helps to capture subtle changes in the signal, as input sequences contain only local information, global information such as the ON/OFF moments and duty cycles of the appliances may be lost, especially for appliances with long operating cycles (e.g.,



(a) Microwave (b) Fridge

(c) Dishwasher (d) Washing machine

Figure 2: Snippets of individual appliances' power consumption in UK-DALE house 2

washing machines, refrigerators). As shown in Figure 1 (left), the input signal remains stable within the sliding window, making it difficult for the model to extract useful information about the appliance. Due to the inherent limitations of smart meters in terms of computational capacity and memory size, long sequence processing models such as Big Bird[27], Informer[31] and Pyraformer[15] also encounter challenges when deployed on smart meters, especially when dealing with input sequences longer than a few thousand time steps.

Figure 2 gives snippets of the power consumption of some appliances from the UK-DALE dataset. We can observe that the majority of time samples in the sequence contain redundant information. We are only concerned about some representative samples in the sequence, such as the state transitions. Thus, we argue that fine-grained sample-wise feature extraction on short sequences is unnecessary and inefficient.

In this paper, we propose the Sequence-to-Segments-to-Sequence (Seq2Seg2Seq) learning scheme, which performs segment-wise feature extraction on long sequences. We segment long input sequence into non-overlapping segments and then perform feature extraction on segments. In other words, the fundamental unit for feature extraction is not per sample but per segment. In this way, the model can handle sequences that are an order of magnitude longer and enables long-time context awareness.

As shown in Figure 1 (right), the input sequence is divided into multiple non-overlapping segments, followed by intra-segment feature extraction and inter-segment feature interaction. We design an Appliance Power Pattern Extraction module for capturing both the global information between segments and the shape features within each segment. When the input signal has a length of S (e.g., 6300 time steps, which covers 10.5 hours), existing Seq2Seq models extract features for S (or $S/2$) units. In comparison, Seq2Seg2Seq scheme divides the input signal into N (e.g., 12) segments, each with length M ($M = S/N$) and extracts features for the N units. Compared with previous Seq2Seq schemes, our Seq2Seg2Seq scheme can handle long sequences more efficiently, i.e., long-time context awareness is achieved using less computational resources. This approach has a clear advantage when dealing with long-time span information.

Our main contributions are summarized as follows:

- We propose Seq2Seg2Seq scheme, which performs segment-wise feature extraction on input signals, rather than sample-wise feature extraction as in existing approaches. In this way, the model can achieve long-time context awareness with lower computational resources.
- Accordingly, we design a Seq2Seg2Seq model that contains an Appliance Power Pattern Extraction module to capture the global information among segments and the shape features within each segment, achieving efficient intra-segment feature extraction and inter-segment feature interaction.
- Experimental results on two real-world datasets REDD [13] and UK-DALE [11] show that our model exhibits better generalization capability where it achieves 11%-18% MAE gain and 11%-24% SAE_δ gain over the state-of-the-art models. In addition, our model has higher efficiency and lower latency.

We release the PyTorch implementation of our Seq2Seg2Seq model in a public repository to ensure reproducibility and facilitate future research. The code is available at https://github.com/ruanych/seq2seg2seq-nilm.

2 Related Works

The concept of NILM was initially introduced by Hart in 1992. The early approach relied on manually observing appliance power consumption levels and employing Finite-State Appliance Models for load monitoring. However, the efficacy of this method diminished when confronted with a diverse array and large quantity of appliances. Subsequently, machine learning techniques such as variations of Hidden Markov Models (HMM) [12, 17], Support Vector Machines (SVM) [7], and other signal processing methods [1, 29] have contributed to enhanced load monitoring.

In recent years, deep learning has propelled NILM to unprecedented performance heights. Techniques involving Recurrent Neural Networks (RNNs), notably Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), emerge as natural contenders for handling sequential data. The literature introduces methodologies based on RNN [11], LSTM [9, 10], and GRU [18]. However, the parallel efficiency of RNNs is a bottleneck. Concurrently, Convolutional Neural Networks (CNNs) have gained prominence. Several studies (e.g., Kelly and Knottenbelt [11], Zhang et al. [28]) have showcased the ability of CNNs to extract salient features from input signals, capturing ON/OFF moments, appliance usage durations, and power levels. This led to a proliferation of CNN-based endeavors. SGN [23] introduces a sub-network for appliance ON/OFF states, significantly boosting power signal identification efficiency. SCANet [3] uses dilated convolutions for multi-scale feature capture. MSDC [8] expands binary ON/OFF states into diverse states, enhancing power signal inference with richer insights. Additionally, attention mechanisms, known for their parallel efficiency in sequence processing and their success in natural language processing, have also

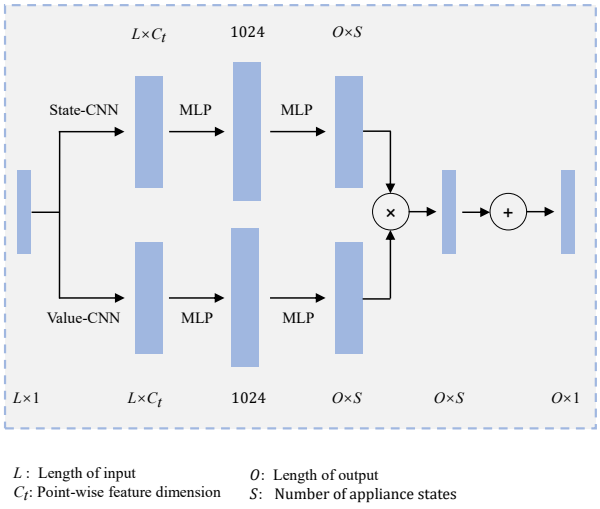


Figure 3: The architecture of the Seq2Seq model.

been explored for NILM [20, 26]. Furthermore, other literature has explored the application of Denoising AutoEncoders (DAE) [5], Generative Adversarial Networks (GAN) [19] and Hybrid Loss-Driven Multi-source Domain Adversarial Network (HLD-MDAN)[2] in the context of NILM.

