energies

MDPI

Article

Non-Intrusive Load Monitoring at Smart Meter-Compatible Resolution: A Multi-Household Evaluation of Deep Learning Models

Vasileios Laitsos [1,2], Filippos Kolimenos [1] and Eleftherios O. Kontis [1,*]

[1] Smart Energy Policy and Networks Laboratory, Department of Electrical and Computer Engineering, University of Thessaly, 38334 Volos, Greece; vlaitsos@uth.gr (V.L.); filipposkolimenos13@gmail.com (F.K.)
[2] Hellenic Electricity Distribution Network Operator S.A. (HEDNO), 11743 Athens, Greece
* Correspondence: elkontis@uth.gr

Abstract

Non-intrusive load monitoring (NILM) estimates the consumption of individual appliances from a single aggregate measurement and is a natural companion to smart meter roll-outs. Most deep learning NILM studies, however, report results on a handful of houses and a few well-behaved appliances, which makes it hard to judge how consistently an architecture performs across households with different appliance mixes. This paper evaluates five models of increasing complexity, namely a linear regression baseline, a multilayer perceptron (MLP), a sequence-to-point convolutional neural network (CNN), a long short-term memory (LSTM) network and a Transformer encoder, on fifteen houses of the REFIT dataset resampled to 30 s. Each house is trained and tested separately. All houses are processed with the same pipeline: median resampling, chronological splitting, 199-sample input windows, a multi-output formulation with up to four target appliances per house and a per-appliance weighted loss that counteracts the scarcity of ON samples. The CNN obtained the highest average F1-score in eleven of the fifteen houses (mean 0.675), whereas a compact Transformer encoder with 71 k parameters was best in the remaining four (mean 0.598). The LSTM and the Transformer were the least stable models, falling below the linear baseline in four and two houses, respectively. Appliance-level results show that, in the houses studied, detection difficulty depends more on how distinct the power signature is and how close the operating level lies to the ON threshold than on how rarely the appliance is used. Detection and energy estimation also diverged in several houses, which argues against single-metric reporting.

Keywords: class imbalance; convolutional neural networks; deep learning; energy disaggregation; LSTM; non-intrusive load monitoring; REFIT; sequence-to-point; smart meters

Check for updates

Academic Editors: Amedeo Buonanno, Salvatore Fabozzi, Maria Valenti and Giorgio Graditi

Received: 5 October 2026
Revised: 18 November 2026
Accepted: 19 November 2026
Published: 24 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Electricity networks are moving away from one-directional supply towards systems with distributed generation, flexible demand and two-way communication with the customer [1]. In this setting, knowing what happens behind the meter is not a curiosity. Appliance-level information supports energy feedback to households, demand response, load forecasting and the detection of faulty equipment. Measuring every appliance with its own sensor is, however, expensive, intrusive and hard to maintain at scale.

For distribution system operators, the value of disaggregated information lies mainly in aggregation. Knowing which share of residential demand comes from thermostatically

controlled loads, wet appliances or electric heating helps to estimate flexibility potential, to design demand response programmes and to interpret changes in feeder load profiles. Smart meters already record whole-house consumption at intervals ranging from seconds on the customer interface to 15 min or more in the billing channel. Any method that extracts appliance-level information from these existing measurements, without extra hardware in the home, is therefore of direct practical interest.

Non-intrusive load monitoring (NILM), introduced by Hart [2], offers an alternative. A single meter at the point of supply records the total demand, and an algorithm infers which appliances are running and how much each one consumes. Early approaches relied on event detection and hand-crafted steady-state features, followed by combinatorial optimisation or hidden Markov models [3]. These methods work reasonably when few loads are present, but they degrade as the number of appliances grows and as multi-state devices overlap in time.

The availability of public datasets such as REDD [4], UK-DALE [5] and REFIT [6] made supervised learning possible, and neural approaches have dominated the field since the work of Kelly and Knottenbelt [7]. The sequence-to-point (seq2point) formulation of Zhang et al. [8], in which a network sees a window of the aggregate signal and predicts the appliance power at its midpoint, has become a common reference. Since then, researchers have proposed recurrent, attention-based and hybrid architectures, often with considerable gains on selected benchmarks [9,10].

Two practical issues remain less explored. First, most studies evaluate on one to five houses and on a short list of appliances, typically the fridge, kettle, washing machine, dishwasher and microwave. Performance reported on such subsets says little about how a model behaves across households with different appliance mixes, data gaps and usage patterns. Bousbiat et al. [11] have shown that the ranking of neural NILM models can change with hyperparameters and experimental protocol, which reinforces the need for broader comparisons under a fixed pipeline. Second, the sampling rate matters. High-frequency methods achieve excellent accuracy but need dedicated hardware, while very coarse data (15 min or more) hide the edges that make appliances recognisable [12]. Resolutions in the range of tens of seconds are compatible with many smart meter and home energy monitor interfaces and keep training costs moderate [13].

This paper addresses both issues with a systematic comparison on fifteen REFIT houses at 30 s resolution. The same preprocessing, windowing, loss function, training schedule and metrics are applied to every house, and only the set of target appliances changes. Models are trained and evaluated separately in each house. The study is therefore a multi-household evaluation under a fixed pipeline: it measures how consistently each architecture performs across heterogeneous households, not how well a model trained in one home transfers to another, which is discussed as future work in Section 6. Compared with most earlier studies, which evaluate on one to five houses and a few well-behaved appliances, the present work evaluates five architectures on fifteen households and 59 appliance instances under a single, fixed protocol, so that differences between models can be attributed to the architecture rather than to the experimental setup. The contributions are the following: (i) We compare five models of increasing capacity (linear regression, MLP, seq2point CNN, a convolutional LSTM and a Transformer encoder) in a multi-output setting where one network predicts up to four appliances at once. (ii) We use a per-appliance weighted mean squared error that raises the penalty on errors during ON periods, which is necessary for appliances that are active for less than 1% of the time. (iii) We report detection and estimation metrics side by side and show where they disagree. (iv) We analyse which appliance and data characteristics make disaggregation hard, using evidence from all fifteen houses rather than a single case. (v) We test the robustness of the model ranking to the ON thresholds and

class weights, and we quantify the memory and computational cost of each model with a view to deployment on metering hardware.

The rest of the paper is organised as follows: Section 2 reviews related work. Section 3 describes the dataset, preprocessing, models and evaluation protocol. Section 4 presents the results, Section 5 discusses their implications and the scope of the study, and Section 6 concludes the paper.

2. Related Work

Surveys of NILM classify household loads into on/off, finite-state, continuously variable and permanent consumers, and note that no single algorithm handles all four types well [3]. Classical probabilistic approaches are still in use. Hussein et al. [14] modelled appliances as multi-state hidden Markov models, combined them through Kronecker products and decoded the joint state with the Viterbi algorithm. Berrettoni et al. [15] examined combinatorial optimisation and found that its accuracy falls quickly with meter uncertainty and with the number of appliances: even with an ideal meter, accuracy dropped from 100% for three loads to 47% for seven. Unsupervised alternatives based on graph signal processing and spectral clustering have also been evaluated on REFIT with competitive event detection results [16].

A recurring difficulty is the overlap of appliance activity. Hart's original work assumed that only one appliance changes state at a time. Makonin [17] tested this switch continuity principle and found simultaneous switching events in about 20.8% of the time on average in modern homes, far more than originally assumed. Models trained without enough overlapping examples therefore tend to fail in realistic conditions.

Unsupervised and feature engineering approaches remain active as well. Jacobs and Henneaux [18] combined appliance-specific detectors for dominant loads, such as the fridge compressor cycle and the washing machine heating element, with a generic stage that pairs ON/OFF events through mixed integer programming and clusters the remaining events. At high sampling rates, voltage–current trajectories can be turned into images and classified with compact convolutional networks. Colour-coding the trajectory according to power-theoretic quantities has been reported to reach about 97.7% accuracy on PLAID with a much smaller model [19]. Such methods, however, require waveform data in the kHz range, which ordinary smart meters do not provide. To study overlap in a controlled way, Nour et al. [20] proposed an artificial aggregation of measured appliance currents, which exposed masking effects where a large load hides the switching of a small one.

Among neural approaches, the seq2point CNN [8] remains a strong baseline. Zhang et al. [10] added a bidirectional LSTM and an attention layer to a seq2point network and reported gains on multi-state appliances such as the dishwasher on UK-DALE. Kong et al. [9] combined CNN, BiLSTM and Transformer blocks with GELU activations and obtained large F1 improvements on REDD and UK-DALE, at a considerably higher computational cost. Comparative studies do not always favour deep models. Shabbir et al. [12] compared LSTM, XGBoost, logistic regression and DTW-KNN on a year of Estonian household data and found XGBoost to be the most accurate, with the LSTM penalised by the 15 min sampling interval.

The sampling rate has a direct effect on training cost. Osathanunkul and Osathanunkul [13] trained an LSTM for fridge disaggregation on UK-DALE at intervals from 6 to 60 s. Training time per epoch fell from about 637 s to 88 s, while F1 remained fairly stable and MAE began to deteriorate beyond roughly 42 s. This supports intermediate resolutions such as 30 s as a reasonable compromise.

Deployment has received increasing attention. Mari et al. [21] ran a seq2point CNN trained on REFIT on an Arm Cortex-M7 microcontroller in two Italian houses for a full year, with energy errors of 10–12% over six months. Lightweight attention networks with fewer

than 70 k parameters have been deployed on embedded GPUs [22], and checkpoint weight averaging has been proposed as a low-cost way to improve single-model predictions [23]. On the learning paradigm side, federated and decentralised training have been proposed to keep consumption data inside the home [24,25], and transfer learning has been used to reuse models across houses and datasets [26].

Table 1 summarises representative studies discussed in this section together with the setting of the present work.

Table 1. Summary of representative NILM studies and of the present work.

