

Article

Nonintrusive Load Monitoring (NILM) Using a Deep Learning Model with a Transformer-Based Attention Mechanism and Temporal Pooling

Mohammad Irani Azad ¹, Roozbeh Rajabi ²  and Abouzar Estebarsari ^{3,*} 

¹ Electrical Engineering Department, Shahid Beheshti University, Tehran 1983969411, Iran; m_iraniazad@sbu.ac.ir

² DITEN Department, University of Genoa, 16145 Genoa, Italy; roozbeh.rajabi@unige.it

³ School of the Built Environment and Architecture, London South Bank University, London SE1 0AA, UK

* Correspondence: estebaaa@lsbu.ac.uk

Abstract: Nonintrusive load monitoring (NILM) is an important technique for energy management and conservation. In this paper, a deep learning model based on an attention mechanism, temporal pooling, residual connections, and transformers is proposed. This article presents a novel approach for NILM to accurately discern energy consumption patterns of individual household appliances. The proposed method entails a sequence of layers, including encoders, transformers, attention, temporal pooling, and residual connections, offering a comprehensive solution for NILM while effectively capturing appliance-specific energy usage in a household. The proposed model was evaluated using UK-DALE, REDD, and REFIT datasets in both seen and unseen cases. It shows that the proposed model in this paper performs better than other methods stated in other papers in terms of F1-score and total error of the results (in terms of SAE). This model achieved an F1-score equal to 92.96 as well as a total SAE equal to -0.036 , which shows its effectiveness in accurately diagnosing and estimating the energy consumption of individual home appliances. The findings of this research show that the proposed model can be a tool for energy management in residential and commercial buildings.

Keywords: nonintrusive load monitoring (NILM); deep learning; attention mechanism; temporal pooling; residual connections; transformers



Citation: Irani Azad, M.; Rajabi, R.; Estebarsari, A. Nonintrusive Load Monitoring (NILM) Using a Deep Learning Model with a

Transformer-Based Attention Mechanism and Temporal Pooling. *Electronics* **2024**, *13*, 407. <https://doi.org/10.3390/electronics13020407>

Academic Editor: Pedro J. Villegas

Received: 2 December 2023

Revised: 7 January 2024

Accepted: 9 January 2024

Published: 18 January 2024



Copyright: © 2024 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

NILM is the process of identifying loads and their power consumption from a power source using a separation algorithm. By continuously monitoring the energy consumption of buildings, one can proactively identify and prevent energy wastage. This information can then be communicated to consumers, empowering them to take necessary actions to optimize energy usage. It has been reported that consumer behavior plays a vital role in the effective use of energy. Additionally, consumers are more likely to modify their energy consumption patterns, if they choose to do so. Smart meters provide information solely on the overall energy consumption at the building level rather than at the individual appliance level. These aggregate energy consumption data are valuable for load forecasting purposes, as highlighted in [1]. Nevertheless, many studies indicate that knowledge of total energy consumption is unlikely to cause a substantial shift in consumer energy consumption behavior [2]. Typically, the advantages of employing NILM include:

1.1. Energy Efficiency

By knowing which appliances consume the most energy, people can take steps to optimize usage patterns and save on energy bills.

1.2. Demand Response

NILM enables the identification of high-demand appliances and their usage patterns. This information is valuable for utilities and grid operators in order to implement demand response programs that include modifying power consumption during peak periods to reduce pressure on the power grid.

1.3. Monitoring and Maintenance of Home Appliances

NILM can help identify irregularities, breakdowns, or inefficient performance by monitoring individual home appliances. It can provide insights into the performance of devices and alert users to potential problems, allowing them to take preventive maintenance actions or replace faulty equipment.

1.4. Resident Behavior Analysis

NILM can provide valuable information about resident behavior and lifestyle patterns based on their energy consumption profile. This information can be used for various purposes such as designing targeted energy-saving programs, understanding occupant comfort, or optimizing building design and operation.

1.5. Load Balancing and Optimization

NILM helps in understanding the distribution of power consumption across different devices. This knowledge can be used to balance loads, manage peak demands more effectively, and optimize energy consumption within a building or across a network.

1.6. Energy Consumption Audit

NILM proves to be a valuable instrument for assessing energy usage across residential, commercial, or industrial settings. It provides intricate details regarding energy consumption at the individual device level, enabling auditors to pinpoint areas for enhancing energy conservation and assess the efficacy of implemented energy efficiency measures [3].

NILM has undergone a shift in methodology with the advent of neural network (NN) methods, presenting notable differences and advantages over traditional approaches. Traditional methods often rely on manually extracted features and rule-based algorithms to disaggregate energy consumption data. In contrast, NN methods harness the power of deep learning, allowing for automatic feature extraction and the learning of delicate patterns within the data. This inherent adaptability enables NN methods to handle diverse and complex energy consumption scenarios more effectively. Additionally, NN methods tend to exhibit improved performance when faced with noisy or unstructured data, enhancing their robustness in real-world applications. The ability of NN methods to adapt and generalize to varying load patterns makes them particularly advantageous in the dynamic and evolving landscape of NILM, offering a promising path for more accurate and versatile energy disaggregation.

In this paper, we introduce a new deep learning model incorporating encoders, temporal pooling, residual connections, and transformers to build up a comprehensive method for NILM applications. The proposed model is applied to different public NILM datasets, and the performance is evaluated based on common metrics. The results show the efficacy of the proposed method in comparison with other, previous models.

The remainder of this paper is structured as follows. Section 2 provides a comprehensive review of prior works pertaining to NILM. Subsequently, Section 3 delves into a discussion on the datasets and evaluation criteria. Section 4 outlines the proposed method in detail. Following this, Section 5 explores experiments and results, drawing comparisons with other methodologies. Lastly, Section 6 concludes the paper, highlighting potential future directions.

2. A Review of Previous Works Related to NILM

NILM approaches can be generally divided into supervised and unsupervised methods [4]. In the supervised method, the power consumption of appliances is collected and can be used to train models. NILM unsupervised methods also include hidden Markov models (HMM) [5,6], factorial hidden Markov models (FHMM) [6,7], and methods based on event detection and clustering [8,9]. Comprehensive reviews of NILM unsupervised methods can be found in [7,10]. Also, with the development of deep neural networks, various methods based on supervised NILM neural networks have also been presented [11,12]. Recently, thanks to convolutional neural networks (CNNs), significant progress has been made in this field [13,14]. Most NILM methodologies presented in the literature are based on approaches using signal processing [7,8], factorial hidden Markov models [5,6,8,14], or deep neural networks [11].

Also, our previous work entitled “Non-Intrusive Load Monitoring (NILM) Using deep neural networks: A Review” can be seen in [15]. This paper reviews some recent NILM methods based on deep learning and introduces the most accurate methods for residential loads. It summarizes public databases for NILM evaluation and compares methods using standard performance metrics. In the following, several methods related to solving the NILM problem are introduced with the help of deep learning methods, which are used to present the proposed method.

2.1. The WaveNILM Method

The method presented in [16] is called WaveNILM. In the paper, the authors explain that NILM is an important tool for energy-saving purposes as it allows the estimation of the energy consumption of a device from a single measurement. The WaveNILM network architecture is based on Dilated causal convolutional layer (DC-CNN). This version of DC-CNN adds a gating mechanism to the output of the DC-CNN filters, which has the ability to control the flow of information during the convolutional layer, and it achieves this by multiplying the output of each filter by the gate values. This mechanism allows the convolutional layer to selectively enhance or suppress the features in our input sequence based on their relevance to the problem in question. Samples related to current and previous time steps have been used as input in dilated causal convolutions. Then, the output of each convolutional layer is given as input to the sigmoid (gate) activation function and the ReLu(regressor) activation function. Then, these two output values (from the two named activator functions) are multiplied together, and the desired block output is obtained. After that, the output of each block is copied. One part is used as an input for the next layer, and the other part passes through all the subsequent convolutional layers and is used in the final layer of the WAVENILM network (skip connection). Each of these layers also has 10% dropout [16].