Deep learning-based NILM methods can be broadly labelled as sequence-to-sequence learning (Seq2Seq), where input sequence are mapped to output sequence. As demonstrated in Figure 3, the Seq2Seq model employs a Value-CNN to extract the appliance power consumption features from the input sequence, while a State-CNN is utilized to capture the probability distribution of the appliance operating states. The integration of these two components is then used to generate the final prediction of the current appliance power consumption. In particular, the Seq2Point model [28] is entirely dependent on the Value-CNN. The SGN model [23] incorporates both a Value-CNN and a binary ON/OFF State-CNN, whereas the MSDC model [8] adopts a Value-CNN in conjunction with a multi-state State-CNN. It should be noted that the length of the output sequence can be shorter than the input sequence (referred to as Seq2Subseq), or even just one time step (referred to as Seq2Point). In this study, we collectively refer to these methods as Seq2Seq. The Seq2Seq model performs fine-grained processing on short sequences, treating each time sample as equally important, while neglecting the importance of global information over a long span.

Although transformer-based long sequence processing models such as Big Bird [27], Informer [30, 31], and Pyraformer [15] have shown some promise, they do not fully meet our expectations. Firstly, they still pose a considerable computational burden for resource-constrained smart meters, despite reducing the overhead compared to the most primitive Transformer. Secondly, some complex operators, such as sparse attention, are less computationally efficient on device hardware. Finally, with respect to task type, these are causal models originally designed for text generation or time series forecasting and thus cannot be applied directly to NILM tasks. Sepformer [24] is a speech separation model that achieves superior performance by grouping inputs into chunks and alternating intra-chunk and inter-chunk processing. Its improved version, RE-Sepformer [4], is resource efficient and can process long sequences with lower computational effort. We included RE-Sepformer as one of the baseline models in our experiments.

Further exploration is required for efficient processing of energy consumption data on resource-constrained smart meters. We argue that the input sequence contains a significant amount of redundant information and that the fine-grained feature extraction approach used by the Seq2Seq model is suboptimal. Therefore, we propose a shift towards performing coarse-grained feature extraction at the segment level, rather than fine-grained extraction at the time sample level. This change has the advantage of maintaining a low computational overhead while extending the acceptable length of input sequences, thereby enabling long-time contextual awareness.

3 Methodology

3.1 Problem formulation

Given the aggregated power consumption $\mathbf{X} = (x_1, x_2, \ldots, x_T)$, $x_t \in \mathbb{R}^+$, the goal of energy disaggregation is to recover the energy signals of I appliances of interest $\mathbf{Y}^i = (y_1^i, y_2^i, \ldots, y_T^i)$, where T is the measurement duration. Let $\mathbf{U} = (u_1, u_2, \ldots, u_T)$ denote the power consumption of other appliances. At each time step, the aggregated power consumption can be formulated as

$$x_t = \sum_{i=1}^{I} y_t^i + u_t + \epsilon_t, \tag{1}$$

where ϵ_t is the noise term.

In practice, a sliding window will be used to limit the length of input sequence. Specifically, denoting the input and output sequences as $\mathbf{X}_{t,L} \coloneqq (x_t, x_{t+1}, \ldots, x_{t+L-1})$ and $\mathbf{Y}_{t,L,i} \coloneqq (y_t^i, y_{t+1}^i, \ldots, y_{t+L-1}^i)$, both starting at t with length L. Note that to avoid losing contextual information at boundary points, the output sequence may correspond to the center subsequence of the sliding window (usually center-aligned). The full form of the output is

$$\mathbf{Y}_{t,L,O,i} \coloneqq (y^i_{t+\lfloor \frac{L-O}{2} \rfloor}, y^i_{t+\lfloor \frac{L-O}{2} \rfloor+1}, \ldots, y^i_{t+\lfloor \frac{L-O}{2} \rfloor+O}), \tag{2}$$

where O is the length of the output.

3.2 Sequence-to-Segments-to-Sequence Learning

In order to efficiently handle longer input sequences, we propose a new scheme called Sequence-to-Segments-to-Sequence (Seq2Seg2Seq). This scheme divides the input sequence into multiple non-overlapping segments. These segments then undergo intra-segment feature extraction and inter-segment feature interactions before being decoded back to the target sequence, which has the same length as the input. Within each segment, there are only a small number of representative points (e.g., power state transitions) and the rest points are less informative. Therefore, in order to improve the model efficiency, we compress multiple time sample features into a single segment feature, which significantly reduces the sequence length that the model needs to deal with. In order to minimize the number of parameters, we employ a shared multi-layer perceptron (MLP)