Ref.	Method	Data and Resolution	Main Finding
[14]	Multi-state HMM, Viterbi decoding	AMPds	91.65% accuracy on six appliances
[15]	Combinatorial optimisation	Simulated, EN 50470-1 meters	Accuracy falls from 100% (3 loads) to 47% (7 loads)
[8]	Seq2point CNN	UK-DALE, REDD	Midpoint prediction outperforms sequence-to-sequence baselines
[10]	CNN + BiLSTM + attention	UK-DALE	Lower MAE than LSTM, GRU and seq2point
[9]	CNN + BiLSTM + Transformer	REDD, UK-DALE	Large F1 gains over BERT and Seq2Seq
[12]	LSTM, XGBoost, LR, DTW-KNN	One Estonian house, 15 min	XGBoost most accurate; LSTM hurt by coarse sampling
[13]	LSTM	UK-DALE, 6–60 s	Epoch time 637 → 88 s; MAE degrades beyond ~42 s
[16]	Spectral clustering (unsupervised)	REFIT	Competitive event detection at low cost
[21]	Seq2point CNN on Cortex-M7	Trained on REFIT; 2 houses, 12 months	Energy error of 10–12% over six months
[22]	Lightweight attention network	UK-DALE, LIT, CEA	Fewer than 70 k parameters on embedded GPU
This work	Linear, MLP, CNN, LSTM, Transformer (multi-output)	REFIT, 15 houses, 30 s	CNN best in 11 of 15 houses, Transformer in 4; LSTM least stable

Two gaps motivate the present work. Evaluations across many houses under a single, fixed protocol are rare, and so are direct checks of whether added architectural complexity pays off against simpler models trained on properly preprocessed data. In addition, rare high-power appliances such as kettles and washing machines are easily swamped by the zero-dominated targets when a plain MSE loss is used, which leads to low recall for exactly the loads that matter most for demand response.

3. Materials and Methods

Figure 1 summarises the processing chain. Each house is handled independently, with the same settings, so that differences in the results can be traced to the house and its appliances rather than to the pipeline.

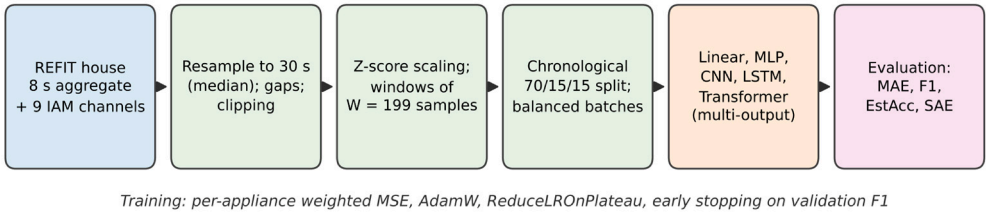


Figure 1. Processing and evaluation pipeline applied identically to each of the fifteen REFIT houses.

3.1. Problem Formulation

Let $y_t \geq 0$ be the aggregate active power measured at time t and $x_t^{(i)} \geq 0$ the power of appliance i, with $i = 1, \ldots, M$. The aggregate is modelled as

$$y_t = \sum_{i=1}^{M} x_t^{(i)} + \varepsilon_t \tag{1}$$

where ε_t collects measurement noise and all loads that are not modelled. Following the seq2point idea [8], a network f_θ receives a window of W aggregate samples centred on t and estimates the power of all target appliances at the midpoint:

$$\hat{\mathbf{x}}_t = f_\theta\left(y_{t-\lfloor W/2 \rfloor}, \cdots, y_{t+\lfloor W/2 \rfloor}\right) \tag{2}$$

Unlike many seq2point implementations that train one network per appliance, a single multi-output network is trained per house with $M = 4$ outputs (three in one house, see Section 3.2). This reduces the training cost by roughly a factor of four and allows for the network to share features between loads that co-occur.

3.2. Dataset and Appliance Selection

The REFIT Electrical Load Measurements dataset [6] was collected in 20 households around Loughborough, UK, between 2013 and 2015. Each house has a whole-house CurrentCost meter and nine individual appliance monitors (IAMs), all sampled nominally every 8 s. The occupants lived normally throughout the campaign, so the data contain real usage habits, seasonal effects and the usual recording gaps.

For each house, four of the nine monitored appliances were selected as targets using three rules: an ON fraction of at least about 1% in the training period, a power signature that is distinguishable from the background, and no pathological behaviour such as a device that is almost permanently on. Preliminary runs with all nine channels showed that very rare appliances mainly added variance, while always-on devices inflated the averaged metrics without testing the model. In REFIT House 11 only three appliances were retained, because the dishwasher and microwave had no activations at all in the test period and their metrics would have been undefined. Table 2 lists the fifteen houses, their target appliances and the fraction of rows lost during cleaning. Houses are referred to by their original REFIT numbers throughout the paper. The complete configuration of each house, including the ON thresholds and class weights of every target appliance, is listed in Appendix A (Table A1), so that the experiments can be reproduced.

The selected houses cover roughly two to four months of data each and differ markedly in composition. Some contain two cold appliances with similar compressor cycles (Houses 4, 10, 16, 17, 18 and 20), others contain several high-power wet appliances that often run on the same day (House 7), and some include low-power electronics such as televisions or a computer. Figure 2 shows three days of House 4 as an example of the input and target signals.

Table 2. REFIT houses used in this study, target appliances and fraction of samples removed during cleaning.

House	Target Appliances	Missing (%)
1	Fridge, freezer, television, electric heater	3.8
2	Fridge, dishwasher, television, kettle	61.1
3	Fridge, washing machine, television, kettle	1.7
4	Freezer, fridge–freezer, television, kettle	1.8
6	Freezer, washing machine, dishwasher, kettle	37.8
7	Freezer, tumble dryer, washing machine, dishwasher	34.1
10	Chest freezer, fridge–freezer, washing machine, dishwasher	28.9
11	Fridge, washing machine, kettle	1.5
13	Television, freezer, dishwasher, computer	0.0
15	Fridge–freezer, television, washing machine, tumble dryer	30.8
16	Two fridge–freezers, washing machine, dishwasher	28.1
17	Freezer, fridge–freezer, washing machine, kettle	0.4
18	Fridge–freezer, freezer, television, dishwasher	0.2
20	Fridge–freezer, freezer, washing machine, kettle	0.1
21	Fridge, multi-purpose socket, washing machine, dishwasher	0.0

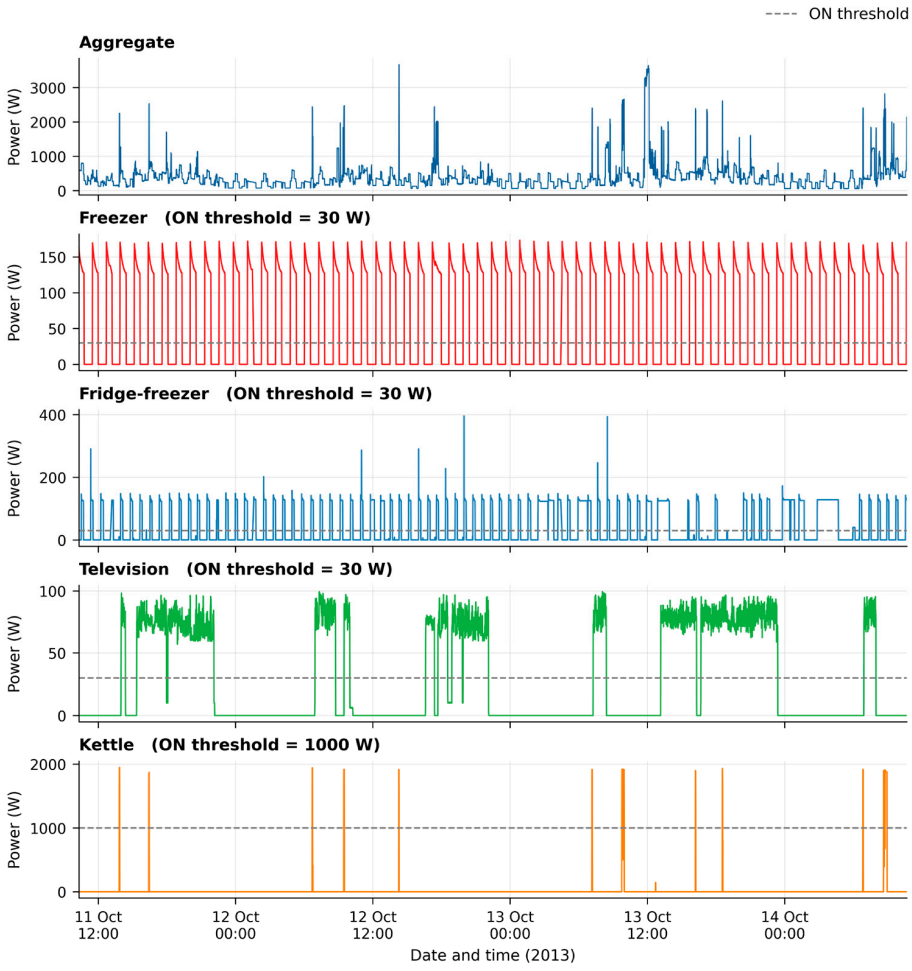


Figure 2. Aggregate power and the four target appliances of REFIT House 4 during the first three days of the selected period (30 s resolution). Dashed lines mark the ON thresholds.

3.3. Preprocessing

Resampling. Timestamps were converted to a regular 30 s grid. The median was used instead of the mean, because averaging smears the sharp ON/OFF edges that characterise most appliances, whereas the median rejects isolated spurious readings without shifting genuine transitions.

