

Article

DiffNILM: A Novel Framework for Non-Intrusive Load Monitoring Based on the Conditional Diffusion Model

Ruichen Sun, Kun Dong * and Jianfeng Zhao

School of Electrical Engineering, Southeast University, Nanjing 210096, China; jianfeng_zhao@seu.edu.cn

* Correspondence: sirelays@seu.edu.cn

Abstract: Non-intrusive Load Monitoring (NILM) is a critical technology that enables detailed analysis of household energy consumption without requiring individual metering of every appliance, and has the capability to provide valuable insights into energy usage behavior, facilitate energy conservation, and optimize load management. Currently, deep learning models have been widely adopted as state-of-the-art approaches for NILM. In this study, we introduce DiffNILM, a novel energy disaggregation framework that utilizes diffusion probabilistic models to distinguish power consumption patterns of individual appliances from aggregated power. Starting from a random Gaussian noise, the target waveform is iteratively reconstructed via a sampler conditioned on the total active power and encoded temporal features. The proposed method is evaluated on two public datasets, REDD and UKDALE. The results demonstrated that DiffNILM outperforms baseline models on several key metrics on both datasets and shows a remarkable ability to effectively recreate complex load signatures. The study highlights the potential of diffusion models to advance the field of NILM and presents a promising approach for future energy disaggregation research.

Keywords: NILM; non-intrusive load monitoring; diffusion models; deep learning



Citation: Sun, R.; Dong, K.; Zhao, J. DiffNILM: A Novel Framework for Non-Intrusive Load Monitoring Based on the Conditional Diffusion Model. *Sensors* **2023**, *23*, 3540. <https://doi.org/10.3390/s23073540>

Academic Editors: Yongliang Xie, Liang Gong and Chang'e Cai

Received: 6 March 2023

Revised: 22 March 2023

Accepted: 26 March 2023

Published: 28 March 2023



Copyright: © 2023 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

In recent years, the demand for fine-grained power data has increased, leading to a growing interest in the energy disaggregation technique for obtaining information on appliance-level power consumption. A commonly cited application of this technique is to generate detailed electricity bills, which encourage energy conservation among residents. Additionally, electric power companies can utilize disaggregated power consumption data to calculate Demand Side Response (DSR) resources and evaluate DSR capability. An intuitive way to obtain such data is through Intrusive Load Monitoring (ILM), which involves the direct installation of sensors on target appliances. While ILM yields accurate results, it is generally considered unfeasible for large-scale deployment due to its high cost. On the other hand, Non-Intrusive Load Monitoring (NILM) has gained better application prospects from an economic standpoint. NILM can be viewed as a software sensor that identifies the operating states of individual appliances and estimates their power consumption using only power or current data recorded by the mains meter; thereby, reducing the overall cost.

The practical implementation of NILM has been facilitated by the development of big data technology in the energy industry. Advanced Metering Facilities (AMIs) provide real-time load monitoring data, and modern Artificial Intelligence (AI) algorithms can effectively process massive amounts of data.

In our study, energy disaggregation is framed as a generation task and a highly promising deep generative model, the diffusion model [1], is employed to reconstruct target power profiles. In the last two years, diffusion models have been gaining significant popularity and have nearly replaced Generative Adversarial Network (GAN) and other generative models due to their ease of training, improved tractability, and flexibility. Diffusion models have demonstrated exceptional performance in various fields, including image generation [2], image segmentation [3], audio synthesis [4] and point cloud reconstruction [5],

etc. However, to the best of our knowledge, no published research has investigated the use of diffusion models for NILM. Therefore, this paper proposes DiffNILM, a diffusion probabilistic model for energy disaggregation. The main contributions of our work are as follows:

- DiffNILM is the first NILM framework adopting the diffusion model. Specifically, We engineer the conditional diffusion model to address the NILM task, where the total active power and embedded time tags are fed to the model as conditional input, and the appliance power waveform is generated step-by-step from Gaussian noise.
- We propose an encoding method for multi-scale temporal features that takes into account the regularity of power consumption behaviors.
- We implement and evaluate the proposed method on two public datasets, REDD and UKDALE. Empirical results demonstrate that DiffNILM outperforms previous models, as evidenced by both classification metrics and regression metrics.