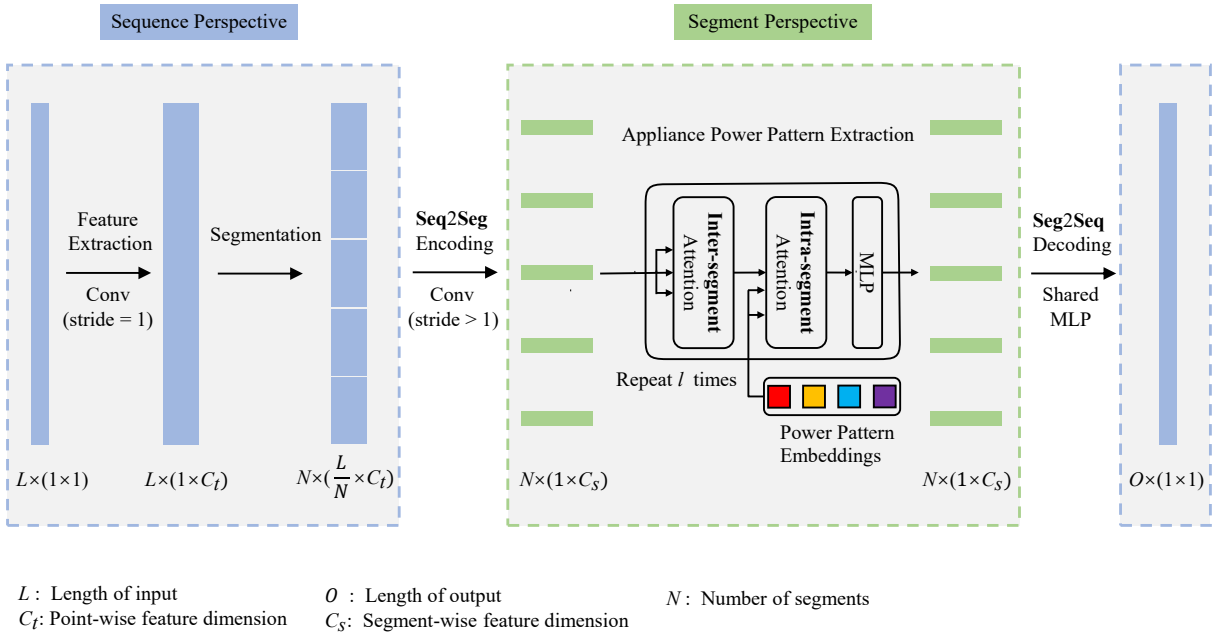


Figure 4: The architecture of the Seq2Seg2Seq model. First, the input sequence is convolved for feature extraction. Second, the sequence is divided and encoded into segments. Then, the segments are processed by intra-segment and inter-segment feature extraction. Finally, the segment features are decoded into the output. In our work, the input length L and output length O are equal.

for feature decoding, which predicts the power consumption for the appliance.

Figure 4 illustrates the architecture of Seq2Seg2Seq. Firstly, the input sequence of length S undergoes feature upscaling via convolutional layers with a stride of 1. Subsequently, the input sequence is divided into N segments, each of length M. The M time samples are compressed into one segment representation, facilitated by convolutional layers with a stride greater than 1. Immediately after that, we employ the Appliance Power Pattern Extraction module (described in the next subsection) to extract the global information of the N segments as well as the shape features within each segment. The segmentation process reduces the sequence length of the feature extraction to $1/M$ of the original length. Finally, for each segment, a shared MLP is utilized for feature decoding, which gives us the prediction results.

3.3 Appliance Power Pattern Extraction

Figure 2 shows some snippets of individual appliances' power consumption. Different appliances exhibit distinct power consumption profiles, which include characteristic features for appliance identification. These features include variations in power levels upon activation or deactivation, duration of active states, and the stability or fluctuation of power levels. The most prominent pattern in power consumption involves an initial increase upon appliance activation, followed by stable or oscillating power levels during the appliance's active state, and finally a decline upon appliance deactivation. After the segmentation of the input sequence, the activation periods of appliances are separated into distinct segments, each characterized by different power consumption patterns.

To capture both intra-segment patterns and global information across multiple segments, we introduce the Appliance Power Pattern Extraction (APPE) module, utilizing scaled dot-product attention mechanism [25]. The APPE module consists a stack of l APPE layers, each containing inter-segment multi-head attention, intra-segment multi-head attention, and a position-wise feed-forward layer.

Inter-segment multi-head attention is designed for capturing dependencies between different segments and achieving a global receptive field.

Multi-head Attention mechanism consists of h single-head scaled dot-product attentions, with each head accepts three input sequences denoted as $q \in \mathbb{R}^{l_q \times d_q}, k \in \mathbb{R}^{l_k \times d_k}, v \in \mathbb{R}^{l_v \times d_v}$, where l_q, l_k, l_v are the sequence lengths (l_k and l_v are required to be identical), d_q, d_k, d_v are the feature dimensions. The input q, k, and v for inter-segment multi-head attention are sourced from the output of the previous layer, constituting what is also known as multi-head self-attention. In individual single head, the input sequence are first linearly transformed by matrix multiplication to project them into the same feature space, yielding query (Q), key (K), value (V) representations: $Q = qW^q, K = kW^k, V = vW^v$, where $W^q \in \mathbb{R}^{d_q \times d_{model}}, W^k \in \mathbb{R}^{d_k \times d_{model}}, W^v \in \mathbb{R}^{d_v \times d_{model}}$, d_{model} is hidden size. The attention scores are generated by performing

softmax operation on the scaled QK^T multiplication result, where the scaling factor is the square root of the hidden size d_{model}. The output of the single-head scaled dot-product attention (Attention) is a weighted matrix obtained by multiplying the attention scores and value (V):

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_{model}}})V \tag{3}$$

In the multi-head attention, each head possesses its own distinct attention, capable of extracting information from individual subspaces. These outputs are subsequently concatenated and transformed to form the final multi-head attention output:

$$\text{MultiHead}(q, k, v) = \text{Concat}(head_1, \ldots, head_h)W^{Out} \tag{4}$$

where $head_i = \text{Attention}(qW_i^q, kW_i^k, vW_i^v)$, $i \in (1, ..., h)$, $W^{Out} \in \mathbb{R}^{d_{model} \times d_{model}}$.

The inter-segment multi-head attention captures dependencies between different segments, enabling the identification of dependencies across segments and the convergence of global information.

Intra-segment multi-head attention focuses on enhancing power pattern features within each segment. It is similar to the inter-segment multi-head attention mechanism, while inter-segment attention concentrates on the relationship between segments. Specifically, the intra-segment multi-head attention focuses on the correlation between the segment and power pattern embeddings. Power pattern embeddings constitutes a set of learnable latent features that encode the appliance power patterns, denoted as $ppe \in \mathbb{R}^{n_e \times d_e}$, where n_e is the number of patterns and d_e is the dimension of pattern features. For each segment, the weights of enhanced features are based on their correlations with appliance power patterns. As depicted in Figure 4, the input q for intra-segment multi-head attention comes from the previous layer's output (segment features), while k and v come from the power pattern embeddings ppe.

Position-wise Feed-Forward Network (FFN) is employed to apply a nonlinear transformation on the output of intra-segment multi-head attention. It consists of two linear layers with a ReLU activation and a dropout layer in between:

$$\text{FFN}(x) = \text{Linear}(\text{Dropout}(\text{ReLU}(\text{Linear}(x)))) \tag{5}$$

Residual Connections and Layer Normalization. After each attention layer and the feed-forward layer, residual connections are employed to retain input features, and dropout regularization is applied to enhance robustness. Additionally, layer normalization (LayerNorm) is applied to stabilize features across different layers. This operation can be formulated as $\text{LayerNorm}(x + \text{Dropout}(\text{PreLayer}(x)))$.