Choice of resolution. The raw REFIT measurements are available at a much finer resolution of about 8 s, and the resampling to 30 s was applied solely to reduce the computational cost, not because of any constraint of the data. Compared with the native 8 s sampling, it reduces the number of samples per day from 10,800 to 2880, a factor of 3.75. This shortens training proportionally and allows for a window of 199 samples to cover about 100 min. Osathanunkul and Osathanunkul [13] reported that, for an LSTM on UK-DALE, the training time per epoch fell from about 637 s at 6 s sampling to 88 s at 60 s, while F1 changed little and MAE deteriorated noticeably only beyond about 42 s; 30 s lies below this point. At 30 s, a kettle run of 2–4 min still spans 4–8 samples, a compressor cycle of 20–30 min spans 40–60 samples, and the steady power levels of most appliances are preserved. For the target appliances, whose operating events last from a few minutes to more than an hour, little relevant information is therefore lost. The loss concerns mainly short transients and low-power loads, whose changes of a few tens of watts can be hidden within a single 30 s interval. This is consistent with the weaker results for televisions and the computer reported in Section 4.4.

Gaps and clipping. Short gaps of up to ten consecutive samples (5 min) were forward-filled. Longer gaps were removed rather than imputed, so that no artificial segments entered the training data. The aggregate was clipped at zero and each appliance was clipped to its physical range, for example, 0–3100 W for kettles and 0–1000 W for fridge–freezers.

Normalisation. Inputs and targets were standardised with the mean and standard deviation of the training set:

$$\tilde{u} = \frac{u - \mu}{\sigma} \tag{3}$$

Windowing. Overlapping windows of $W = 199$ samples were extracted, corresponding to about 100 min at 30 s. Window lengths between 40 and 199 samples were tried in preliminary experiments. The longest one was kept, because washing machine and dishwasher cycles last 60–90 min and a shorter context leaves the network blind to most of the cycle. The series edges were zero-padded by $W/2$ samples.

Window length and appliance time scales. Earlier seq2point and sequence-to-sequence studies used appliance-specific windows, ranging from about 13 min for kettles to an hour or more for dishwashers [7,8]. Such windows are not compatible with the multi-output formulation used here, in which one network predicts all target appliances from the same input window. A common window therefore has to cover the longest relevant cycle, and W = 199 was chosen for this reason. For short appliances, the window contains more context than they strictly need, but the results give no sign that this penalised them: with the CNN, kettles had the highest mean F1 among appliance types with more than one instance. Representative results are provided in Section 4.4. Because all five models receive the same window, the comparison between architectures is not affected by this choice, although appliance-specific or multi-scale windows might improve the results for individual appliances.

Splitting and balancing. Each house was split chronologically into 70% training, 15% validation and 15% test data, which prevents leakage between neighbouring windows. This split measures how well a model trained on the earlier part of a household's record performs on a later period of the same household. It does not measure generalisation to households that were not seen during training. The ON fractions in the training data are highly unequal: cold appliances are on for roughly 20–45% of the time, whereas kettles

and washing machines are typically on for 0.2–1.5%. A model that always predicts OFF for the kettle would reach more than 99% accuracy and learn nothing. The data loader therefore draws batches in which half of the windows contain at least one rare appliance in the ON state.

3.4. Models

Five models were trained per house. They share the input window and the output layer size, so that they differ only in how they process the window. Their configuration is given in Table 3.

Table 3. Model architectures and number of trainable parameters for four outputs (three in House 11).

Model	Architecture	Parameters
Linear	Single fully connected layer, 199 → 4	800
MLP	199 → 256 → 128 → 64 → 4; ReLU; dropout 0.2 after each hidden layer	92,612
CNN	Five Conv1D layers (30, 30, 40, 50, 50 filters; kernels 10, 8, 6, 5, 5; ReLU); flatten; dense 1024; dropout 0.2; dense 4	10,384,924
LSTM	Conv1D (16 filters, kernel 4); two-layer LSTM (128 units, dropout 0.2); last hidden state; dense 64 (tanh); dense 4	215,444
Transformer	Linear projection 1 → 64; sinusoidal positional encoding; two encoder layers (4 heads, feed-forward 128, GELU, dropout 0.1); mean pooling over the window; dense 64 (GELU); dense 4	71,492

The linear model is a reference that shows how much of the task can be solved by a weighted sum of the window. The MLP adds non-linearity but treats the window as an unordered vector. The CNN follows the seq2point design [8]: convolutional filters detect local patterns such as steps and plateaus wherever they occur in the window, and weight sharing keeps the convolutional part small. Most of its ten million parameters sit in the dense layer that follows the flattened feature maps. The LSTM [27] processes the window step by step and can retain information over long spans through its gated cell state. A light convolutional front end reduces noise before the recurrent layers. A bidirectional variant was also tested in preliminary runs, but it trained much more slowly without a consistent F1 gain, so the unidirectional model was kept.

The Transformer model uses the encoder of Vaswani et al. [28] in a seq2point configuration. Each aggregate sample of the window is projected to a 64-dimensional embedding by a linear layer, and a fixed sinusoidal positional encoding is added. Two encoder layers with four attention heads, a feed-forward dimension of 128, GELU activations and dropout 0.1 then process the sequence. The encoder outputs are averaged over the 199 time steps and passed to a small dense head that predicts all target appliances. Self-attention lets every time step attend directly to every other one, so the model can relate events that are far apart in the window without the step-by-step recurrence of the LSTM. With 71,492 parameters, it is the smallest of the neural models.

All five models share the multi-output head, which is a deliberate choice. A per-appliance model would have multiplied the number of trained networks by four and, across fifteen houses and five architectures, would have required 300 training runs instead of 75. The shared head also forces the hidden representation to account for all target loads at once, which is closer to the situation of a real disaggregation service. The price is that the loss

of one appliance can interfere with another during training. The per-appliance weights of Section 3.5 are meant to keep this interference in check.

3.5. Training

With a plain MSE loss, errors on the few ON samples of rare appliances hardly affect the total loss, and the network learns to predict zero. A per-appliance weighted MSE was therefore used, in which errors are multiplied by a weight ω_c whenever the true power of appliance c exceeds its normalised ON threshold τ_c:

$$\mathcal{L} = \frac{1}{|B|}\sum_{j\in B}\sum_{c=1}^{M} w_{j,c}\left(\hat{x}_{j,c} - x_{j,c}\right)^2, \quad w_{j,c} = \begin{cases} \omega_c, & x_{j,c} \geq \tau_c \\ 1, & \text{otherwise} \end{cases} \tag{4}$$

The weights were selected on the validation set with the procedure described at the end of this section. The resulting weights were higher for rarer appliances: 3–4 for cold appliances; 3–6 for televisions, the computer, the multi-purpose socket and the electric heater; and 5–18 for dishwashers, washing machines, tumble dryers and kettles (Appendix A, Table A1). The approach is related in spirit to cost-sensitive losses such as the focal loss [29], but it keeps the regression target and needs no extra hyperparameter beyond the weight. ON thresholds were set per appliance and house from the observed power levels (Appendix A, Table A1). Table 4 summarises, per appliance type, the range of ON fractions, recorded peak power and thresholds over the fifteen houses. It shows at a glance why a single loss weight or threshold cannot suit every appliance.

Table 4. Characteristics of the target appliances over the fifteen houses (training data).

Appliance Type	Instances	ON Fraction (%)	Peak Power (W)	ON Threshold (W)
Fridge, freezer, fridge–freezer	22	19.0–49.3	600–2213 [1]	30–100
Television	7	11.2–100	56–2521	5–100
Dishwasher	8	0.5–13.5	1912–3009	100–1000
Washing machine	10	0.2–2.1	884–2484	100–500
Kettle	7	0.4–1.1	1985–3071	1000
Tumble dryer	2	1.0–5.3	2658–3589	500
Other (heater, computer, socket)	3	5.1–15.3	120–2936	30–500

[1] Peaks of cold appliances include short start-up and defrost transients. Normal compressor operation is around 70–120 W.

All models were trained in PyTorch on an NVIDIA T4 GPU with the following default settings: batch size 256, AdamW [30] with learning rate 10^{-3} and weight decay 10^{-4}, and at most 100,000 training and 30,000 validation windows per epoch (120,000 and 40,000 in House 2) drawn at random. A ReduceLROnPlateau scheduler halved the learning rate after two epochs without improvement of the validation loss. Early stopping [31] monitored the mean validation F1 rather than the loss, with a patience of seven epochs (ten in House 11) and a maximum of 30 to 50 epochs; the weights with the best validation F1 were restored before testing. Dropout [32] was used in the MLP, CNN, LSTM and Transformer as listed in Table 3.

Hyperparameter selection. Before fixing the configuration in Table 3, the dropout rate was varied from 0.1 to 0.5 in steps of 0.05, and a value of 0.2 was retained for the MLP, CNN and LSTM. The learning rate was kept at the AdamW default of 10^{-3}, which gave the most stable training in preliminary runs. For the LSTM, 32 and 64 hidden units were

tested in addition to 128, and for the MLP a variant with two hidden layers was tested; both gave lower validation F1. For the CNN, the five-layer configuration of Table 3 gave the best results among the configurations tried. Window lengths between 40 and 199 samples and a bidirectional LSTM variant were also compared, as described above. Combinations were compared on the mean validation F1, and the retained settings were then kept fixed across houses.

ON thresholds. For each appliance, the ON threshold was selected from the discrete set {5, 30, 100, 500, 1000} W (occasionally an intermediate level such as 50 or 300 W was also checked). The value was fixed from the training period power histogram as the level separating the appliance's OFF/standby band from its active power band: 5 W for standby-dominated television channels, 30 W for refrigeration, 100 W for medium loads, 500 W for the active phase of washing machines, tumble dryers and dishwashers, and 1000 W for kettles. No exhaustive per-house search was performed. The same thresholds define the ON samples in the weighted loss and are used to compute the detection metrics for all models.