2.2. The Variational Autoencoders Method

In the article [17], a new method based on variational autoencoders is proposed to calculate the power consumption of each electrical device. This method is an unsupervised method, and it has been shown that the said method performs better than many algorithms used in NILM. The network used in this paper consists of two main parts: IBN-Net and VAE. The IBN-Net network is used to extract relevant features from the raw power consumption measurements. This structure has been used to extract features in the VAE model. The VAE model also consists of an encoder and a decoder. The encoder network maps the measured power values to the space with lower dimensions, while on the other hand, the decoder network maps the data from this space with lower dimensions to the estimated power consumption values of each of the existing electrical devices. Figure 1 shows the proposed network structure in the variational autoencoders method.

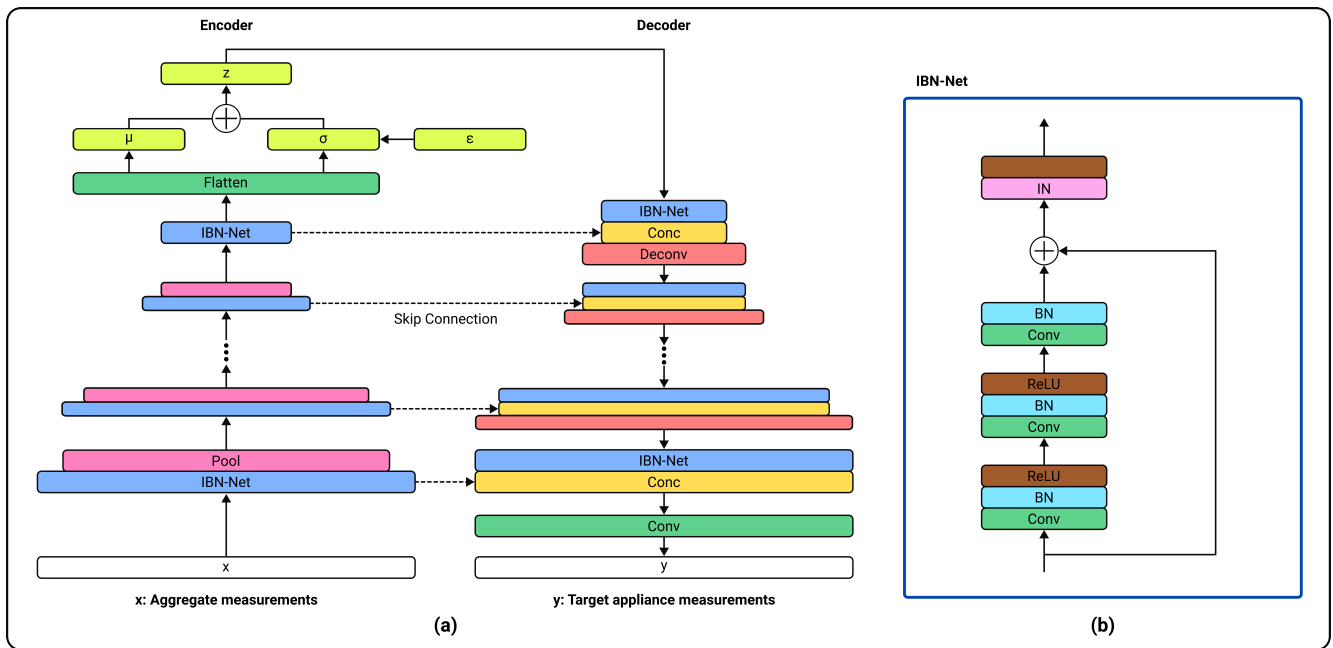


Figure 1. Proposed network in variational autoencoders method (a) VAE structure (b) IBN-Net. Adapted with permission from Ref. [17].

2.3. The COLD Method

The article [18] used the SNS algorithm to solve the NILM problem. In the article, a neural network-based structure called COLD is developed, which is able to identify 1 to 10 electrical devices that work simultaneously. The synthetic data generated with the help of SNS provide a less accurate model than the real measurements. In the article, artificial data generated through the utilization of the SNS algorithm is employed, wherein simulated mass power consumption data are simulated. a maximum of 10 times at the same time. This synthetic data generated, with the help of the SNS algorithm, are used as input for COLD network training and evaluation. The network introduced in the article is based on the deep ReLU network with the proposed self-attention mechanism. The core of the proposed network is the ReLU feedforward network, which is able to estimate any continuous function. The input of the network is the matrix of spectrograms obtained from the STFT corresponding to the cumulative consumption data signal, and the output of the network are the binary vectors that indicate the activity or inactivity of electrical appliances at each time step. Figure 2 shows the COLD network structure.

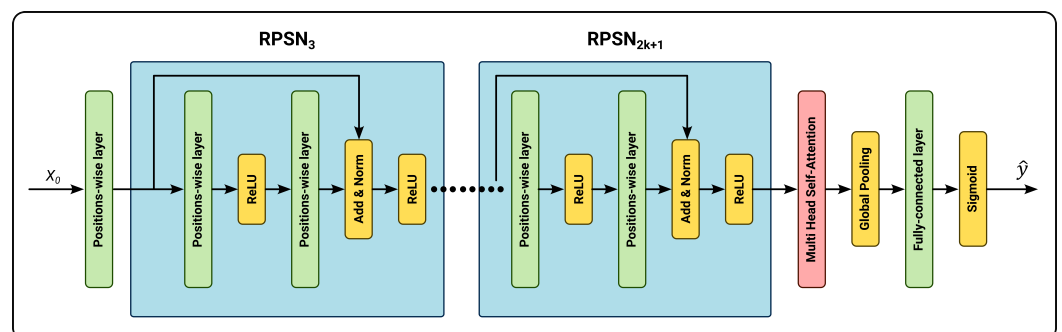


Figure 2. COLD network structure [18].

2.4. The ELECTRICity Method

In the article [19], a new method for solving the NILM problem based on transformers is presented. ELECTRICity uses a transformer to extract features from the cumulative signal. In the article, the UK-DALE dataset and a dataset collected from a household in Greece

are used. The proposed network consists of two main parts: a preprocessing part and a training part. During the preprocessing stage, the model consists of a transformer-based generator and a discriminator to improve the performance of the model. The generator has the task of producing synthetic signals for electrical devices using a cumulative signal. The discriminator is also responsible for distinguishing the artificial data produced by the generator and the real data, as well as separating them from each other. Throughout the training phase, the pretrained transformer undergoes supervised fine-tuning to enhance its capability in predicting the electricity consumption of electrical appliances. Here, the encoder–decoder structure along with the attention mechanism are used to extract the features in the cumulative signal of electricity consumption. Figure 3 also shows the ELECTRICity network structure.

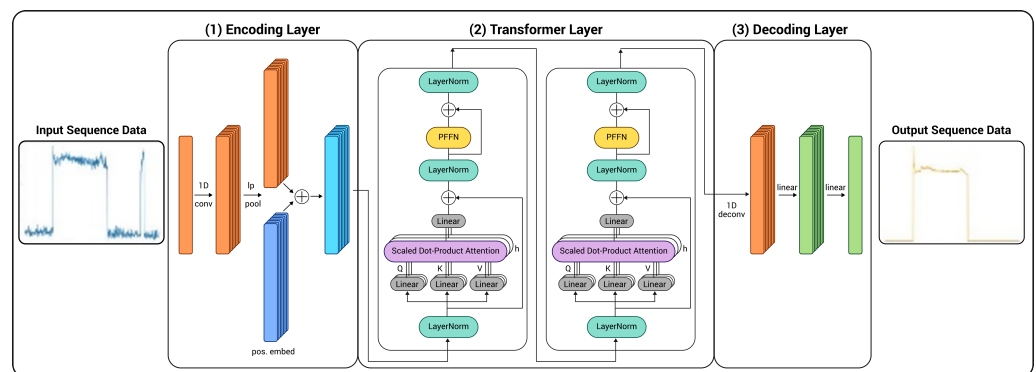


Figure 3. ELECTRICity network structure [19].