3.4 Batch Normalization and Inverse

In line with previous NILM literature [8, 28], we assume that the appliance power consumption follows a normal distribution. Nevertheless, in contrast to their approach of data preprocessing (z-score normalization), we use batch normalization (BN) and its inverse process to achieve end-to-end training. Batch normalization is applied to the input signal, while its inverse process is utilized for the final output of the network. The formal expressions for these two processes are as follows:

$$BN(x) = \gamma \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta, \tag{6}$$

$$BN_{inv}(x) = \frac{(x - \beta')}{\gamma'} \sqrt{\sigma'^2} + \mu', \tag{7}$$

where μ and σ are the mean and standard deviation of the mini-batch input, μ' and σ' are the exponential moving average of the mean and standard deviation of the ground truth obtained during the training period, γ, γ', β and β' are learn-able parameters that allow the network to scale and shift the normalized values, ϵ is a small constant added for numerical stability to avoid division by zero.

4 Experiments

In this paper, we have selected two real-world datasets, UK-DALE[1] [11] and REDD [13], as the benchmark datasets. We compare our model's performance with Seq2Point [28], SGN [23], BERT4NILM [26], and MSDC [8]. Since current NILM models exhibit limitations in processing long sequences effectively, we additionally chose to adopt the long sequence processing model RE-SepFormer[4] as our baseline model. This model is commonly used in speech separation tasks.

4.1 Datasets and Baselines

The UK-DALE and REDD datasets measure appliance-level and whole-home energy consumption for five UK houses from November 2012 to April 2017 and six US houses from April 2011 to June 2011, respectively. The aggregated power channel readings and appliance power channel readings for the UK-DALE data were recorded every 6s, while for REDD they were recorded at 1s and 3s, respectively. Similar to previous literature, we conducted experiments on kettle (only available in UK-DALE), refrigerator, washing machine, microwave, and dishwasher.

Households differ in appliance brands, wiring configurations, and usage behaviours, which introduces significant intra-dataset diversity. For instance, refrigerators exhibit different standby and defrost cycles, dishwashers are operated at varying times of day depending on user routines, and variations in household wiring further affect aggregate load signals. This inter-household diversity provides a realistic and challenging environment for evaluating model robustness. To evaluate the model's generalization capability, we used data from house 2 of UK-DALE and house 1 of REDD as unseen houses (test split). For the remaining houses, the data were divided into training and validation sets, with the first 80% used for training and the last 20% used for validation. Notably, the two datasets embody distinct regional electrical standards (e.g., grid frequency and nominal voltage levels). Such system-level discrepancies directly influence both aggregate and appliance-level power waveforms, suggesting that cross-dataset evaluation would primarily capture electrical mismatches rather than the intrinsic generalization capability of the model. Consequently, in accordance with previous NILM research, we abstain from conducting cross-dataset testing.

[1]The April 2017 version of UK-DALE

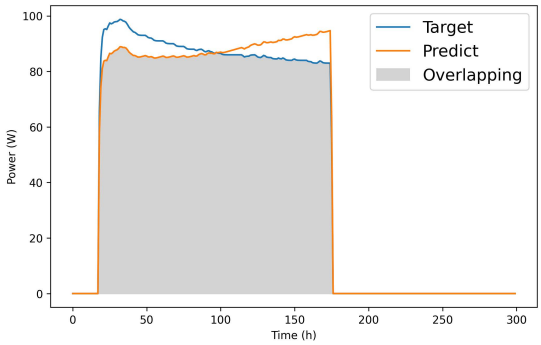


Figure 5: Illustration of overlap energy (gray).

In terms of data preprocessing, we adopted a simple approach that aligns the timestamps of the aggregated power channel and the appliance power channel, re-samples every 6 seconds, and backward-fills missing values. The maximum number of consecutive missing values for backward padding was set to 1, consistent with Zhang et al. [28].

We conducted a comparative analysis of our proposed model against five baseline models: **(1) Seq2Point**[28], a single-state CNN-based architecture that takes sequential input and produces point output. **(2) SGN**[23], a dual-CNN model designed to capture the ON/OFF states of appliances. **(3) BERT4NILM**[26], an architecture leveraging bidirectional encoder representations from transformers (BERT). **(4) MSDC**[8], a Multi-State Dual CNN model focusing on extracting information pertaining to the multiple states and state transitions of the appliance. **(5) RE-SepFormer**[4], a resource efficient enhanced version of the transformer-based speech separation model Sepformer [24], which adopts a multiscale approach to learn both short- and long-term dependencies.

4.2 Evaluation Metrics

Let y_t^i and $\hat{y}_t^i$ be the target and the predicted power for appliance i at time step t. We use the following three metrics as performance indicators:

(1) **MAE** (Mean Absolute Error) is a general metric used for regression problems to measure the prediction error at each point:

$$\text{MAE}^i = \frac{1}{T}\sum_{t=1}^{T}\left|y_t^i - \hat{y}_t^i\right|.$$

(2) SAE_δ (Signal Aggregate Error per δ period) represents the average total error in a sub-period δ of the total time, which compares the prediction sum and the target sum over the time period of δ [23]:

$$\text{SAE}_\delta^i = \frac{1}{T_\delta}\sum_{j=1}^{T_\delta}\frac{1}{N_\delta}\left|r_j^i - \hat{r}_j^i\right|,$$

where N_δ is the number of time steps in time period δ, $T_\delta = T/N_\delta$, $r_j^i = \sum_{t=1}^{N_\delta} y^i_{N_\delta j+t}$, and $\hat{r}_j^i = \sum_{t=1}^{N_\delta} \hat{y}^i_{N_\delta j+t}$. In our experiments, N_δ is 600, which corresponds to the number of data points in an hour.