Class weights. For each appliance, the class weight was tuned over the discrete set {3, 4, 5, 6, 8, 10, 12, 14, 15, 18}, starting from the value implied by its ON fraction (rarer appliance, larger weight) and from values reported in the literature. Thresholds were fixed first. Then, with all optimiser, scheduler and early stopping settings held constant, the weight of each appliance was increased or decreased within this set, and the value giving the best mean validation F1 was retained (a one-factor-at-a-time search, not a joint grid). In each house, the weights were selected on the validation F1 of the four models that share them (linear, MLP, CNN and LSTM), and the retained weights were then applied unchanged to all models, including the Transformer. The sensitivity analysis of Section 4.7 shows that the comparison is robust to the exact values. The retained per-household values are listed in Table A1.

The Transformer uses standard architectural settings for small encoders. It used the same optimiser, learning rate, batch size, scheduler and early stopping criterion as the other models, with at most 20 epochs, a patience of six epochs, and 80,000 training and 20,000 validation windows per epoch.

3.6. Evaluation Metrics

NILM performance has two partly independent aspects: detecting when an appliance is on and estimating how much power it draws [33]. Both were measured on the unseen test set. The mean absolute error in watts is

$$\mathrm{MAE} = \frac{1}{T}\sum_{t=1}^{T}|x_t - \hat{x}_t| \tag{5}$$

and was also computed over ON samples only ($\mathrm{MAE}_{\mathrm{on}}$), since the long OFF periods of most appliances make the overall MAE look optimistic. For detection, predictions and ground truth were converted to ON/OFF states with the appliance threshold, and the F1-score was computed as

$$\mathrm{F1} = \frac{2\,\mathrm{TP}}{2\,\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{6}$$

F1 ignores true negatives and is therefore more informative than accuracy for heavily imbalanced targets. Energy allocation was assessed with the estimation accuracy [33] and the signal aggregate error:

$$\mathrm{EstAcc} = 1 - \frac{\sum_t |\hat{x}_t - x_t|}{2\max\left(\sum_t x_t,\, 1\right)} \tag{7}$$

$$\mathrm{SAE} = \frac{\left|\sum_t \hat{x}_t - \sum_t x_t\right|}{\sum_t x_t} \tag{8}$$

EstAcc equals 1 for a perfect estimate and becomes negative when the absolute error exceeds twice the true energy. All metrics were computed per appliance and then averaged over the target appliances of each house.

4. Results

4.1. Overall Comparison Across Houses

Table 5 reports the average test set F1 of the five models in every house, together with the MAE and EstAcc of the CNN. Figure 3 shows the same F1 values graphically.

Table 5. Average test set F1 per house and model, and MAE and EstAcc of the CNN. Best F1 per house in bold.

House	Linear	MLP	CNN	LSTM	Transformer	CNN MAE (W)	CNN EstAcc
1	0.490	0.530	**0.562**	0.538	0.536	59.4	0.443
2	0.350	0.566	0.600	0.448	**0.618**	22.0	0.353
3	0.514	0.578	**0.665**	0.486	0.599	13.6	0.431
4	0.484	0.627	**0.737**	0.657	0.690	23.7	0.584
6	0.420	0.696	0.784	0.529	**0.792**	16.5	0.579
7	0.398	0.644	**0.765**	0.567	0.686	53.5	0.649
10	0.389	0.269	**0.635**	0.227	0.276	241.5	−1.338
11	0.328	0.442	**0.515**	0.260	0.161	21.9	0.161
13	0.353	0.584	**0.695**	0.368	0.624	18.4	0.553
15	0.444	0.706	0.726	0.607	**0.781**	15.8	0.498
16	0.356	0.588	0.600	0.514	**0.641**	28.2	0.510
17	0.336	0.654	**0.746**	0.449	0.717	19.5	0.256
18	0.443	0.686	**0.766**	0.652	0.749	24.5	0.654
20	0.515	0.694	**0.764**	0.489	0.681	15.6	0.546
21	0.304	0.446	**0.568**	0.442	0.419	28.0	0.421

Mean F1 over the fifteen houses: Linear 0.408, MLP 0.581, CNN 0.675, LSTM 0.482, Transformer 0.598. In the Transformer runs reported in Tables 5–7, the ON thresholds of House 2 (50, 80, 15 and 1700 W) and the input channels of the House 3 washing machine and the House 11 fridge differed from those used for the other models. Section 4.7 reports a repetition of the Transformer with exactly the same channels, ON thresholds and class weights as the CNN and the MLP.

Table 6. Per-appliance test set results in REFIT House 1. Best F1 in bold.

Appliance	ON (%)	F1 Linear	F1 MLP	F1 CNN	F1 LSTM	F1 Transformer	CNN MAE (W)	CNN EstAcc
Fridge	22.0	0.408	0.391	0.410	0.384	**0.427**	28.7	0.262
Freezer	38.4	0.586	0.596	0.575	0.553	**0.610**	31.3	0.482
Television	15.8	0.321	0.386	**0.459**	0.458	0.424	7.7	0.351
Electric heater	15.3	0.647	0.749	**0.805**	0.757	0.682	169.7	0.676
Average	–	0.490	0.530	**0.562**	0.538	0.536	59.4	0.443

The CNN achieved the highest average F1 in eleven of the fifteen houses, with values between 0.515 (House 11) and 0.784 (House 6) and a mean of 0.675. The Transformer was best in the other four (Houses 2, 6, 15 and 16), by margins of 0.008–0.055, and the CNN was second in each of them. The CNN also had the lowest median MAE (22.0 W, against 22.1 W for the Transformer, 25.6 W for the MLP, 41.6 W for the LSTM and 71.1 W for the linear model). The consistency of this result is notable, since the houses differ in the number

and type of target appliances, in data completeness and in the degree of class imbalance, yet the same five architectures were applied in every house.

Table 7. Test set F1 of the CNN and the Transformer grouped by appliance type across the fifteen houses.

Appliance Type	Instances	Mean F1 CNN	Range CNN	Mean F1 Transformer
Kettle	7	0.789	0.654–0.861	0.710
Dishwasher	8	0.769	0.515–0.970	0.620
Tumble dryer	2	0.710	0.664–0.756	0.744
Fridge, freezer, fridge–freezer	22	0.692	0.410–0.905	0.617
Washing machine	10	0.583	0.072–0.888	0.510
Television [1]	6	0.583	0.235–0.855	0.544
Electric heater	1	0.805	–	0.682
Computer	1	0.453	–	0.334
Multi-purpose socket	1	0.136	–	0.352

[1] Excluding House 3, where the television was ON in 100% of the samples.

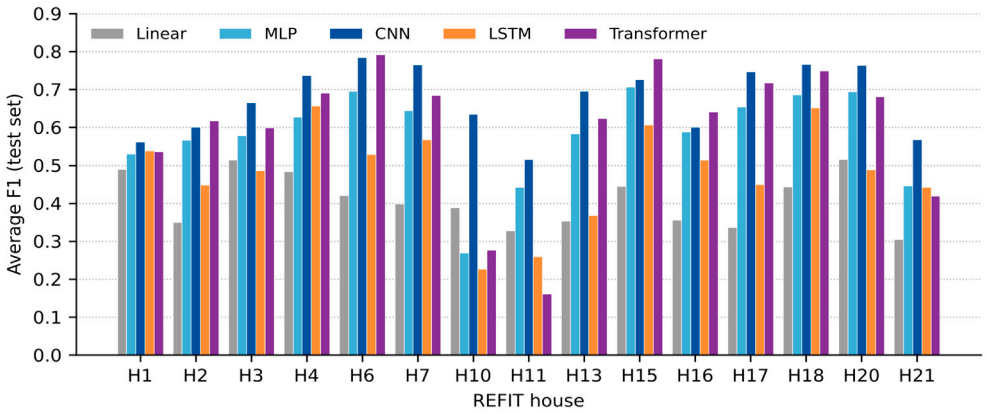


Figure 3. Average test set F1 of the five models in each REFIT house. Houses are labelled by their REFIT number.

Over all fifteen houses, the CNN had a higher mean F1 than the Transformer in Table 5 (0.675 against 0.598; Wilcoxon signed rank test, $p = 0.015$), and the Transformer had the second-highest mean F1 of the five models. Part of this gap, however, is due to differences in the original Transformer runs (see the note to Table 5). When both models were trained with identical channels, thresholds and weights (Section 4.7), the Transformer reached 0.637 against 0.683 for the CNN, and the difference was no longer statistically significant ($p = 0.083$). Its clearest weakness was House 10 (F1 = 0.274 against 0.641 for the CNN), and in House 11 it also stayed well below the CNN (0.380 against 0.506). Where it worked, it was competitive with the CNN with less than 1% of its parameters. The Transformer therefore shares both the strength and the weakness of the LSTM: it can match or exceed the CNN in favourable houses but it is less predictable across households.

The ordering of the remaining models was less predictable. Among the linear model, the MLP and the LSTM, the MLP was best in twelve houses, the LSTM in two (Houses 1 and 4) and the linear model in one (House 10). In House 4, the LSTM clearly outperformed the MLP (0.657 against 0.627). Both houses where the LSTM came out ahead are dominated by appliances with long, fairly regular cycles, such as the two cold appliances of House 4,

where the recurrent state has something to remember. In houses whose targets were short, sparse, high-power events, the LSTM struggled. It even fell below the linear baseline in Houses 3, 10, 11 and 20, despite having about 270 times more parameters.

House 10 is an outlier. The CNN still had the best F1 (0.635), but its average EstAcc was −1.338, the only negative value recorded for the CNN, and its MAE exceeded 240 W. The cause was a severe amplitude error on the washing machine (EstAcc −6.65), while the dishwasher of the same house was detected with F1 = 0.957. Good detection and poor energy estimation can therefore coexist within the same model and house. The Transformer showed the reverse pattern in the same house, with a less negative average EstAcc (−0.316) but a much lower F1 (0.276).

4.2. Case Study: REFIT House 1

House 1 illustrates how the average figures of Table 5 arise from quite different appliance-level behaviour. Its four targets are a fridge, a freezer, a television and an electric heater, covering 85 days between October 2013 and January 2014 with 3.8% of samples removed. Table 6 lists the per-appliance results of all five models.