2.5. The Deep Dilated Residual Network Method

Like the previous articles, the aim of the article [20] is to separate the consumption of electrical appliances based on the amount of consumption of the whole household. The data used in the article are WikiEnergy and UK-DALE datasets. The WikiEnergy dataset encompasses information on the power consumption of over 600 households in Beijing, measured at 60-second intervals. The architecture considered for the paper is a combination of ResNet and dialed convolution network architecture. To solve the gradient vanishing problem, a network called a residual network is proposed. The difference between this network and normal networks is that it has a shortcut connection that passes through one or more layers and does not consider them; actually, it takes a shortcut and connects one layer to a further layer. The presence of this connection implies that a value of 1 is added to each of the coefficients of the primary derivatives, preventing them from diminishing regularly. Figures 4 and 5 show the structure of the residual layer and the ResNet network, respectively.

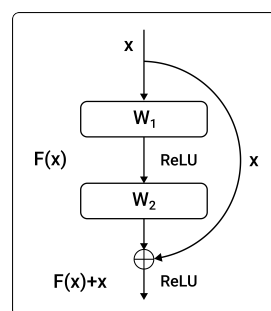


Figure 4. The structure of the residual layer. Adapted with permission from Ref. [20].

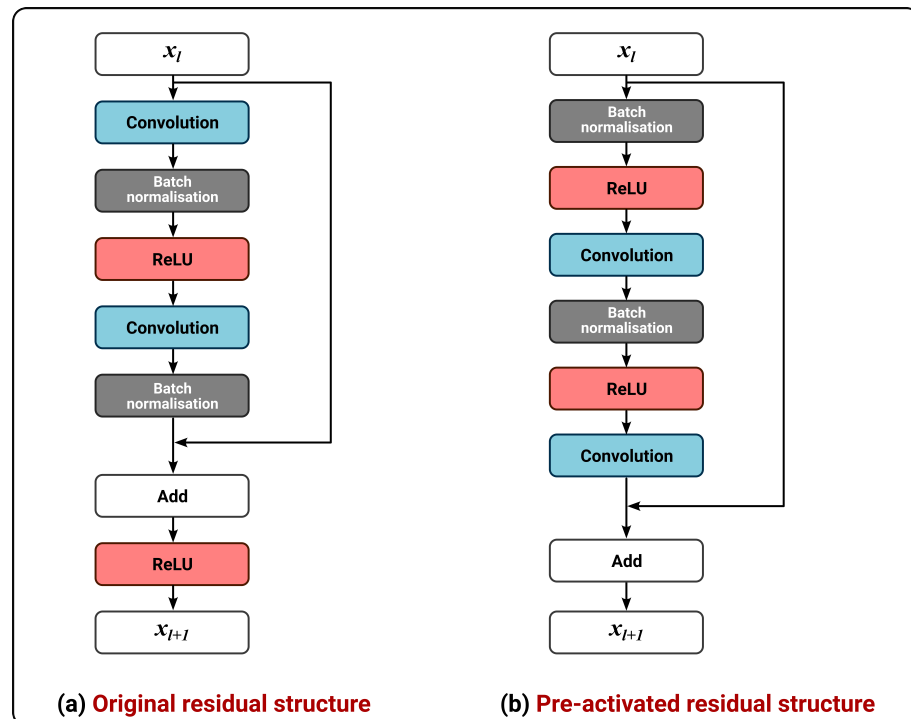


Figure 5. ResNet network structure. (a) Original, and (b) Pre-activated residual structure. Adapted with permission from Ref. [20].

In the main architecture of the ResNet network, the order of the layers is as in (a). The problem in this architecture is that the presence of the ReLU activator function at the end of the block makes the output of each block always non-negative, and as a result, during training, the output of the middle blocks becomes larger and larger regularly, while the output of each block should be a value between $(-\infty, +\infty)$ so that the range of changes of the final outputs is not too far from the range of inputs. For this reason, the structure of the block has been changed to structure (b). In this structure, the same layers as in structure (a) exist, but they are used in a different order, and the ReLU layer is located at the end of the block. As a result, the output of the block can be negative or positive. The network used in the above article is of the ResNet type, which uses dilated convolution in its convolutional layers. The details of the network are also stated in Table 1.

Table 1. D-ResNet network details [20].

Residual Block	Number of Residual Units	Number of Convolutional Layers in the Block	Dilation Rate
1	3	30	1
2	4	40	2
3	6	50	3
4	3	50	3

2.6. The Attention-Based Method

The objective of the article [21] is to disaggregate the power consumption of household electrical appliances. However, the methodology employed in the article diverges from previous studies, incorporating an attention mechanism. The data used in the article are the REDD and UK-DALE datasets. The network consists of two blocks: classifier and regression. The classifier block includes six convolutional layers, one fully connected layer of a length of 1024, and the ReLU activation function. The regression block also includes four convolutional layers, a bilateral block, an attention layer, and a fully connected layer.

Finally, the output of the network is obtained by multiplying the output of the classifier block and the regression block.

3. Proposed Method

This article presents a novel approach for NILM by leveraging a model incorporating attention mechanism, temporal pooling, residual connections, and transformer architecture to accurately discern energy consumption patterns of individual household appliances, which are explained in detail in [22] and this section. The proposed method entails a sequence of layers, including encoders, transformers, attention, temporal pooling, and residual connections, offering a comprehensive solution for NILM while effectively capturing appliance-specific energy usage in a household. The components of the proposed method are discussed in the following subsections:

3.1. Attention Mechanism

An attention mechanism [23] is a technique used in deep learning models to improve the performance of sequence-to-sequence models. This mechanism enables the model to selectively focus on parts of the input sequence that are most relevant to the current time step, rather than processing the entire sequence at once. An attention mechanism is used to enable the model to focus on significant parts of the input sequence. This mechanism allows the model to pay more attention to the important parts of the input and ignore the irrelevant parts.

3.2. Temporal Pooling

Temporal pooling [24] is a technique used in machine learning and computer vision to extract useful information from sequential data such as video or speech signals. This method allows the model to work with fixed-size inputs (by summarizing the information in the sequence) in a compact representation. In NILM, temporal pooling is used to sample the input sequence and provides the possibility of effective management of longer sequences. This approach enables the model to extract relevant features from the input sequence while reducing the computational complexity.

3.3. Residual Connection

A residual connection [25], also known as a skip connection, is a type of connection used in deep neural networks. It involves connecting the output of one layer to the input of the next layer, bypassing one or more layers in between. The purpose of a residual connection is to make the gradient flow more smoothly in the network during training. This can help avoid the vanishing gradient problem that can occur in deep networks. In the context of NILM, residual connection can be used to improve the performance of deep learning models by allowing them to learn the residual power consumption of each device. This residual electricity consumption can be used to estimate the energy consumption of individual appliances, which can improve the accuracy of the overall NILM system.

3.4. Transformers

A transformer is a type of neural network architecture that has become increasingly popular in NLP tasks such as language translation, text classification, and language modeling. The transformer architecture is based on the self-attention mechanism, which allows the model to evaluate the importance of different words in a sentence when making a prediction. It also consists of an encoder and a decoder, each of which consists of several layers of self-attention and feed-forward neural networks. In the encoder, the self-attention mechanism allows the model to compute a representation of each word in the sentence with respect to other words in the sentence. The decoder uses this representation to produce a translation of the input sentence in the target language. Recently, transformers have been applied in the field of NILM to separate household power consumption into individual appliances.

In general, it can be said that the proposed model is a comprehensive solution to the NILM problem and includes a variety of innovative techniques that enable it to identify energy consumption patterns of personal appliances in a household.

3.5. Architecture of the Proposed Method

The overall model is initially composed of 4 encoder layers that are brought together. The final output of these 4 encoder layers is given as input to a fully connected layer. In the next step, the output of this layer is given to a transformer, and the tensor obtained from this transformer layer along with the output of the encoder layer is given as input to the attention layer. Then, the output from the attention layer is added to the output of the final encoder, and the first jump connection of the model is applied here. Then, this obtained output enters 4 consecutive blocks of temporal pooling (TP layer), and finally, the outputs obtained from these 4 blocks are glued together. Then, as a jump connection, the tensor resulting from pasting the output of these blocks is added with the output of the fully connected layer (FC layer), and in the final step, the tensor resulting from this addition is given as input to the decoder and passes through a convolution layer after that. Figure 6 shows the general structure of the model.