(3) **Overlap Energy** is the common part of the predicted and the target energy [21]:

$$\text{Overlap energy}^i = \frac{\sum_{t=0}^{T}\min\{y_t^i, \hat{y}_t^i\}}{\sum_{t=0}^{T}\max\{y_t^i, \hat{y}_t^i\}} \times 100\%$$

MAE and SAE_δ are the most commonly used performance metrics in NILM research, showing the closeness of the predicted value to the ground truth. Smaller values of MAE and SAE_δ indicate better model performance. The overlap energy represents the percentage of correlation between the predicted and target energy, with higher values approaching 100% indicating better model performance. In cases where appliance data contains significantly fewer active cycles than inactive cycles, performance metrics normalized by time such as MAE are susceptible to a certain degree of dilution, while overlap energy is more stable in this regard.

4.3 Implementation Details

The deep learning models are implemented in Python using PyTorch, and trained on machines with NVIDIA GeForce RTX 4090. **Baseline models** were implemented and trained according to the descriptions in the original author's paper and open source code. We train model per appliance. For each experiment, we ran it more than 20 times independently and reported the average of the results. The confidence value is 0.95. The hyperparameters of RE-Sepformer are chosen as follows: The encoder and decoder both have a number of convolutional filters of 128, a size of 16, and a stride of 8. The number of sources is 1, the chunk size is 150, the IntraTransformer and InterTransformer both have 4 layers, each has 8 parallel attention heads, and 1024-dimensional positional feed-forward networks. The IntraTransformer and InterTransformer dual-path processing pipeline is repeated 2 times. This setup reduces the number of parameters in RE-Sepformer by half compared to the original paper, making it comparable to Seq2Seg2Seq.

Regarding the implementation details of the **Seq2Seg2Seq model**, the length of the segment is 525 and the number of segments is 12, hence the input sequence length is 6300. Convolutional layers with a stride of 1 are composed of four layers. The number of filters used is 32, 32, 64, 64, and the filter sizes are 13, 11, 9, 7. The convolutional layers with a stride greater than 1 are also composed of four layers. The number of filters used are 128, 128, 256, 256, the stride length are 7, 5, 5, 3, and the filter sizes are 9, 7, 7, 5. Following each convolutional layer, ReLU activation and batch normalization are applied. The Appliance Power Pattern Extraction module utilizes two TransformerDecoderLayers from the PyTorch library, each with a hidden size of 256, the number of heads is 4. The feed-forward hidden dimension is 1024, with a dropout rate of 0.1. The shared multilayer perceptron contains a hidden layer with a size of 1050. The output length is 6300, which is the same as the input.

In terms of **training details**, we use mean square error loss, and the optimizer is AdamW [16] with a weight decay of 10^{-2}. The batch size is 256, and maximum epochs is 20. A checkpoint is saved at the end of each epoch, and the model parameters of the checkpoint with the best MAE on the validation set will be retained

Metric	Methods	Kettle	Fridge	Dish washer	Micro-wave	Washing machine	Average	Average improvement
MAE	Seq2Point	12.25	27.47	32.31	10.49	11.76	18.86 ± 0.74	-
	SGN	12.31	27.24	19.42	8.94	16.96	16.97 ± 0.69	-
	BERT4NILM	14.23	25.01	20.72	6.21	**7.25**	14.68 ± 0.55	-
	RE-Sepformer	17.10	13.68	**14.65**	8.42	11.93	13.11 ± 0.27	-
	MSDC	10.23	20.07	19.65	9.36	11.90	14.24 ± 0.56	-
	Seq2Seg2Seq (our)	**8.96**	**13.34**	22.33	**6.16**	7.41	**11.64 ± 0.12**	18.26%
SAE_δ	Seq2Point	8.56	19.53	27.65	8.94	9.93	14.92 ± 0.67	-
	SGN	10.06	20.23	14.93	7.92	14.72	13.57 ± 0.66	-
	BERT4NILM	13.97	20.89	17.52	6.13	**5.73**	12.85 ± 0.59	-
	RE-Sepformer	15.55	8.19	**12.46**	7.00	10.89	10.82 ± 0.29	-
	MSDC	8.22	11.24	15.22	8.18	10.11	10.59 ± 0.58	-
	Seq2Seg2Seq (our)	**5.39**	**8.11**	15.32	**5.38**	5.83	**8.01 ± 0.18**	24.36%
Overlap energy	Seq2Point	0.634	0.455	0.376	0.062	0.253	0.356 ± 0.012	-
	SGN	0.630	0.454	0.634	0.075	0.237	0.406 ± 0.013	-
	BERT4NILM	0.519	0.481	0.537	0.056	**0.453**	0.409 ± 0.016	-
	RE-Sepformer	0.443	0.724	**0.670**	0.211	0.268	0.465 ± 0.011	-
	MSDC	0.676	0.557	0.634	0.085	0.260	0.442 ± 0.011	-
	Seq2Seg2Seq (our)	**0.715**	**0.730**	0.527	**0.239**	0.402	**0.523 ± 0.005**	18.33%

Table 1: Experimental results on the UK-DALE house 2 (unseen). The calculation of the average improvement is based on the state-of-the-art NILM model MSDC. Bold numbers indicate the best results.

Metric	Methods	Fridge	Dish washer	Micro-wave	Washing machine	Average	Average improvement
MAE	Seq2Point	35.99	20.43	21.62	14.59	23.16 ± 0.66	-
	SGN	33.61	20.44	**21.04**	13.01	22.02 ± 0.53	-
	BERT4NILM	35.11	25.25	21.96	26.14	27.12 ± 0.92	-
	RE-Sepformer	26.69	21.98	54.58	15.65	29.71 ± 1.25	-
	MSDC	33.89	**14.51**	21.16	**12.78**	20.58 ± 0.58	-
	Seq2Seg2Seq (our)	**22.93**	15.04	21.30	13.27	**18.13 ± 0.19**	11.90%
SAE_δ	Seq2Point	20.72	18.26	**15.72**	11.99	16.67 ± 0.58	-
	SGN	21.22	17.26	15.85	9.94	16.07 ± 0.50	-
	BERT4NILM	21.29	25.15	21.32	23.34	22.78 ± 1.06	-
	RE-Sepformer	17.15	21.33	46.61	12.52	24.39 ± 1.15	-
	MSDC	21.99	**12.08**	16.33	10.05	15.11 ± 0.58	-
	Seq2Seg2Seq (our)	**13.94**	13.94	16.98	**8.56**	**13.36 ± 0.21**	11.58%
Overlap energy	Seq2Point	0.471	0.363	0.285	0.611	0.432 ± 0.011	-
	SGN	0.523	0.362	0.266	0.666	0.454 ± 0.010	-
	BERT4NILM	0.504	0.016	0.030	0.240	0.198 ± 0.031	-
	RE-Sepformer	0.593	0.244	0.088	0.634	0.391 ± 0.009	-
	MSDC	0.494	**0.483**	**0.287**	0.669	0.483 ± 0.008	-
	Seq2Seg2Seq (our)	**0.654**	0.439	0.182	**0.688**	**0.491 ± 0.006**	1.66%