2. Related Works

The overall framework of NILM was pioneered by Professor Hart in the 1980s, as documented in [6]. This approach was based on the notion that electric appliances exhibit unique features during state transition, which formed the basis for the event-based load monitoring method. However, Hart's original approach only extracted steady-state features, which proved inadequate for appliances with multiple states and relatively low power consumption. To improve Hart's algorithm, researchers discovered that repeatable transient profiles could be observed with high sampling rates, which allowed for the recognition of appliances' transient signatures [7]. Various signal processing techniques, including Fourier Transform [8], Wavelet Transform [9], and Hilbert Transformation [10], were attempted to process transient power, current, and voltage signals.

To improve the identification accuracy, multiple electrical parameters were combined as input. The most prevailing method is the V-I trajectory, which maps current and voltage signals as a 2D image. Both steady-state and transient data can be utilized to generate V-I trajectories. For instance, Wang et al. [11] extracted the V-I trajectory based on steady-state data and developed an approach to quantify ten trajectory features. The research work in [12] utilized instantaneous voltage and current waveforms and proposed an algorithm that demonstrated high precision and strong robustness.

NILM based on traditional machine learning methods is mainly realized by the Hidden Markov Model (HMM) [13–15], Conditional Random Field (CRF) [16], and Support Vector Machine (SVM) [17]. These algorithms are supported by explicable mathematical principles, but their performances are often constrained by stringent assumptions (the characterization of load state transitions may not align with the actual operational features of various appliances), leading to limited accuracy and generalization abilities. Efforts have also been made to tackle the issue by framing it as a Combinatorial Optimization (CO) problem [18], but this method has proven to be computationally intractable, since it relies on enumeration.

With the significant progress of Deep Learning (DL), DL-based solutions have brought fresh insights to the practical advancement of artificial intelligence, which have been extensively adopted in various fields, including Computer Vision, Natural Language Processing, Signal Processing, etc. The application of DL-based techniques to energy disaggregation started with Kelly and Knottenbelt's pioneering work in 2015 [19], where they introduced three deep neural network architectures to NILM, surpassing CO and diverse HMM-based algorithms in terms of both accuracy and generalization capability. Since then, DL-based methods have gradually dominated NILM research.

Recurrent Neural Networks (RNNs) are a type of deep learning architecture particularly well-suited for handling sequential data [20]. However, the vanishing gradient problem has posed a major challenge in the field. To address this issue, Long Short-Term Memory (LSTM) networks [21] have been commonly used in NILM. Convolutional Neural Networks (CNNs) have proven to be highly effective for image tasks and excel in sequential

data analysis as well [22]. Zhang et al. [23] compared Seq2Point and Seq2Seq learning approaches using CNN-based mappings for training.

The models mentioned above have also been optimized to enhance computational efficiency since real-time load disaggregation is crucial for certain use cases, such as DSR and fault detection [24,25]. While accurate approaches have been proposed, there are also light-weight approaches to enable online computation, including a super-state hidden Markov model and a new variant of the Viterbi algorithm in an HMM-based framework for computationally efficient exact inference [26], as well as methods based on Gated Recurrent Unit (GRU), which reduce memory usage and computational complexity [27,28]. In addition, an experimental platform has been developed to realize real-time computation with a calculation time limit of one second [29].

In the past few years, Attention Mechanism has gained widespread popularity in handling sequential data processing tasks. The fundamental idea is to direct focus onto the most essential segment in the input sequence by assigning the highest weights to the most relevant parts. Capitalizing on the advantages of Attention Mechanism, Google introduced the Transformer architecture in 2017 [30], which allows parallel computation, as opposed to RNNs, and demonstrates a significantly superior capability to capture sequential features compared to CNNs. Building upon Transformer, the research work in [31,32] designed an architecture, based on Bidirectional Encoder Representations, from Transformers (BERT) for NILM, and proposed comprehensive loss functions that incorporate both regression and classification metrics.

NILM can also be regarded as a generation task aimed at creating synthetic waveforms for individual appliances, so the implementation of deep generative models, which are capable of modeling the underlying distribution of the power data, has been explored. A deep latent generative model for NILM, based on the Variational Recurrent Neural Network (VRNN), has been proposed, which performs sequence-to-many-sequence prediction [33]. The strong generational ability of the Variational Autoencoder (VAE) improves the formation of complex load profiles [34]. Conditional Generative Adversarial Network (cGAN) was used to avoid manually designing loss functions [35]. The work in [36] unified auto-encoder and GAN to realize the source separation of nonlinear power signals. Drawing inspiration from the favorable outcomes attained by several non-autoregressive generative models, the proposed study endeavors to employ the diffusion model, a more advanced approach, to the task of NILM.