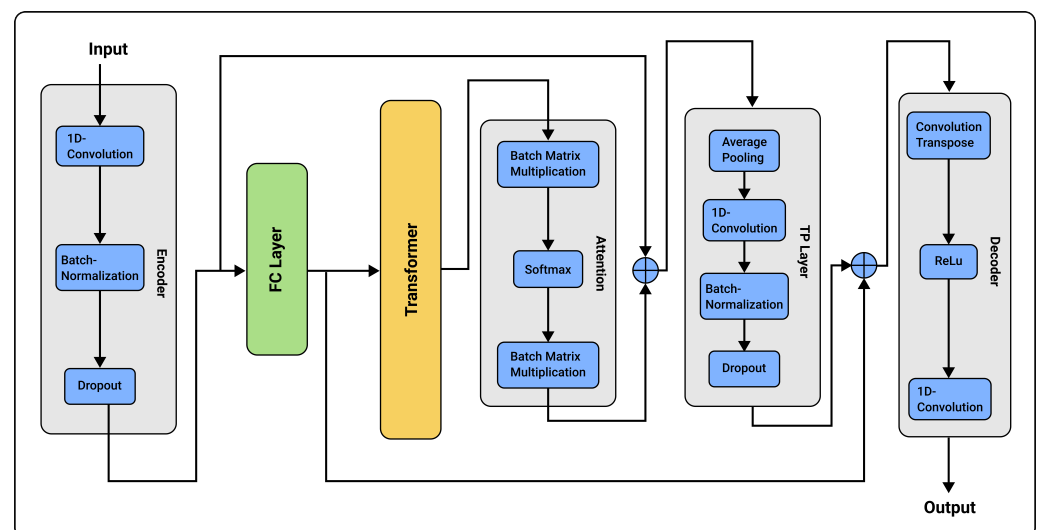


Figure 6. Structure of the proposed model.

3.6. Data Preprocessing

The consumption data of each device and the total number of houses in the dataset used in this article were preprocessed before being processed by the neural network. The preprocessing performed in this research is similar to the preprocessing performed in [26]. Also, the DS CleanerPython library [27] was used for preprocessing, cleaning, and converting time series data into a standard file format. This library also has a function for resampling datasets. In general, the preprocessing performed in this article can be summarized as follows.

3.7. Removal of Excessively High Powers

Measuring devices have errors and sometimes record too much power. We considered a maximum power for each device, and powers higher than that value are removed.

3.8. Changing the Sampling Interval from 6 s to 1 min

The power consumed at any moment will be equal to the average power measured in the previous minute.

3.9. Removing the Meter Error

If a device is off and turns on for a short period of time, or if it is on and turns off for a short period of time, that short off/on period is not considered.

3.10. Forming the off/on Data Set

The main dataset includes the power consumption of the devices. Based on the values of the exponents, we form a new binary dataset that indicates whether the device was on (one) or off (zero) at any given moment in time.

3.11. Total Power Normalization

The total power values of all devices are divided by 2000 (watts) so that their value is normal. Also, the average power value is subtracted from its original value to make the average signal zero, because it is more desirable to use small values with zero average for input in neural network training.

3.12. Network Input and Output

The inputs of the network are the consumption data of individual devices and the total consumption data of the house, which are preprocessed before being processed by the neural network. The network outputs are also an estimate of the activation status of the equipment for each moment, which are obtained through the classification of several classes of active loads at the same time. It can be said that each input contains 510 consecutive samples from the training set (1×510), and the corresponding output contains 480 activation status for each device (480×3).

3.13. Simulation and Testing Environment

In this research, the Python programming language and the Torch, Pandas, NumPy, Scikit-Learn, Matplotlib, Math, etc., libraries were used in a Google Colab environment to train and test the proposed model. The Google Colab platform offers the added benefit of running tests in a virtual environment, which makes code management easier. In addition, Google Colab makes use of Google's high-performance computing resources without the need for expensive hardware. In our experiments, we utilized Google Colab with the following specifications: 12 GB of RAM, a T4 GPU with 16 GB of memory, and a 78 GB hard disk.

In a real-world scenario, both edge-based and cloud-based approaches can be utilized to implement this solution. However, choosing a cloud implementation is more practical when striving to uphold the solution's cost-effectiveness.

3.14. Computational Complexity

The computational complexity of the proposed method depends on the number of operations within its architecture compared with other deep learning models. Transformer-based models are recognized for their self-attention mechanisms, introducing a quadratic dependency on sequence length and leading to increased computational complexity in sequence processing. Nevertheless, the inclusion of a temporal pooling component in the proposed method enhances the efficiency of temporal feature processing. In contrast to traditional models like recurrent neural networks (RNNs) or long short-term Memory (LSTM) networks, which exhibit sequential dependencies and can be computationally intensive, the transformer-based model offers advantages in parallelization due to its attention mechanism. It is important to note that there exists a trade-off between computational burden and the accuracy achieved by the algorithms.

4. Datasets and Evaluation Methods

In this section, we first introduce the publicly available NILM datasets. Then, performance criteria to evaluate the NILM methods will be discussed.

4.1. NILM Datasets

In the research community, many NILM datasets have been made publicly available for developing NILM algorithms and benchmarking their results and performance in the public domain [15]. Each dataset has its own characteristics due to changes in devices that were controlled in different periods of time in certain environments or buildings [28]. Table 2 lists popular NILM datasets that are publicly available for research purposes (some of which are also used in the described methods).

Table 2. Datasets publicly accessible for the development of NILM algorithms.

Dataset	Sampling Rate/Interval	Duration	Country
UK-DALE	16 kHz	2 years	UK
REFIT	8 s	2 years	UK
REDD	16.5 kHz	19 days	US
BLUED	12 kHz	1 week	US
Dataport	1 Hz	+4 years	US
AMPds	1 min	2 years	Canada
COMBED	30 s	1 month	India
PLAID	30 kHz	5 s	US

In this article, UK-DALE [29], REDD [30], and REFIT [31] datasets are used. The UK-DALE dataset is a widely used dataset in NILM research, which stands for “UK Domestic Appliance-Level Electricity”. This collection contains energy consumption data collected from a group of UK households. The REDD dataset, which stands for “Reference Energy Disaggregation Data Set”, is also a famous dataset used for research related to NILM. This dataset contains high-frequency energy consumption data collected from various Canadian households and includes a wide range of different appliances and devices such as lights, refrigerators, air conditioners, etc. The REFIT dataset also includes nine individual device measurements at 8 s intervals per house collected from 20 houses. In this article, the results pertaining to three types of electrical appliances, namely “Refrigerator,” “Washing machine,” and “Dishwasher,” are analyzed within these datasets.

4.2. Evaluation Metrics

A confusion matrix is used to obtain a more comprehensive picture in evaluating the performance of the model. This matrix is an $N \times N$ matrix used to evaluate the performance of a classification model, where N is the number of target classes. This matrix compares the actual target values with the values predicted by the machine learning model. Figure 7 shows the confusion matrix.

The following four terms constitute the fundamental terminology that will assist us in discerning the metrics used for evaluation [32]:

True positives (TP): When the actual value is positive and the predicted value is also positive.

True negatives (TN): When the real value is negative and the prediction is negative.

False positives (FP): When the actual is negative but the prediction is positive. Also known as a type 1 error.

False negatives (FN): When the actual is positive but the prediction is negative. Also known as a type 2 error.