The electric heater, with a sharp step of about 2 kW, was the easiest target for every model. Even the linear model reached F1 = 0.647, although its MAE on this appliance was 399 W against 170 W for the CNN, because a weighted sum of the window cannot reproduce the step amplitude. The television, whose ON level is only 30–35 W, separated the models most clearly: the CNN and the LSTM reached about 0.46, the linear model 0.32 with a negative EstAcc. The fridge was difficult for all models (F1 0.38–0.41) despite running 22% of the time. Its compressor cycles resemble those of the freezer in amplitude and period, and at 30 s the two are hard to tell apart in the aggregate. For the freezer, the MLP and even the linear model slightly exceeded the CNN, a reminder that the CNN's overall lead does not hold for every single appliance. The Transformer was the most accurate model on both cold appliances of this house (F1 0.427 for the fridge and 0.610 for the freezer), but its average F1 (0.536) stayed below that of the CNN because of a weaker result on the electric heater (0.682).

4.3. Training Behaviour

Figure 4 shows the training and validation curves in House 4, which is representative of the patterns seen elsewhere. The linear model converged within about ten epochs to a low validation F1 (below 0.5), a sign of underfitting. The MLP improved gradually over some 35 epochs. The CNN reached a validation F1 above 0.7 within the first ten epochs and then stayed almost flat. Its validation loss began to rise slightly while F1 was still stable, which supports stopping on F1 rather than on the loss. The LSTM was the slowest and least regular: its validation F1 was still increasing at epoch 49, close to the 50-epoch limit, and both loss and F1 showed a sharp transient around epoch 28. Similar incomplete convergence of the LSTM was observed in several other houses. The Transformer converged smoothly, with training and validation loss decreasing together and no sign of overfitting, but its validation F1 was still rising when training stopped at the 20-epoch limit, where it reached its best value.

4.4. Appliance-Level Results

Table 7 groups the CNN and Transformer results by appliance type across all houses. Figure 5 gives an example of the appliance-by-model F1 matrix for House 6. Compared with the CNN, the Transformer had a lower mean F1 for most appliance types but kept the same broad ordering: kettles and tumble dryers were among the easiest targets, and washing machines, televisions and the computer among the hardest.

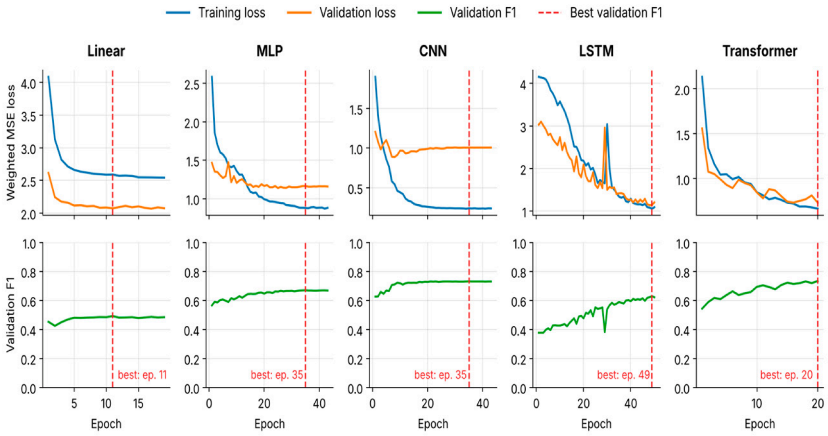


Figure 4. Training and validation loss (**top**) and validation F1 (**bottom**) per epoch for the five models in REFIT House 4. Vertical dashed lines mark the epoch with the best validation F1.

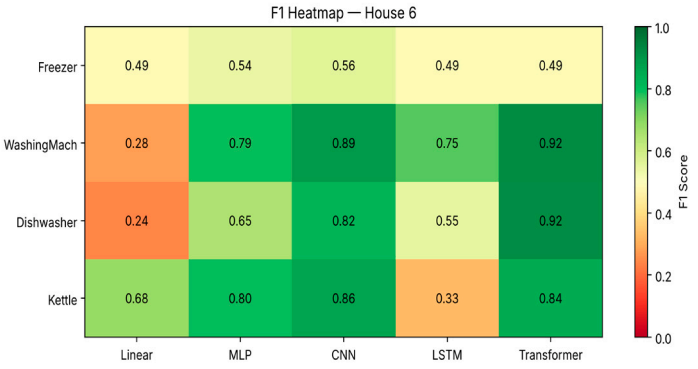


Figure 5. Test set F1 per appliance and model in REFIT House 6. The colour scale runs from 0 (red) to 1 (green).

Rarity alone did not predict difficulty. Kettles and dishwashers, which are typically on for only a few percent of the time, were among the best-detected appliances: the dishwasher of House 18 reached F1 = 0.970, and that of House 13 reached 0.842 while being active in less than 1% of the samples. Their high, flat and well-separated power plateaus stand out from the background. The electric heater in House 1, at about 2 kW, behaved similarly (F1 = 0.805). House 6 in Figure 5 illustrates the point: the washing machine, dishwasher and kettle, all rare, scored above 0.8 with the CNN, while the frequently running freezer stayed at 0.56. The Transformer showed the same pattern in House 6, with F1 values of 0.921, 0.916 and 0.841 for the three rare appliances and 0.489 for the freezer.

Figure 6 generalises this observation to all 58 appliance instances (the always-on television of House 3 is excluded). There is no monotonic relation between the ON fraction and the CNN F1 (Spearman $\rho = 0.03$, $p = 0.81$). Appliances that are on for less than 5% of the time reached a median F1 of 0.736 ($n = 25$), practically the same as the 0.724 of the more frequent ones ($n = 33$). Both groups contain easy and hard cases, and the spread within each appliance type is larger than the difference between the groups.

This observation should be read with some caution. The number of instances per appliance type is small and uneven, from 22 cold appliances to a single electric heater, computer and multi-purpose socket, and all houses come from one UK dataset with its own mix of appliances and usage habits. Households with other appliance portfolios, for example, with air conditioning, electric water heating or heat pumps, may behave differently.

The result therefore shows that rarity alone was not a reliable predictor of difficulty in the REFIT houses studied, rather than establishing a general ranking of appliance types.

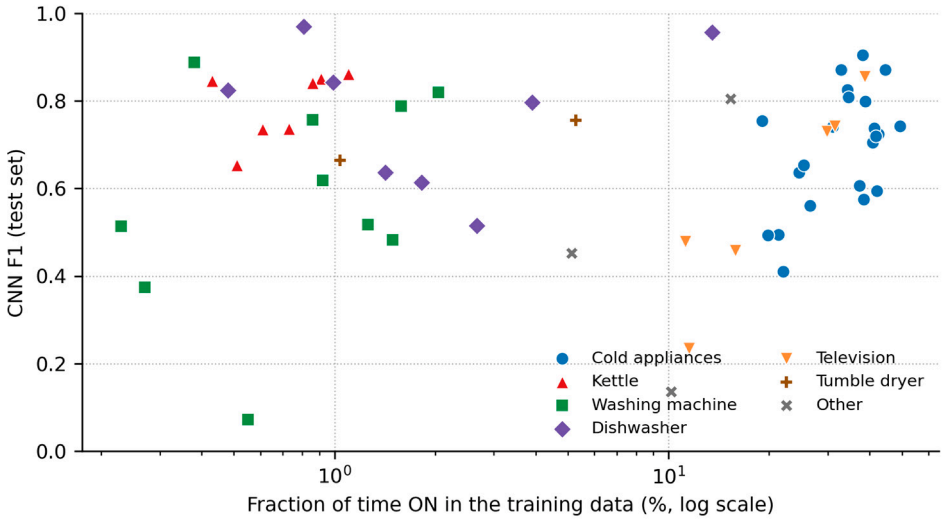


Figure 6. Test set F1 of the CNN versus the fraction of time each appliance is ON in the training data, for all appliance instances of the fifteen houses. The horizontal axis is logarithmic, and the always-on television of House 3 is excluded.

The hardest cases were low-power loads and loads whose operating level lies close to the ON threshold. Televisions ranged from 0.235 to 0.855 and a computer with a peak of 120 W reached only 0.453. In House 18, the linear model scored F1 = 0.001 on the television because its normal operating power almost coincided with the decision threshold. Washing machines showed the widest spread (0.072–0.888). Their multi-phase cycle, with short heating bursts separated by long low-power motor phases, is partly lost at 30 s and is easily confused with other loads. The poorest washing machine result came from House 11, which also had the lowest overall CNN F1.

Temporal overlap between target appliances was the second systematic source of error. It appeared in houses with two cold appliances (Houses 4, 10 and 16), in House 15, where the tumble dryer usually follows the washing machine, and in House 7 with three wet appliances that often run on the same day. The CNN was more robust to overlap than the MLP and the LSTM but did not eliminate it. The Transformer exceeded the CNN in two of these houses (15 and 16), which suggests that attention over the whole window helps when one load follows another, but it failed in House 10. Figure 7 shows CNN predictions for House 15: the fridge–freezer cycles and washing machine runs are tracked closely, whereas the television produces false activations and the two short tumble dryer peaks in this test segment are missed.

4.5. Data Quality and Performance

The fraction of removed samples varied from 0% to 61% (Table 2), yet it was essentially uncorrelated with the CNN F1 (Pearson $r \approx 0.01$). House 6, with 37.8% of samples removed, gave the second-best CNN F1 of the study (0.784), while House 11, with only 1.5% removed, gave the lowest. Within the range covered here, the distinctiveness of the target signatures mattered more than the completeness of the record. This does not mean that data quality is irrelevant: the chronological split means that long gaps change the season and usage covered by each subset, which may explain part of the distribution shift observed, for instance, for the freezer of House 17.