Table 2: Experimental results on the REDD house 1 (unseen). The calculation of the average improvement is based on the state-of-the-art NILM model MSDC. Bold numbers indicate the best results.

for the test. The learning rate is set to 5×10^{-4} using linear schedule with 5 epochs warm-up, i.e. the learning rate linearly increases from 0 to 5×10^{-4} over 5 epochs, and then linearly decreases to 0 over 15 epochs. The stride of the sliding window is 100. Both Seq2Seg2Seq and RE-Sepformer use this training strategy, while the other models use the approach in their papers and code.

4.4 Experimental results

Table 1 and Table 2 show the experimental results for the UK-DALE and REDD datasets, respectively. The results of the Seq2Seg2Seq model consistently exhibit superiority over all baseline models, demonstrating favorable performance on most appliances. On both the UK-DALE and REDD datasets, Seq2Seg2Seq achieves significant

Methods	Input length	Output length	Params (M)	Mem. (MB)	MACs (M)	Fwd latency (ms)
Seq2Point	599	1	30.71	244.6	52.94	38.830
SGN	200	32	20.65	168.0	1.29	0.856
BERT4NILM	480	480	**1.94**	**22.5**	515.79	0.113
MSDC	200	32	24.9	204.3	1.55	0.341
RE-Sepformer	6300	6300	3.98	125.9	0.62	0.096
Seq2Seg2Seq	**6300**	**6300**	3.74	29.4	**0.08**	**0.008**

Table 3: Comparison of model profiles. Params means the number of parameters; Mem. means the memory consumed by creating and running an ONNX session. MACs means the number of multiply-accumulate operations; Fwd latency means forward propagation latency. MACs and Fwd latency are normalized using the output length.

improvements over the best-performing NILM model, with reductions of 18.26% and 11.90% in overall MAE, and 24.36% and 11.58% in overall SAE_δ, respectively. Additionally, overall overlap energy comparisons exhibit increments of 18.33% and 1.66%, respectively. In particular, Seq2Seg2Seq exhibits excellent performance in predicting the fridge's power consumption, manifesting gains exceeding 30% across all metrics on both datasets compared to the second-best NILM model. This significant improvement is attributed to fridge's continuously-on feature, together with the comparatively regular shape of its power profile. In contrast, the power consumption data of the washing machine are characterized by increased complexity and flunctuations during its active phases, which makes the model's approximation of the washing machine's power profile less accurate. A comparative analysis of the experimental results between the two datasets shows the superior performance of UK-DALE over REDD, which is in line with expectations, as dataset size is widely recognized as a crucial determinant of the performance of deep neural networks, as UK-DALE is a significantly larger dataset than REDD.

The performance of RE-Sepformer is unstable. For the dishwasher data in the UK-DALE dataset, RE-Sepformer outperforms all other models. Furthermore, RE-Sepformer marginally outperforms MSDC in terms of average MAE on the UK-DALE dataset. However, its performance significantly decreases when applied to the REDD dataset, particularly in the case of microwave data. After reviewing the training log of RE-Sepformer on the REDD dataset for microwave data, it was observed that the model performed well during both the training and validation phases, but yielded numerous false positives during the test phase. Therefore, despite its proficiency in language separation tasks, RE-Sepformer is not well-suited for direct application to NILM tasks.

To investigate the real-world performance of the model in deployment scenarios, we measured the profiles of our model and the baseline model using the open-source library Deepspeed. Specifically, for the measurements of forward propagation latency and memory consumption, we first converted the models to ONNX format, and then ran them on a Raspberry Pi 4B using the InferenceSession of onnxruntime library. To measure forward propagation latency, we utilize the time module from the Python library to record the ONNX session's runtime. We include a warm-up time of 10 and a repetition count of 20. To measure memory consumption, we use the memory_profiler library to record the memory consumed by creating and running an ONNX session. The results are depicted in Table 3.

For the number of parameters, BERT4NILM, RE-Sepformer and Seq2Seg2Seq are in the same order of magnitude, with fewer parameters than Seq2Point, SGN and MSDC. Multiply-accumulate operations (MACs) and forward propagation latency (Fwd latency) are normalized using the output length. Notably, Seq2Seg2Seq has significantly lower MACs and Fwd latency compared to the baseline models. This indicates that Seq2Seg2Seq has a distinct advantage in terms of computational efficiency and low latency. In resource-constrained smart meters, it is equally important to have a small memory footprint and low computational resource consumption as it is to have high accuracy.

4.5 Ablation study

Methods	avg MAE	avg SAE_δ	avg Overlap
S^3-N6	12.22 ± 0.16	8.51 ± 0.24	0.507 ± 0.006
S^3-N18	**11.55 ± 0.11**	8.03 ± 0.15	0.521 ± 0.004
S^3-M175	11.67 ± 0.25	8.62 ± 0.31	0.519 ± 0.009
S^3-M875	12.07 ± 0.30	8.67 ± 0.41	0.506 ± 0.009
S^3-SA	12.11 ± 0.16	8.64 ± 0.21	0.510 ± 0.006
S^3	11.64 ± 0.12	**8.01 ± 0.18**	**0.523 ± 0.005**

Table 4: The ablation study results on the UK-DALE house 2. Based on S^3 (Seq2Seg2Seq, N = 12, M = 525), S^3-N6, S^3-N18 using 6 and 18 segments, respectively, S^3-M175, S^3-M875 using segment lengths of 175 and 875, respectively, and S^3-SA changed the APPE to self attention only.