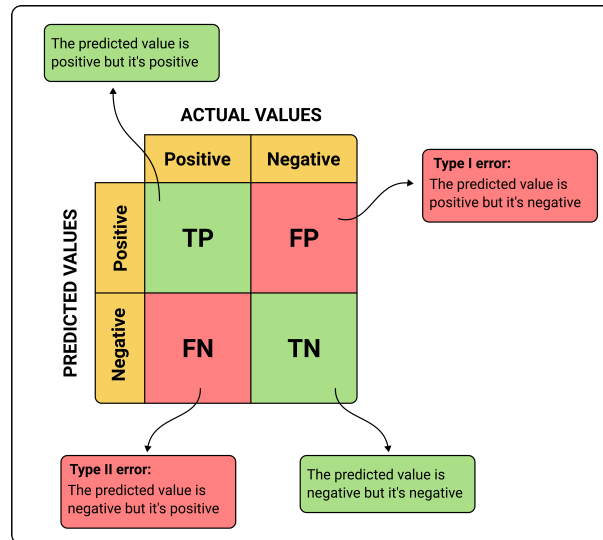


Figure 7. Confusion matrix for binary classification.

There are measures other than the confusion matrix that can help achieve better understanding and analysis of the models and their performance.

4.3. Accuracy

Accuracy is a common measure in classification that measures the overall accuracy of the predictions made by a classification model by comparing them with the real labels of the samples. This measure is calculated using the following formula:

$$Accuracy = \frac{(\text{True Positives} + \text{True Negatives})}{(\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives})} \quad (1)$$

4.4. Precision

It is a performance measure that evaluates the accuracy of positive predictions made by a classification model. This measure is one measures the proportion of correctly predicted positives out of all predicted positives. Precision is often used in conjunction with other evaluation metrics, such as recall and F1-score, to provide a comprehensive analysis of a model's performance. This criterion is calculated using the following formula:

$$Precision = \frac{\text{True Positives}}{(\text{True Positives} + \text{False Positives})} \quad (2)$$

4.5. Recall

Recall, also known as sensitivity or the true positive rate, is a performance measure that evaluates the ability of a classification model to correctly identify positive samples. This metric measures the proportion of true positives that are correctly predicted out of all true positives. Recall is calculated using the following formula:

$$Recall = \frac{\text{True Positives}}{(\text{True Positives} + \text{False Negatives})} \quad (3)$$

4.6. F1-Score

The F1-score is a commonly used performance measure in classification problems, especially when dealing with unbalanced datasets. This measure combines the precision and

recall measures into a single value to provide a balanced measure of a model's performance. The F1-score is calculated using the following formula:

$$\text{F1-Score} = \frac{2 \times (\text{precision} \times \text{recall})}{\text{precision} + \text{recall}} \quad (4)$$

4.7. MCC

The MCC is commonly used in machine learning to evaluate binary classification models. To provide a function of a model's performance, especially in situations where the dataset is unbalanced, it considers positives, true negatives, false positives, and false negatives. Its formula is given below:

$$\text{MCC} = \frac{(\text{TP} \times \text{TN}) - (\text{FP} \times \text{FN})}{\sqrt{(\text{TP} + \text{FP}) \times (\text{TP} + \text{FN}) \times (\text{TN} + \text{FP}) \times (\text{TN} + \text{FN})}} \quad (5)$$

4.8. MAE

MAE stands for mean absolute error, which is a common measure to evaluate the accuracy of regression models. This measure measures the average absolute difference between the predicted values and the actual values of the target variable. This criterion is calculated using the following formula [33]:

$$\text{MAE} = \left(\frac{1}{n}\right) \times \sum |y_i - \hat{y}_i| \quad (6)$$

4.9. SAE

As stated earlier, SAE or sum absolute error, is measure to evaluate the accuracy of regression models. Instead of averaging the absolute differences between the predicted and actual values, SAE sums these absolute differences for all cases (according to the formula below).

$$\text{SAE} = \sum |y_i - \hat{y}_i| \quad (7)$$

5. Experiments and Results

To evaluate the effectiveness of the proposed model, a simple LSTM block model with temporal pooling is tested first. The results of this initial test serve as a basis for comparison with the proposed model. In the next step, a more advanced model is designed based on the identified key elements. The results of the experiments show that the proposed model works better than the basic model. In general, the results of this study show the importance of combining key elements such as an attention mechanism, temporal pooling, residual connection, and transformers in NILM models. These elements allow the model to better capture the complex temporal patterns and dependencies of energy consumption data, ultimately leading to improved performance and accuracy. Tables 3 and 4 show the results of network training in a seen and unseen case.

Table 3. The results of network training in a seen case.

Model	Metrics/Appliances	F1	Precision	Recall	Acc	MCC	MAE	SAE
LSTM	Fridge	0.883	0.891	0.874	0.894	0.787	13.94	−0.02
	Dishwasher	0.922	0.913	0.933	0.996	0.92	20.99	0.004
	Washing machine	0.979	0.976	0.983	0.997	0.978	41.89	−0.076
Proposed model	Fridge	0.886	0.892	0.88	0.897	0.79	13.85	−0.018
	Dishwasher	0.925	0.926	0.925	0.996	0.923	20.58	−0.017
	Washing machine	0.978	0.975	0.982	0.997	0.978	42.02	−0.74

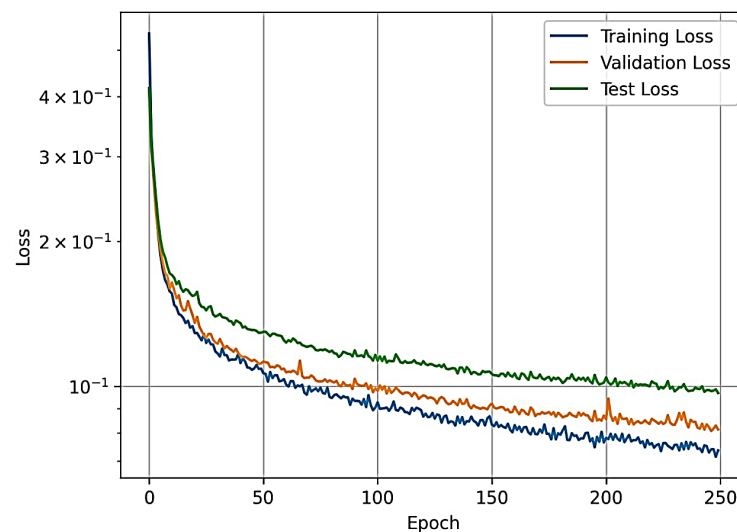
Table 4. The results of network training in an unseen case.

Model	Metrics/Appliances	F1	Precision	Recall	Acc	MCC	MAE	SAE
LSTM	Fridge	0.878	0.895	0.859	0.907	0.795	16.99	−0.041
	Dishwasher	0.816	0.798	0.939	0.99	0.797	32.99	0.02
	Washing machine	0.859	0.843	0.956	0.996	0.844	8.53	0.012
Proposed model	Fridge	0.876	0.891	0.862	0.908	0.802	16.8	−0.038
	Dishwasher	0.849	0.803	0.901	0.993	0.809	30.24	0.061
	Washing machine	0.857	0.842	0.874	0.997	0.848	8.26	0.029

5.1. Model Training

In the first step, the UK-DALE dataset is used to train the network. The designed model is trained using the data of houses 1 and 5 of this dataset, while the performance check is performed on the whole dataset of house 2. The training and testing periods are completely separated. The network is trained on homes other than the one tested (the unseen case) to evaluate the model's ability to generalize and recognize general features of a type of home appliance. This approach is based on multiclass classification of simultaneous active loads, and it estimates appliance consumption as a constant average value during activation.

Network parameters are optimized by the gradient descent method using Adam's optimization algorithm, with a learning rate of 10^{-4} and a batch size of 32. For both seen and unseen modes, training is planned for 250 epochs. Figures 8 and 9 show the plots of loss by epoch during network training for each of the two seen and unseen cases.

**Figure 8.** Plot of loss by epoch in the seen case in the UK-DALE dataset.

In the following, to check the performance of the designed model on other datasets, this model was trained on REDD and REFIT datasets, and the loss plots by epoch in the seen and unseen cases in REDD and REFIT datasets are shown in Figures 10–13.