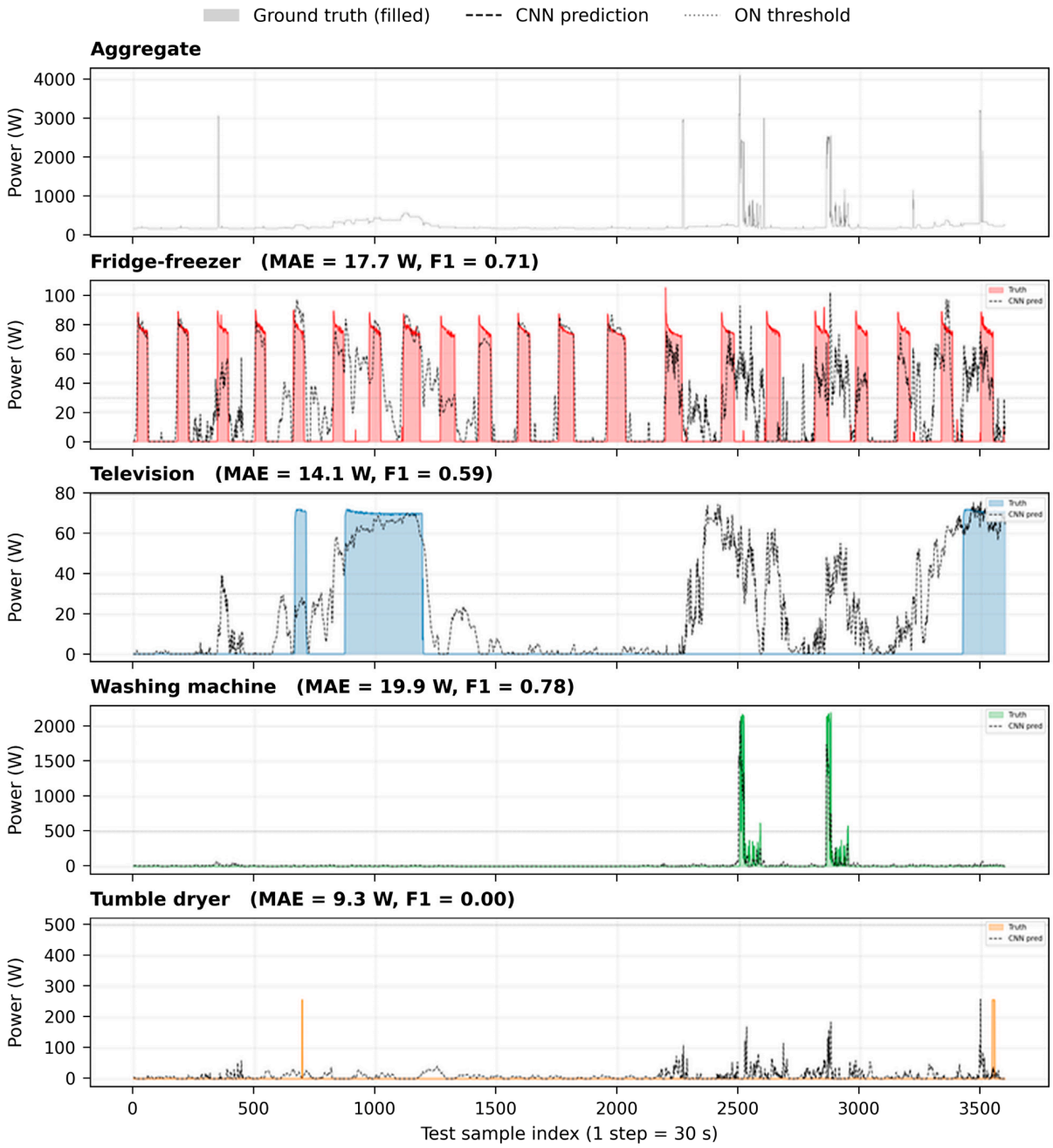


Figure 7. CNN predictions (dashed) against ground truth (filled) on a test segment of REFIT House 15. The MAE and F1 values in the panel titles refer to the displayed segment.

4.6. Model Size and Practical Cost

The five models span four orders of magnitude in size (Table 3): 800 parameters for the linear model, about 71 k for the Transformer, about 93 k for the MLP, 215 k for the LSTM and 10.4 million for the CNN. Size did not translate into accuracy in a simple way. The LSTM had more than twice the parameters of the MLP but a lower mean F1 (0.482 against 0.581), and in four houses it fell below a model with 800 parameters. The CNN was the most accurate, but more than 99% of its parameters belong to the dense layer that maps the 50 × 202 feature maps to 1024 units, not to the convolutional feature extractor. This suggests that most of the CNN's advantage comes from its convolutional front end and that its size could be reduced substantially, for example, with global pooling or a narrower dense layer, without giving up the inductive bias that makes it effective. The LSTM, in turn, is

the slowest to train because its recurrence over 199 time steps cannot be parallelised, which also limits its appeal for retraining on edge devices.

The Transformer makes the same point from the other side. With fewer parameters than the MLP, it reached a higher mean F1 (0.598 against 0.581) and the best F1 in four houses. Its cost, however, grows with the square of the window length, because every time step attends to every other one. At W = 199 this is still modest, but longer windows or higher sampling rates would increase it quickly.

Table 8 summarises the memory needed to store the weights of each model and the arithmetic cost of one prediction. The CNN needs about 42 MB in 32-bit floating point and still about 10 MB after 8-bit quantisation. This exceeds the 1–2 MB of flash memory typical of the microcontrollers used in meters and home gateways, such as the Cortex-M7 platform of [21], whereas the other neural models need less than 1 MB. In arithmetic cost, by contrast, the CNN is not the most expensive model: one prediction requires about 18 million multiply–accumulate operations, against 23 million for the Transformer and 40 million for the LSTM. Since a new prediction is needed only every 30 s, the computational load is modest for all models, and the binding constraint for the CNN is memory. The compression techniques discussed in Section 5 were not evaluated in this study, so the accuracy that a CNN small enough for such hardware would retain remains an open question. The trade-off between predictive accuracy and computational resources has also been examined in energy consumption prediction [34], and our results point to the same conclusion: the most accurate model is not necessarily the most suitable one for deployment, and the choice depends on the memory and update rate of the target device.

Table 8. Memory footprint and computational cost of one prediction for each model (four outputs).

Model	Parameters	Memory FP32 (MB)	Memory INT8 (MB)	MACs Per Prediction (Million)
Linear	800	0.003	<0.001	<0.001
MLP	92,612	0.37	0.09	0.09
CNN	10,384,924	41.5	10.4	17.9
LSTM	215,444	0.86	0.22	40.2
Transformer	71,492	0.29	0.07	23.2

Memory refers to the stored weights only. MACs are multiply–accumulate operations for one prediction (one window of 199 samples), computed analytically from the architectures in Table 3.

4.7. Sensitivity to Thresholds and Class Weights

The ON thresholds and class weights were set per appliance, so the ranking of the models could in principle depend on these choices. To test this, two additional experiments were carried out with the CNN, the MLP and the Transformer, all trained with identical input channels, ON thresholds and class weights (Table A1) and with the training budgets of Section 3.5. In the first, the three models were trained once in all fifteen houses with the nominal settings, and their test predictions were scored with every ON threshold multiplied by 0.5, 0.75, 1.0, 1.25 and 1.5. In the second, the three models were retrained in five houses (4, 6, 11, 16 and 20) with all class weights multiplied by 0.5, 1.0 and 2.0. The results are summarised in Table 9.

The nominal thresholds gave the highest mean F1 for all three models. The CNN had the highest mean F1 at every threshold scale and was ahead of the MLP in at least 13 of the 15 houses at each scale (Wilcoxon signed rank test, $p < 0.001$). With identical settings, the Transformer came closer to the CNN than in Table 5. At the nominal thresholds, its mean F1 was 0.637 against 0.683 for the CNN, the CNN was best in eleven houses and the Transformer in four, and the difference was not statistically significant ($p = 0.083$). The

gap between the two models was 0.033–0.046 for threshold scales up to 1.25 and almost vanished at 1.5 (0.006).

Table 9. Mean test set F1 of the CNN, the MLP and the Transformer when the ON thresholds or the class weights are multiplied by the given scale. Nominal settings in bold.

Experiment	Scale	CNN	MLP	Transformer	CNN − MLP	CNN − Transf.
ON threshold (15 houses)	0.5	0.634	0.557	0.601	0.077	0.033
	0.75	0.669	0.583	0.627	0.086	0.042
	1.0	**0.683**	**0.587**	**0.637**	**0.096**	**0.046**
	1.25	0.669	0.576	0.626	0.093	0.043
	1.5	0.642	0.561	0.636	0.082	0.006
Class weights (5 houses)	0.5	0.698	0.643	0.690	0.055	0.008
	1.0	**0.698**	**0.627**	**0.706**	**0.072**	**−0.008**
	2.0	0.674	0.600	0.638	0.075	0.036

Threshold experiment: models trained once per house with the nominal settings and scored with scaled thresholds. Weight experiment: models retrained in Houses 4, 6, 11, 16 and 20 with scaled class weights. All three models used identical input channels, ON thresholds and class weights.

Halving or doubling the class weights changed the mean F1 of the CNN by at most 0.024, against 0.027 for the MLP and 0.069 for the Transformer. The CNN remained ahead of the MLP at every weight scale (gap 0.055–0.075). Against the Transformer, the picture was mixed: the CNN was ahead at half and double weights (by 0.008 and 0.036), whereas at the nominal weights the Transformer was slightly higher in these five houses (0.706 against 0.698). The House 11 washing machine, scored with the same 500 W threshold for all models, remained difficult for every model (F1 = 0.106 for the CNN, 0.091 for the MLP and 0.077 for the Transformer). Overall, the advantage of the CNN over the MLP is robust to the choice of thresholds and class weights, whereas the CNN and the Transformer perform at a similar level, with the CNN slightly higher on average and less sensitive to the class weights.