To investigate the effectiveness of the APPE module and the effect of the number of segments, we constructed ablation experiments. Based on the Seq2Seg2Seq, S^3-N6 and S^3-N18 are obtained by changing the number of segments from 12 to 6 and 18, respectively, S^3-M175 and S^3-M875 are obtained by changing the segment length from 525 to 175 and 875, respectively, and S^3-SA is obtained by replacing the APPE module with a self-attention module (transformer encoder).

The results of the ablation experiment are shown in Table 4. Seq2Seg2Seq outperforms S^3-SA proving the effectiveness of the APPE module, which is able to learn the target appliance power patterns and enhance the associated features. Compared to Seq2Seg2Seq, S^3-N6 shows a 5% degradation in MAE performance and a 6% degradation in SAE_δ performance, indicating that the input sequence length is insufficient. However, S^3-N18 only leads to less than 1%

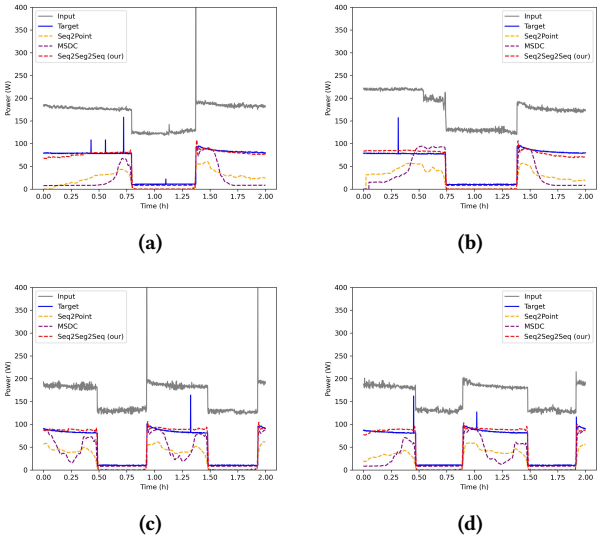


Figure 6: Seq2Seq model prediction fails when the power remains stable in the sliding window. The sliding window lengths for Seq2Point, MSDC, and Seq2Seg2Seq are 1, 0.667, and 10.5 hours, respectively.

improvement in MAE performance and weaker SAE_δ and overlap energy metrics compared to Seq2Seg2Seq. Despite this, it does result in an increase in computational complexity. It demonstrates that selecting 12 as the number of segments N is reasonable. Both S^3-M175 and S^3-M875 exhibit a decrease in performance compared to Seq2Seg2Seq.

4.6 Result Visualization

We give a visualization of the model predictions. Figure 6 illustrates a snipet of fridge data from House 2 in the UK-DALE dataset, along with the corresponding predictions generated by the baseline and our models. The Seq2Seq model, which uses relatively shorter sliding windows, encounters challenges in accurately detecting appliance usage when the input signal remains stable within the window, such as during the first half hour in Figure 6. This issue arises due to the lack of global information within the input signal. Conversely, the Seq2Seg2Seq model, which is designed to accommodate long input sequences, successfully addresses this concern. This observation emphasizes the necessity of using long sliding windows.

5 Conclusions and Future work

In this paper, we introduce a segment-based approach called Sequence-to-Segments-to-Sequence (Seq2Seg2Seq). This approach aims to effectively manage long input signals. Complemented by the Appliance Power Pattern Extraction module, our solution enables the incorporation of long-time contextual information without significantly increasing computational overhead. A concrete Seq2Seg2Seq model has been developed and applied to real-world datasets. The experimental results demonstrate the superiority of our model over previous approaches in terms of accuracy, computational efficiency, and time consumption. Through visualizations, we illustrate how the Seq2Seg2Seq scheme addresses the issues encountered in earlier research, where short input sequences lack global information and lead to model failures. The shift from time sample granularity to segment-based analysis promises to yield a more robust and efficient framework for NILM, ultimately contributing to improved accuracy.

Future research could investigate the improvement of segment-based energy disaggregation techniques, including automatic identification of appliance-specific segment length from data or dynamic real-time determination of segment lengths from input signals.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62472452); in part by Guangdong Basic and Applied Basic Research Foundation (No. 2025A1515010262). Q. Huang is the corresponding author.

References

[1] Nipun Batra, Amarjeet Singh, and Kamin Whitehouse. 2016. Gemello: Creating a Detailed Energy Breakdown from Just the Monthly Electricity Bill. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (San Francisco, California, USA) *(KDD '16)*. 431–440. doi:10.1145/2939672.2939735

[2] Xiaomin Chang, Wei Li, Yunchuan Shi, and Albert Y. Zomaya. 2023. Taming the Domain Shift in Multi-source Learning for Energy Disaggregation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Long Beach, CA, USA) *(KDD '23)*. Association for Computing Machinery, New York, NY, USA, 3805–3816. doi:10.1145/3580305.3599910

[3] Kunjin Chen, Yu Zhang, Qin Wang, Jun Hu, Hang Fan, and Jinliang He. 2020. Scale- and Context-Aware Convolutional Non-Intrusive Load Monitoring. *IEEE Transactions on Power Systems* (May 2020), 2362–2373. doi:10.1109/tpwrs.2019.2953225

[4] Luca Della Libera, Cem Subakan, Mirco Ravanelli, Samuele Cornell, Frédéric Lepoutre, and François Grondin. 2024. Resource-efficient separation transformer. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 761–765.

[5] Felan Carlo C. Garcia, Christine May C. Creayla, and Erees Queen B. Macabebe. 2017. Development of an Intelligent System for Smart Home Energy Disaggregation Using Stacked Denoising Autoencoders. *Procedia Computer Science* 105 (Jan 2017), 248–255. doi:10.1016/j.procs.2017.01.218

[6] George William Hart. 1992. Nonintrusive appliance load monitoring. *Proc. IEEE* 80, 12 (1992), 1870–1891.