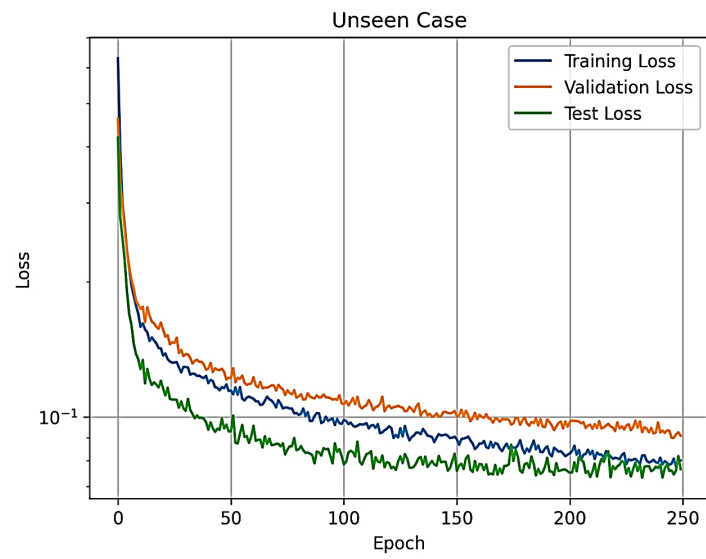


Figure 9. Plot of loss by epoch in the unseen case in the UK-DALE dataset.

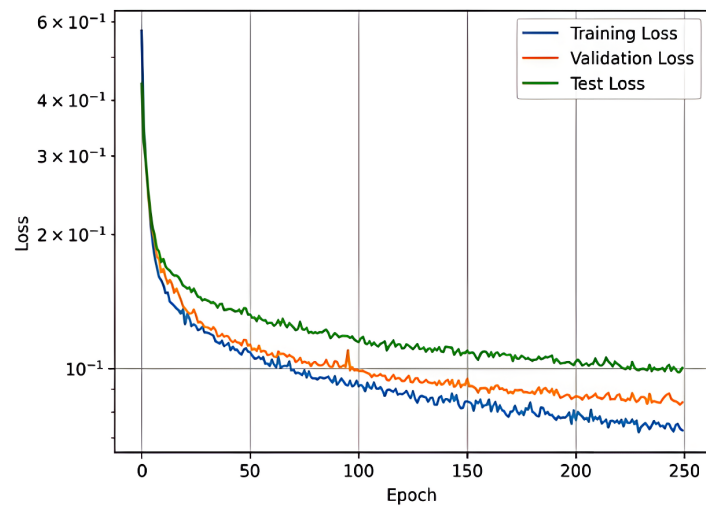


Figure 10. Plot of loss by epoch in the seen case in the REDD dataset.

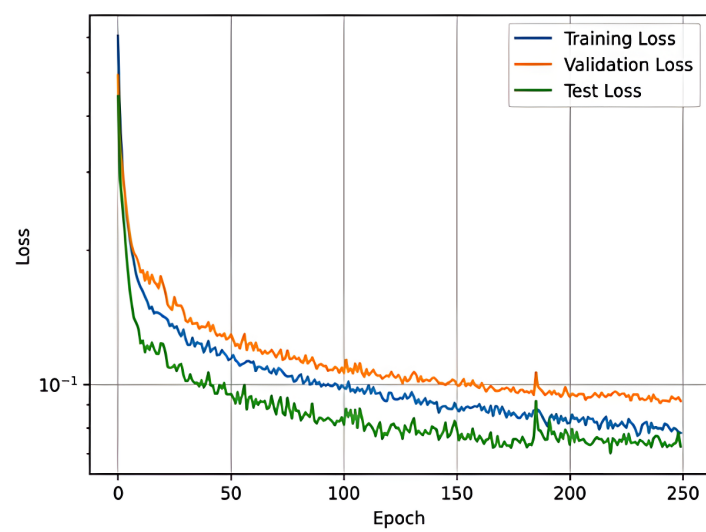


Figure 11. Plot of loss by epoch in the unseen case in the REDD dataset.

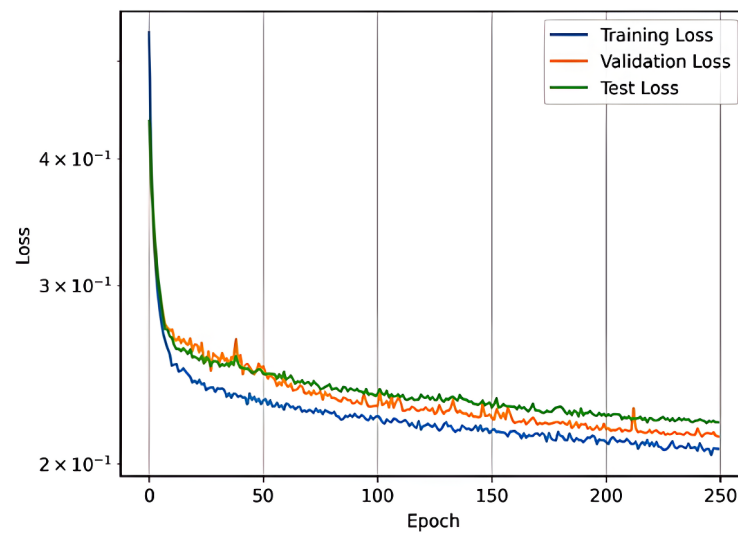


Figure 12. Plot of loss by epoch in the seen case in the REFIT dataset.

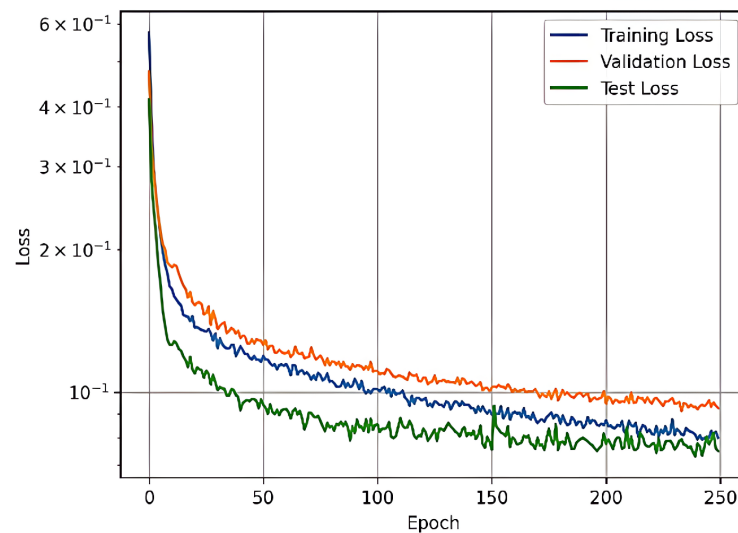


Figure 13. Plot of loss by epoch in the unseen case in the REFIT dataset.

The results of the model designed on UK-DALE, REDD, and REFIT datasets are shown in Tables 5–7. It can be seen that the results obtained on the REDD and REFIT datasets are relatively less accurate than the results obtained on the UK-DALE data set.

Table 5. The results of the proposed model in the seen and unseen modes on the UK-DALE dataset.

State	Appliances/Metrics	F1	Precision	Recall	Acc	MCC	MAE	SAE
Seen	Fridge	0.886	0.892	0.88	0.897	0.79	13.85	−0.018
	Dishwasher	0.925	0.926	0.925	0.996	0.923	20.58	−0.017
	Washing machine	0.978	0.975	0.982	0.997	0.978	42.02	−0.074
Unseen	Fridge	0.876	0.891	0.862	0.908	0.802	16.8	−0.038
	Dishwasher	0.849	0.803	0.901	0.993	0.809	30.24	0.061
	Washing machine	0.857	0.842	0.874	0.997	0.848	8.26	0.029

Table 6. The results of the proposed model in the seen and unseen modes on the REDD dataset.