5. Discussion

The main result is that a comparatively simple seq2point CNN was the most reliable choice, best in eleven of fifteen houses and second in the remaining four, with the same architecture in every house. This agrees with the popularity of the seq2point design as a baseline [8] and with its successful use in long-term embedded deployments [21]. Local convolutional features match the structure of the problem well: at 30 s, most appliances are recognised by edges, plateau heights and durations, and these can appear anywhere in the window.

The results are more cautious about recurrent models. The LSTM should in principle benefit from long temporal context, and it did so in the two houses dominated by regular cold appliance cycles. Elsewhere it was slow to converge and unstable, and it was sometimes worse than a linear model. This matches the observation that more complex neural NILM models do not automatically win under a fixed protocol [11] and that simpler learners can be competitive [12]. Architecture comparisons drawn from one or two houses should therefore be read with care, since the MLP–LSTM ordering reversed depending on the appliance mix.

The Transformer adds a further nuance. It was trained with the same data pipeline, input window and form of loss as the other models, yet a compact encoder with 71 k parameters matched or exceeded the CNN in four houses. The setup therefore does not by itself favour convolutions, and the CNN's consistency across the remaining houses points

to its local inductive bias as the main reason for its lead. The failure of the Transformer in House 10 resembles that of the LSTM: models that learn long-range dependencies from data appear more sensitive to the sparse and imbalanced targets of NILM than a convolutional network. A larger Transformer, or a hybrid that combines convolutional feature extraction with attention [9,10], may narrow the gap further.

A second point concerns evaluation. F1 and the amplitude-based metrics did not always move together. House 10 combined the best single-appliance detection with a strongly negative EstAcc. In House 2, low MAE did not imply high F1. In House 3, an always-on television inflated the averaged F1 of every model while telling nothing about disaggregation. We therefore recommend reporting detection (F1), amplitude (MAE, MAE_{on}) and energy allocation (EstAcc, SAE) together, as advocated by Makonin and Popowich [33], and checking appliance-level values before averaging.

From the perspective of a network operator, the results have a practical reading. Cold appliances, kettles, dishwashers and tumble dryers, which together account for a large part of the controllable residential load, were detected with mean F1 values between 0.69 and 0.79 using only 30 s active power. Washing machines and low-power electronics were far less reliable. Flexibility estimates built on NILM output should therefore carry appliance-specific uncertainty, and aggregated indicators such as the share of cold appliance load in a neighbourhood are likely to be more robust than single-household estimates. Because the CNN was best or close to best in every house with the same architecture, a single architecture could be deployed across a fleet of meters, with local fine-tuning reserved for the appliances where it is most needed.

The size of the observed differences also has a practical meaning. The gap of about 0.09 in mean F1 between the CNN and the MLP means that the simpler model misses or falsely reports a noticeably larger share of appliance activations. For feedback applications, this determines whether users can trust the appliance breakdown they are shown, and for demand response, whether the activations of flexible loads can be counted reliably. Differences between households matter as much as differences between models: the F1 of the CNN ranged from 0.515 to 0.784 across houses, so the accuracy expected for a new customer is uncertain and is better reported as a range than as a single figure. Across appliance categories, the reliable detection of cold appliances, kettles and dishwashers, together with the weak results for washing machines and low-power electronics, suggests that practical services should start with the appliance types that are already well recognised.

Beyond flexibility, the results connect to wider applications in energy efficiency and sustainable electricity use. Appliance-level feedback shows households where their consumption goes, and reliable detection of cold appliances, for example, can reveal old or faulty fridges and freezers whose consumption is unusually high. For demand-side management, knowing when wet appliances and kettles are used allows for aggregators and system operators to estimate how much load could be shifted away from peak hours without additional metering. For smart grid development, the fact that useful disaggregation was obtained from 30 s active power means that existing smart meter infrastructure can support such services, avoiding the cost and the material footprint of dedicated sensors in every home. Together, these applications make residential demand more visible and more flexible, which becomes increasingly important as the share of variable renewable generation grows.

The evaluation was designed to compare architectures under identical conditions, and several design choices follow from this aim. Each model was trained and tested within the same house, so the results describe how consistently each architecture performs across households rather than how it transfers to unseen homes. A leave-one-house-out evaluation is not compatible with the multi-output formulation, because the target appliances differ between houses; it calls for appliance-specific models trained across houses, which is a

natural extension of this work. The hyperparameters of each deep learning model were selected in preliminary trials (Section 3.5), and the ON thresholds and class weights were set for every appliance (Appendix A). The comparison deliberately focused on networks that learn directly from the raw power window; feature-based methods such as gradient-boosted trees, which performed well at 15 min resolution [12], answer a different question and can be included in future benchmarks. The sensitivity analysis of Section 4.7 showed that the lead of the CNN over the MLP is robust to the ON thresholds and class weights, whereas the CNN and the Transformer performed at a similar level; the class weights were tested in five houses. All experiments used active power at 30 s. Based on [13], the ranking is expected to hold at nearby resolutions, while low-power and short-cycle appliances may benefit from finer sampling and from appliance-specific or multi-scale windows. Finally, the memory footprint of the CNN, dominated by a single dense layer, is the main obstacle to on-device deployment and a clear target for compression.

These points suggest concrete next steps. Transfer learning [26] and federated training [24,25] could test whether a model pre-trained on many REFIT houses adapts to a new one with little data while keeping raw consumption at home. Overlap could be addressed with hybrid CNN–LSTM models with attention [10] or with source separation architectures. Rare events could benefit from focal-type losses [29] or event-level augmentation instead of fixed weights. Adaptive, house-specific thresholds would reduce the sensitivity observed for low-power appliances, and extra inputs such as reactive power or the power derivative could help separate similar signatures. For deployment, replacing the large dense layer with global pooling, or compressing the CNN by pruning, quantisation or knowledge distillation [35], would bring it closer to the lightweight models already demonstrated on embedded hardware [22].

6. Conclusions and Future Study Proposals

This paper compared a linear baseline, an MLP, a seq2point CNN, an LSTM and a Transformer encoder for NILM on fifteen REFIT houses at 30 s resolution (trained and tested separately in each house), using one fixed pipeline with a multi-output formulation and a per-appliance weighted loss. The CNN obtained the best average F1 in eleven of fifteen houses (mean 0.675) and the lowest median MAE (22.0 W). A compact Transformer with 71 k parameters was best in the other four houses, and with the same settings as the CNN its mean F1 was within 0.05 of that of the CNN. The ranking of the remaining models depended on the appliances of each house, and the LSTM and the Transformer were the least stable models, falling below the linear baseline in four and two houses, respectively. In the REFIT houses studied, appliance difficulty was governed mainly by the distinctiveness of the power signature, the distance between the operating level and the ON threshold, and temporal overlap with other loads, rather than by how rarely the appliance was used. Detection and energy estimation diverged in several houses, so multi-metric reporting is necessary. A sensitivity analysis showed that the lead of the CNN over the MLP is preserved under substantial changes in the ON thresholds and class weights, while the CNN and the Transformer remained close. In practical terms, the study shows that useful appliance-level information for cold appliances, kettles and dishwashers can be obtained from 30 s active power with a single architecture across households, which is a prerequisite for NILM-based feedback and flexibility services built on existing smart meters. Future work will address cross-house generalisation, overlap-aware and compressed architectures, and adaptive thresholds, with the aim of NILM models that can run on smart meter and home gateway hardware. In particular, implementation and testing on real metering hardware and the transfer of trained models to previously unseen households are the two steps needed before such services can be deployed at scale.

Several directions for future study follow from these results. The first is to move from house-specific training to models that generalise across households. A network pre-trained on the pooled REFIT houses and fine-tuned with a few days of data from a new home would show how much of the per-house effort can be avoided, and federated training would allow for the same while keeping raw consumption data on the customer side. The second concerns the appliances that remained unreliable. Washing machines and low-power electronics could benefit from event-level data augmentation, focal-type losses and adaptive thresholds derived from each appliance's own power distribution, while hybrid convolutional–recurrent or attention-based models could be tested specifically on houses with strong load overlap. The third is the sampling rate: repeating the experiments at 8–10 s, and with additional inputs such as reactive power where available, would quantify how much detail is lost at 30 s and whether it justifies the extra data volume. Finally, the evaluation should be extended beyond the United Kingdom. Southern European households, with air conditioning, electric water heaters and a growing number of heat pumps and electric vehicles, present load profiles that are absent from REFIT and are highly relevant to distribution-level flexibility. Testing compressed versions of the CNN on real smart meter or gateway hardware in such homes would be the natural step from a benchmark study towards an operational NILM service.

Author Contributions: Conceptualization, V.L.; methodology, V.L.; software, V.L.; validation, V.L. and F.K.; formal analysis, V.L. and F.K.; investigation, V.L. and F.K.; data curation, V.L. and F.K.; writing—original draft preparation, V.L.; writing—review and editing, E.O.K.; visualization, V.L.; supervision, E.O.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The REFIT Electrical Load Measurements dataset analysed in this study is publicly available from the University of Strathclyde [6]. The processing code is available from the corresponding author upon reasonable request.

Conflicts of Interest: Author Vasileios Laitsos was employed by the HEDNO. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional neural network
EstAcc	Estimation accuracy
F1	F1-score
GELU	Gaussian error linear unit
IAM	Individual appliance monitor
LSTM	Long short-term memory
MAE	Mean absolute error
MLP	Multilayer perceptron
MSE	Mean squared error
NILM	Non-intrusive load monitoring
SAE	Signal aggregate error
Seq2point	Sequence-to-point

Appendix A

Table A1 lists the ON thresholds and class weights used for each target appliance and house. The same thresholds were used to define the ON samples in the weighted loss and to compute the detection metrics.

Table A1. ON thresholds and class weights of the target appliances in each REFIT house.