[7] Taha Hassan, Fahad Javed, and Naveed Arshad. 2014. An Empirical Investigation of V-I Trajectory based Load Signatures for Non-Intrusive Load Monitoring. *IEEE Transactions on Smart Grid* 5, 2 (Mar 2014), 870–878. doi:10.1109/tsg.2013.2271282

[8] Jialing He, Jiamou Liu, Zijian Zhang, Yang Chen, Yiwei Liu, Bakh Khoussainov, and Liehuang Zhu. 2023. MSDC: Exploiting Multi-State Power Consumption in Non-intrusive Load Monitoring Based on a Dual-CNN Model. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 4 (Jun. 2023), 5078–5086. doi:10.1609/aaai.v37i4.25636

[9] Maria Kaselimi, Nikolaos Doulamis, Athanasios Voulodimos, Eftychios Protopapadakis, and Anastasios Doulamis. 2020. Context Aware Energy Disaggregation Using Adaptive Bidirectional LSTM Models. *IEEE Transactions on Smart Grid* (Jul 2020), 3054–3067. doi:10.1109/tsg.2020.2974347

[10] Jack Kelly and William Knottenbelt. 2015. Neural NILM: Deep Neural Networks Applied to Energy Disaggregation. In *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*. doi:10.1145/2821650.2821672

[11] Jack Kelly and William Knottenbelt. 2015. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. *Scientific Data* 2, 1 (Mar 2015). doi:10.1038/sdata.2015.7

[12] J.Zico Kolter and TommiS. Jaakkola. 2012. Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation. *International Conference on Artificial Intelligence and Statistics,International Conference on Artificial Intelligence and Statistics* (Mar 2012).

[13] J Zico Kolter and Matthew J Johnson. 2011. REDD: A public data set for energy disaggregation research. In *Workshop on data mining applications in sustainability (SIGKDD), San Diego, CA*.

[14] Dasheng Lee and Chin-Chi Cheng. 2016. Energy savings by energy management systems: A review. *Renewable and Sustainable Energy Reviews* 56 (Apr 2016), 760–777. doi:10.1016/j.rser.2015.11.067

[15] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. 2022. Pyraformer: Low-Complexity Pyramidal Attention for Long-Range Time Series Modeling and Forecasting. In *International Conference on Learning Representations*.

[16] Ilya Loshchilov and Frank Hutter. 2017. Fixing Weight Decay Regularization in Adam. *ArXiv* abs/1711.05101 (2017).

[17] Lukas Mauch and Bin Yang. 2016. A novel DNN-HMM-based approach for extracting single loads from aggregate power signals. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. doi:10.1109/icassp.2016.7472104

[18] David Murray, Lina Stankovic, Vladimir Stankovic, Srdjan Lulic, and Srdjan Sladojevic. 2019. Transferability of Neural Network Approaches for Low-rate Energy Disaggregation. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. doi:10.1109/icassp.2019.8682486

[19] Yungang Pan, Ke Liu, Zhaoyan Shen, Xiaojun Cai, and Zhiping Jia. 2020. Sequence-To-Subsequence Learning With Conditional Gan For Power Disaggregation. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. doi:10.1109/icassp40776.2020.9053947

[20] Veronica Piccialli and Antonio M. Sudoso. 2021. Improving Non-Intrusive Load Disaggregation through an Attention-Based Deep Neural Network. *Energies* (Feb 2021), 847. doi:10.3390/en14040847

[21] Hasan Rafiq, Xiaohan Shi, Hengxu Zhang, Huimin Li, Manesh Kumar Ochani, and Aamer Abbas Shah. 2021. Generalizability Improvement of Deep Learning-Based Non-Intrusive Load Monitoring System Using Data Augmentation. *IEEE Transactions on Smart Grid* (Jul 2021), 3265–3277. doi:10.1109/tsg.2021.3082622

[22] Pascal A. Schirmer and Iosif Mporas. 2023. Non-Intrusive Load Monitoring: A Review. *IEEE Transactions on Smart Grid* 14, 1 (Jan 2023), 769–784. doi:10.1109/tsg.2022.3189598

[23] Changho Shin, Sunghwan Joo, Jaeryun Yim, Hyoseop Lee, Taesup Moon, and Wonjong Rhee. 2019. Subtask Gated Networks for Non-Intrusive Load Monitoring. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (Jul. 2019), 1150–1157. doi:10.1609/aaai.v33i01.33011150

[24] Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong. 2021. Attention Is All You Need In Speech Separation. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 21–25. doi:10.1109/ICASSP39728.2021.9413901

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6000–6010.

[26] Zhenrui Yue, Camilo Requena Witzig, Daniel Jorde, and Hans-Arno Jacobsen. 2020. BERT4NILM: A Bidirectional Transformer Model for Non-Intrusive Load Monitoring. In *Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring*. 89–93. doi:10.1145/3427771.3429390

[27] Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big Bird: Transformers for Longer Sequences. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS'20)*. Curran Associates Inc., Red Hook, NY, USA, Article 1450, 15 pages.

[28] Chaoyun Zhang, Mingjun Zhong, Zongzuo Wang, Nigel Goddard, and Charles Sutton. 2018. Sequence-to-Point Learning with Neural Networks for Non-Intrusive Load Monitoring. *Proceedings of the AAAI Conference on Artificial Intelligence* 32, Article 318 (Feb. 2018), 8 pages.

[29] Bochao Zhao, Lina Stankovic, and Vladimir Stankovic. 2015. Blind non-intrusive appliance load monitoring using graph-based signal processing. In *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. 68–72. doi:10.1109/GlobalSIP.2015.7418158

[30] Haoyi Zhou, Jianxin Li, Shanghang Zhang, Shuai Zhang, Mengyi Yan, and Hui Xiong. 2023. Expanding the prediction capacity in long sequence time-series forecasting. *Artificial Intelligence* 318, C (may 2023), 29 pages. doi:10.1016/j.artint.2023.103886

[31] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference*, Vol. 35. AAAI Press, 11106–11115.