State	Appliances/Metrics	F1	Precision	Recall	Acc	MCC	MAE	SAE
Seen	Fridge	0.877	0.882	0.874	0.855	0.768	14.38	−0.02
	Dishwasher	0.917	0.917	0.918	0.993	0.917	22.75	−0.021
	Washer–dryer	0.973	0.971	0.976	0.993	0.962	45.11	0.078
Unseen	Fridge	0.87	0.888	0.854	0.899	0.792	16.91	0.043
	Dishwasher	0.845	0.8	0.896	0.991	0.803	30.44	−0.072
	Washer–dryer	0.855	0.84	0.872	0.996	0.845	8.39	0.038

Table 7. The results of the proposed model in the seen and unseen modes on the REFIT dataset.

State	Appliances/Metrics	F1	Precision	Recall	Acc	MCC	MAE	SAE
Seen	Fridge	0.833	0.841	0.826	0.848	0.735	16.83	0.036
	Dishwasher	0.908	0.905	0.911	0.993	0.908	23.26	−0.024
	Washing machine	0.97	0.971	0.97	0.995	0.966	45.74	0.089
Unseen	Fridge	0.868	0.888	0.849	0.896	0.795	18.66	0.054
	Dishwasher	0.844	0.797	0.897	0.99	0.801	32.52	−0.073
	Washing machine	0.854	0.84	0.869	0.995	0.839	9.02	−0.038

5.2. Comparison of Results of the Proposed Model with the Previous Studies

In this section, the results of the proposed model for solving the NILM problem are compared with the results obtained from previous studies. The purpose of this comparison is to evaluate the effectiveness of the proposed model and determine whether it has a better performance than other existing methods. To achieve this goal, the results of the proposed model are compared with the results of seven introduced articles. In Table 8, the results of the different methods mentioned before for dishwasher (DW), fridge–freezer (FR), kettle (KE), microwave (MW), and washing machine (WM) are compared. It is important to note that the criteria used in these articles may differ, making direct comparisons challenging. As a result, some criteria are not included in the table for an overall comparison of all methods. Looking at the row related to the presented method, it can be seen that the proposed model reaches an F1-core of 92.96 and SAE equal to −0.036, which is better than the results obtained from other methods. The value of the F1-score of the proposed model shows that the model is able to achieve a balance between accuracy and recall, which is important for accurately identifying home appliances and their electricity consumption. The negative SAE score also shows that the proposed model is able to estimate the electricity consumption of each home appliance with high accuracy without underestimating the actual electricity consumption. As a result, the presented model shows superior performance in the accurate estimation of the electricity consumption of individual household appliances compared to previous methods in the literature.

5.3. The Results of Different Data Resolutions

In the following, the proposed model was tested in seen and unseen cases on the UK-DALE dataset with a resolution of 2 min and a resolution of 30 s. From Tables 9 and 10, it can be seen that by reducing the frequency of data collection from 30 s to two minutes, the accuracy of the model also decreased. Also, by reducing the sampling frequency, despite the increase in the volume of network input data, the results did not change much.

Table 8. Overall comparison of the proposed method with previous methods.

Model	Metric	DW	FR	KE	MW	WM	Overall
WaveNILM	Acc	-	-	-	-	-	94.7
	MAE	-	-	-	-	-	-
	SAE	-	-	-	-	-	-
	F1 (%)	-	-	-	-	-	-
VAE-NILM	Acc	-	-	-	-	-	-
	MAE	23.4	21.6	22.1	10.8	6.7	16.9
	SAE	-	-	-	-	-	-
	F1 (%)	32.1	80.6	73.5	64.6	87.1	67.6
COLD	Acc	-	-	-	-	-	-
	MAE	-	-	-	-	-	-
	SAE	-	-	-	-	-	-
	F1 (%)	-	-	-	-	-	94.55
ELECTRIcity	Acc	98.4	84.3	99.9	99.6	99.4	96.32
	MAE	18.96	22.61	9.26	6.28	3.65	12.152
	SAE	-	-	-	-	-	-
	F1 (%)	81.8	81.0	93.9	27.7	79.7	72.82
D-ResNet	Acc	98.8	99.6	99.8	100	99.6	99.56
	MAE	7.8	2.627	2.518	1.505	2.966	3.48
	SAE	0.010	0.020	0.024	0.162	0.072	0.0576
	F1 (%)	79.6	99.4	85.9	97.8	82.6	89.06
LDwA	Acc	-	-	-	-	-	-
	MAE	6.57	13.24	5.69	3.79	7.26	7.31
	SAE	3.91	6.02	3.74	2.98	4.87	4.30
	F1 (%)	68.99	87.01	99.81	67.55	71.94	79.06
Proposed model	Acc	99.6	89.7	-	-	99.7	96.33
	MAE	20.58	13.85	-	-	40.02	24.82
	SAE	-0.017	-0.018	-	-	-0.074	-0.036
	F1 (%)	92.5	88.6	-	-	97.8	92.96

Table 9. The results of the proposed model in the seen and unseen cases on the UK-DALE dataset with a resolution of 2 min.

State	Appliances/Metrics	F1	Precision	Recall	Acc	MCC	MAE	SAE
Seen	Fridge	0.829	0.846	0.814	0.831	0.725	18.94	0.062
	Dishwasher	0.908	0.916	0.902	0.992	0.909	23.87	-0.036
	Washing machine	0.972	0.966	0.969	0.995	0.953	45.18	-0.089
Unseen	Fridge	0.857	0.854	0.862	0.897	0.784	18.68	0.058
	Dishwasher	0.834	0.796	0.876	0.991	0.791	36.85	0.084
	Washing machine	0.845	0.829	0.863	0.994	0.813	11.28	0.033

Table 10. The results of the proposed model in the seen and unseen cases on the UK-DALE dataset with a resolution of 30 s.

State	Appliances/Metrics	F1	Precision	Recall	Acc	MCC	MAE	SAE
Seen	Fridge	0.889	0.895	0.884	0.898	0.793	13.69	-0.017
	Dishwasher	0.92	0.922	0.918	0.995	0.919	20.74	-0.017
	Washing machine	0.974	0.968	0.981	0.997	0.974	42.11	-0.078
Unseen	Fridge	0.886	0.899	0.875	0.912	0.816	16.63	0.031
	Dishwasher	0.838	0.789	0.894	0.99	0.798	31.3	0.072
	Washing machine	0.85	0.836	0.864	0.996	0.842	8.18	0.032

6. Conclusions and Future Works

This article addressed the challenge of nonintrusive load monitoring (NILM) by proposing an advanced Seq2Seq model combined with a transformative approach. The primary objective was to accurately disaggregate household energy consumption into individual appliances. To tackle this, the methodology incorporated key innovations: an attention mechanism, temporal pooling, residual connection, and transformers. These techniques collectively enabled the model to focus on relevant segments of the input sequence, manage longer sequences effectively, facilitate smoother gradient flow, and leverage the self-attention mechanism for precise data representation. The comparison in Table 8 shows that the proposed model is able to improve the results compared to previous methods. The proposed model is compared with previous models in terms of various criteria such as accuracy, F1-score, precision, and recall. In addition, the proposed model is able to overcome some of the limitations of previous models, such as their inability to accurately identify the consumption patterns of individual household appliances in a household. The proposed model is able to achieve better accuracy and precision in identifying each device. From Table 8, it can be seen that the presented model is able to improve the results in general.

Finally, considering the findings and limitations of the current research, recommendations for future studies are presented:

- This model can be expanded to add additional features such as time, day, weather conditions, etc., to increase the accuracy of device detection and energy consumption estimation.
- The proposed model can be tested on a larger and more diverse dataset to further evaluate its effectiveness and generalization in practical settings where various factors such as noise, interference, and data quality can affect its accuracy.
- The effect of different metaparameters on model performance can be investigated to identify the optimal configuration for specific datasets and scenarios.
- The proposed model can be extended to multitask learning settings to simultaneously perform other related tasks such as device identification.

Overall, these future research directions can help advance the field of NILM and contribute to the development of more accurate and efficient energy management systems.