House	Target Appliance: ON Threshold (W)/Class Weight ω
1	Fridge 30/4; freezer 30/3; television 5/6; electric heater 500/5
2	Fridge 30/3; dishwasher 100/6; television 5/5; kettle 1000/10
3	Fridge 30/3; washing machine 100/8; television 5/3; kettle 1000/12
4	Freezer 30/3; fridge–freezer 30/3; television 30/4; kettle 1000/15
6	Freezer 30/3; washing machine 500/12; dishwasher 500/12; kettle 1000/15
7	Freezer 30/3; tumble dryer 500/6; washing machine 500/6; dishwasher 500/6
10	Chest freezer 30/3; fridge–freezer 30/3; washing machine 500/10; dishwasher 100/5
11	Fridge 30/3; washing machine 500/18; kettle 1000/12
13	Television 30/3; freezer 30/3; dishwasher 500/15; computer 30/4
15	Fridge–freezer 30/3; television 30/3; washing machine 500/10; tumble dryer 500/12
16	Fridge–freezer 1 30/3; fridge–freezer 2 30/3; washing machine 500/10; dishwasher 100/12
17	Freezer 100/3; fridge–freezer 30/3; washing machine 100/12; kettle 1000/15
18	Fridge–freezer 30/3; freezer 30/3; television 100/4; dishwasher 1000/15
20	Fridge–freezer 30/3; freezer 30/3; washing machine 500/8; kettle 1000/15
21	Fridge 30/3; multi-purpose socket 30/4; washing machine 500/10; dishwasher 500/15

Values used for all models. In the Transformer runs reported in Tables 5–7, the ON thresholds of House 2 (50, 80, 15 and 1700 W) and the input channels of the House 3 washing machine and the House 11 fridge differed from these; the runs of Section 4.7 use the values above for all models.

References

1. Babaa, S.E.; Armstrong, M.; Pickert, V. Evolution of Conventional Power Systems to Smart Grids. In Proceedings of the 2019 54th International Universities Power Engineering Conference (UPEC), Bucharest, Romania, 3–6 September 2019; pp. 1–6. [CrossRef]
2. Hart, G.W. Nonintrusive Appliance Load Monitoring. *Proc. IEEE* **1992**, *80*, 1870–1891. [CrossRef]
3. Zoha, A.; Gluhak, A.; Imran, M.A.; Rajasegarar, S. Non-Intrusive Load Monitoring Approaches for Disaggregated Energy Sensing: A Survey. *Sensors* **2012**, *12*, 16838–16866. [CrossRef] [PubMed]
4. Kolter, J.Z.; Johnson, M.J. REDD: A Public Data Set for Energy Disaggregation Research. In Proceedings of the SustKDD Workshop on Data Mining Applications in Sustainability, San Diego, CA, USA, 21 August 2011.
5. Kelly, J.; Knottenbelt, W. The UK-DALE Dataset, Domestic Appliance-Level Electricity Demand and Whole-House Demand from Five UK Homes. *Sci. Data* **2015**, *2*, 150007. [CrossRef] [PubMed]
6. Murray, D.; Stankovic, L.; Stankovic, V. An Electrical Load Measurements Dataset of United Kingdom Households from a Two-Year Longitudinal Study. *Sci. Data* **2017**, *4*, 160122. [CrossRef] [PubMed]
7. Kelly, J.; Knottenbelt, W. 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 (BuildSys), Seoul, Republic of Korea, 4–5 November 2015; pp. 55–64. [CrossRef]
8. Zhang, C.; Zhong, M.; Wang, Z.; Goddard, N.; Sutton, C. Sequence-to-Point Learning with Neural Networks for Non-Intrusive Load Monitoring. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32, pp. 2604–2611. [CrossRef]
9. Kong, X.; Gui, Z.; Wu, M.; Miao, C.; Luo, Z. A Hybrid Deep Learning Framework for Non-Intrusive Load Monitoring. *Electronics* **2026**, *15*, 453. [CrossRef]
10. Zhang, J.; Sun, J.; Gan, J.; Liu, Q.; Liu, X. Improving Domestic NILM Using an Attention-Enabled Seq2Point Learning Approach. In Proceedings of the 2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Calgary, AB, Canada, 25–28 October 2021; pp. 434–439. [CrossRef]

11. Bousbiat, H.; Faustine, A.; Klemenjak, C.; Pereira, L.; Elmenreich, W. Unlocking the Full Potential of Neural NILM: On Automation, Hyperparameters and Modular Pipelines. *IEEE Trans. Ind. Inform.* **2023**, *19*, 7002–7010. [CrossRef]
12. Shabbir, N.; Vassiljeva, K.; Nourollahi Hokmabad, H.; Husev, O.; Petlenkov, E.; Belikov, J. Comparative Analysis of Machine Learning Techniques for Non-Intrusive Load Monitoring. *Electronics* **2024**, *13*, 1420. [CrossRef]
13. Osathanunkul, K.; Osathanunkul, K. Different Sampling Rates on Neural NILM Energy Disaggregation. In Proceedings of the 2019 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON), Nan, Thailand, 30 January–2 February 2019; pp. 318–321. [CrossRef]
14. Hussein, N.M.; Hesham, A.M.; Rashawn, M.A. States and Power Consumption Estimation for NILM. In Proceedings of the 2019 14th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, 17 December 2019; pp. 275–281. [CrossRef]
15. Berrettoni, G.; Bourelly, C.; Capriglione, D.; Ferrigno, L.; Miele, G. Preliminary Sensitivity Analysis of Combinatorial Optimization (CO) for NILM Applications: Effect of the Meter Accuracy. In Proceedings of the 2021 IEEE 6th International Forum on Research and Technology for Society and Industry (RTSI), Naples, Italy, 6–9 September 2021; pp. 486–490. [CrossRef]
16. Ghaffar, M.; Sheikh, S.R.; Naseer, N.; Din, Z.M.U.; Rehman, H.Z.U.; Naved, M. Non-Intrusive Load Monitoring of Buildings Using Spectral Clustering. *Sensors* **2022**, *22*, 4036. [CrossRef] [PubMed]
17. Makonin, S. Investigating the Switch Continuity Principle Assumed in Non-Intrusive Load Monitoring (NILM). In Proceedings of the 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), Vancouver, BC, Canada, 15–18 May 2016; pp. 1–4.
18. Jacobs, G.; Henneaux, P. Specific and Generic Unsupervised Algorithms for NILM Applications. In Proceedings of the 2020 International Conference on Smart Energy Systems and Technologies (SEST), Istanbul, Turkey, 7–9 September 2020; pp. 1–6.
19. Meng, F.; Feng, Z.; Lu, L.; Bao, Z.; Yu, M. Enhancing Appliance Classification in NILM by Power-Theoretic Color-Coding V-I Trajectory Visualization. In Proceedings of the 2025 IEEE 14th Global Conference on Consumer Electronics (GCCE), Osaka, Japan, 23–26 September 2025.
20. Nour, M.; Ravier, P.; Le Bunetel, J.-C.; Raingeaud, Y.; Lamarque, G. Artificial Aggregation of Domestic Appliance Currents for NILM Evaluation. In Proceedings of the 2021 IEEE International Conference on Environment and Electrical Engineering and 2021 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), Bari, Italy, 7–10 September 2021; pp. 1–6. [CrossRef]
21. Mari, S.; Bucci, G.; Ciancetta, F.; Fiorucci, E.; Fioravanti, A. An Embedded Deep Learning NILM System: A Year-Long Field Study in Real Houses. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 2531215. [CrossRef]
22. Lei, Y.; Feng, Z.; Lin, X.; Cao, D.; Zhang, J. NILM-LANN: A Lightweight Attention-Based Neural Network in Non-Intrusive Load Monitoring. In Proceedings of the 2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Tianjin, China, 8–10 May 2024; pp. 181–186. [CrossRef]
23. Gowrienanthan, B.; Kiruthihan, N.; Rathnayake, K.D.I.S.; Kiruthikan, S.; Logeeshan, V.; Kumarawadu, S. Deep Learning Based Non-Intrusive Load Monitoring for a Three-Phase System. *IEEE Access* **2023**, *11*, 49337–49350. [CrossRef]
24. Bousbiat, H.; Klemenjak, C.; Himeur, Y.; Elmenreich, W.; Amira, A.; Mansoor, W.; Atalla, S. Neural NILM Learning Paradigms: From Centralised to Decentralised Learning. In Proceedings of the 2022 5th International Conference on Signal Processing and Information Security (ICSPIS), Dubai, United Arab Emirates, 7–8 December 2022; pp. 138–142. [CrossRef]
25. Zhang, Y.; Tang, G.; Huang, Q.; Wang, Y.; Wu, K.; Yu, K.; Shao, X. FedNILM: Applying Federated Learning to NILM Applications at the Edge. *IEEE Trans. Green Commun. Netw.* **2023**, *7*, 857–868. [CrossRef]
26. D'Incecco, M.; Squartini, S.; Zhong, M. Transfer Learning for Non-Intrusive Load Monitoring. *IEEE Trans. Smart Grid* **2020**, *11*, 1419–1429. [CrossRef]
27. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
28. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
29. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988. [CrossRef]
30. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
31. Prechelt, L. Early Stopping—But When? In *Neural Networks: Tricks of the Trade*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1998; Volume 1524, pp. 55–69. [CrossRef] [PubMed]
32. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.

33. Makonin, S.; Popowich, F. Nonintrusive Load Monitoring (NILM) Performance Evaluation: A Unified Approach for Accuracy Reporting. *Energy Effic.* **2015**, *8*, 809–814. [CrossRef]
34. Ziran, Z.; Muzi, F.; Piras, G.; Arman, A.; Mecella, M. Supporting Energy Consumption Prediction: A Sustainable Approach. In Proceedings of the 2025 21st International Conference on Intelligent Environments (IE), Darmstadt, Germany, 23–26 June 2025; pp. 1–8. [CrossRef]
35. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* **2015**, arXiv:1503.02531.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.