Author Contributions: Conceptualization, M.I.A., R.R. and A.E.; Methodology, M.I.A. and R.R.; Software, M.I.A.; Validation, R.R. and A.E.; Formal analysis, R.R. and A.E.; Data curation, M.I.A.; Writing—original draft, M.I.A.; Writing—review & editing, R.R. and A.E.; Supervision, R.R. and A.E.; Funding acquisition, A.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data analyzed in this study are openly available in UK-DALE at <http://doi.org//10.1038//sdata.2015.7>, [29], [REDD] at <https://tokhub.github.io//dbecd//links//redd.html>, [30], and [REFIT] at <https://doi.org//10.17028//rd.lboro.2070091.v1>, [31].

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Seen	In this case, part of the test and training data are the same.
Unseen	In this case, different data are used for testing and training.
SAE	Signal aggregate error is a variant of mean absolute error (MAE) that is used to evaluate the accuracy of regression models.
ReLU	Rectified linear unit.
SNS	Synthesizer of normalized signatures.
COLD	Concurrent loads disaggregator.
STFT	Short time Fourier transform.
DS Cleaner	Dataset Cleaner.

References

1. Rajabi, R.; Estebarsari, A. Deep Learning Based Forecasting of Individual Residential Loads Using Recurrence Plots. In Proceedings of the 2019 IEEE Milan PowerTech, Milan, Italy, 23–27 June 2019; pp. 1–5.
2. Scott, J.; Bernheim Brush, A.; Krumm, J.; Meyers, B.; Hazas, M.; Hodges, S.; Villar, N. PreHeat: Controlling home heating using occupancy prediction. In Proceedings of the 13th International Conference on Ubiquitous Computing, Beijing, China, 17–21 September 2011; pp. 281–290.
3. Athanasiadis, C.; Doukas, D.; Papadopoulos, T.; Chrysopoulos, A. A Scalable Real-Time Non-Intrusive Load Monitoring System for the Estimation of Household Appliance Power Consumption. *Energies* **2021**, *14*, 767. [[CrossRef](#)]
4. Parson, O.; Ghosh, S.; Weal, M.; Rogers, A. Non-intrusive load monitoring using prior models of general appliance types. In Proceedings of the AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012; Volume 26, pp. 356–362.
5. Parson, O.; Ghosh, S.; Weal, M.; Rogers, A. An unsupervised training method for non-intrusive appliance load monitoring. *Artif. Intell.* **2014**, *217*, 1–19.
6. Kolter, J.Z.; Jaakkola, T. Approximate inference in additive factorial hmms with application to energy disaggregation. In Proceedings of the Artificial Intelligence and Statistics, PMLR, La Palma, Spain, 21–23 April 2012; pp. 1472–1482.
7. Ng, Y.C.; Chilinski, P.M.; Silva, R. Scaling factorial hidden markov models: Stochastic variational inference without messages. In Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 5–10 December 2016; pp. 1–9.
8. Gonçalves, H.; Ocleanu, A.; Bergés, M.; Fan, R. Unsupervised disaggregation of appliances using aggregated consumption data. In Proceedings of the 1st KDD Workshop on Data Mining Applications in Sustainability (SustKDD), San Diego, CA, USA, 21 August 2011; ACM: New York, NY, USA, 2011.
9. Zhao, B.; Stankovic, L.; Stankovic, V. On a training-less solution for non-intrusive appliance load monitoring using graph signal processing. *IEEE Access* **2016**, *4*, 1784–1799.
10. Zhuang, M.; Shahidehpour, M.; Li, Z. An overview of non-intrusive load monitoring: Approaches, business applications, and challenges. In Proceedings of the 2018 International Conference on Power System Technology (POWERCON), Guangzhou, China, 6–8 November 2018; pp. 4291–4299.
11. 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, Seoul, Republic of Korea, 4–5 November 2015; pp. 55–64.
12. Mauch, L.; Yang, B. A new approach for supervised power disaggregation by using a deep recurrent LSTM network. In Proceedings of the 2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Orlando, FL, USA, 14–16 December 2015; pp. 63–67.
13. Shin, C.; Joo, S.; Yim, J.; Lee, H.; Moon, T.; Rhee, W. Subtask gated networks for non-intrusive load monitoring. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 1150–1157.
14. 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.
15. Azad, M.I.; Rajabi, R.; Estebarsari, A. Non-Intrusive Load Monitoring (NILM) using Deep Neural Networks: A Review. In Proceedings of the 2023 IEEE International Conference on Environment and Electrical Engineering and 2023 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), Madrid, Spain, 6–9 June 2023; pp. 1–6.
16. Harell, A.; Makonin, S.; Bajić, I.V. Wavenilm: A causal neural network for power disaggregation from the complex power signal. In Proceedings of the ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 8335–8339.
17. Langevin, A.; Carbonneau, M.A.; Cheriet, M.; Gagnon, G. Energy disaggregation using variational autoencoders. *Energy Build.* **2022**, *254*, 111623. [[CrossRef](#)]
18. Kamyshev, I.; Kriukov, D.; Gryazina, E. Cold: Concurrent loads disaggregator for non-intrusive load monitoring. *arXiv* **2021**, arXiv:2106.02352.
19. Sykiotis, S.; Kaselimi, M.; Doulamis, A.; Doulamis, N. Electricity: An efficient transformer for non-intrusive load monitoring. *Sensors* **2022**, *22*, 2926. [[CrossRef](#)] [[PubMed](#)]
20. Xia, M.; Liu, W.; Wang, K.; Zhang, X.; Xu, Y. Non-intrusive load disaggregation based on deep dilated residual network. *Electr. Power Syst. Res.* **2019**, *170*, 277–285.
21. Piccialli, V.; Sudoso, A.M. Improving non-intrusive load disaggregation through an attention-based deep neural network. *Energies* **2021**, *14*, 847. [[CrossRef](#)]
22. Azad, M.I.; Rajabi, R.; Estebarsari, A. Sequence-to-Sequence Model with Transformer-based Attention Mechanism and Temporal Pooling for Non-Intrusive Load Monitoring. In Proceedings of the 2023 IEEE International Conference on Environment and Electrical Engineering and 2023 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), Madrid, Spain, 6–9 June 2023; pp. 1–5.
23. 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. 1–11.

24. Lazebnik, S.; Schmid, C.; Ponce, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; Volume 2, pp. 2169–2178.
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
26. Massidda, L.; Marrocu, M.; Manca, S. Non-intrusive load disaggregation by convolutional neural network and multilabel classification. *Appl. Sci.* **2020**, *10*, 1454. [[CrossRef](#)]
27. Pereira, M.; Velosa, N.; Pereira, L. dsCleaner: A Python Library to Clean, Preprocess and Convert Non-Intrusive Load Monitoring Datasets. *Data* **2019**, *4*, 123. [[CrossRef](#)]
28. Batra, N.; Singh, A.; Singh, P.; Dutta, H.; Sarangan, V.; Srivastava, M. Data driven energy efficiency in buildings. *arXiv* **2014**, arXiv:1404.7227.
29. 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. [[PubMed](#)]
30. Kolter, J.Z.; Johnson, M.J. REDD: A public data set for energy disaggregation research. In Proceedings of the Workshop on Data Mining Applications in Sustainability (SIGKDD), San Diego, CA, USA, 21–24 August 2011; Volume 25, pp. 59–62.
31. Firth, S.; Kane, T.; Dimitriou, V.; Hassan, T.; Fouchal, F.; Coleman, M.; Webb, L. *REFIT Smart Home Dataset*; Loughborough University: Loughborough, UK, 2017. [[CrossRef](#)]
32. Theodoridis, S.; Koutroumbas, K. Chapter 10—Supervised Learning: The Epilogue. In *Pattern Recognition*, 4th ed.; Theodoridis, S., Koutroumbas, K., Eds.; Academic Press: Boston, MA, USA, 2009; pp. 567–594.
33. Vaygan, E.K.; Rajabi, R.; Estebansari, A. Short-Term Load Forecasting Using Time Pooling Deep Recurrent Neural Network. 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. 1–5.

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.