3. Denoising Diffusion Probabilistic Models

With inspiration from non-equilibrium thermodynamics, the basic idea of DDPMs is to destroy the original data by gradually adding Gaussian noise, and then to learn to reconstruct the data through an inference process. The noising and denoising Markov chains are defined as the forward process and reverse process, respectively.

The step-by-step destruction and reconstruction of a power waveform in a diffusion model is illustrated in Figure 1. Random noise is successively added to x_0 , a segment of clean power waveform of the target appliance, until the discernible features are completely lost. In reverse, we start from a random Gaussian noise x_T and progressively remove extra noise to generate the target distribution. The original data x_0 and whitened latent variables x_1, x_2, \dots, x_T share the same dimensionality.

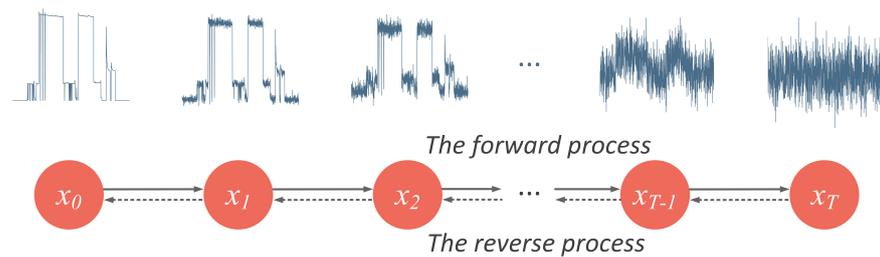


Figure 1. Illustration of the forward process and the reverse process in a NILM task. The pink arrows point out the process of forward diffusion, where a clean power pattern is gradually destroyed. The green arrows indicate the process of denoising inference, where the target waveform is recovered.

3.1. The Forward Process

Diffusion models can be seen as latent variable models which create mappings to a hidden feature space, and this process is controlled by a predefined linear schedule $\beta_{1:T} = [\beta_1, \beta_2, \dots, \beta_T]$. According to the defining characteristic of the Markov chain, the distribution of x_t at any arbitrary time step depends solely on its previous state x_{t-1} , so we add Gaussian noise to a x_t by means of the following formula:

$$x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1 - \alpha_t}\varepsilon_t \quad (1)$$

where $\alpha_t = 1 - \beta_t$ and $\varepsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The iterative formula of the forward process is given as:

$$q(x_t | x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t \mathbf{I}) \quad (2)$$

In order to spare us from having to do step-by-step iteration, we derived the closed-form expression to directly calculate x_t by x_0 using a reparameterization trick:

$$q(x_t | x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)\mathbf{I}) \quad (3)$$

where $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$.

In many applications of the diffusion process, the parameters $\beta_{1:T}$ are often assigned small values following an increasing pattern. For instance, in [1], $\beta_{1:T}$ is defined as a linear function with values ranging from 10^{-4} to 0.02 over 1000 time steps. As T grows sufficiently large, $\bar{\alpha}_t$ converges to zero, and the resulting distribution of the latent variable x_T approaches a standard normal distribution. The diffusion process ceases when the final distribution becomes sufficiently disordered to be considered an isotropic Gaussian distribution.

3.2. The Reverse Process

The reverse process is where the desired output data is generated by tracing the Markov chain backward. Starting from x_T , if the distribution of any x_{t-1} can be derived from the prior term x_t , the original distribution x_0 can be recovered from pure Gaussian noise. Unfortunately, the reverse transfer distribution $q(x_{t-1} | x_t)$ is not inferrable by simple mathematical derivation, so we used a deep learning model with parameter θ to estimate this reverse distribution, as depicted in Figure 2.

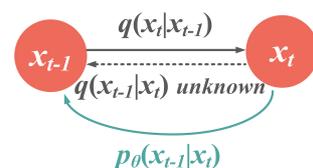


Figure 2. Illustration of transfer distributions between two diffusion steps. With the unsolvable $q(x_{t-1} | x_t)$, we approximate this reverse distribution with a parameterized model p_θ .

Conditioned on x_0 , the reverse conditional probability can be derived on the basis of the Bayes Rule:

$$\begin{aligned} q(x_{t-1} | x_t, x_0) &= q(x_t | x_{t-1}, x_0) \frac{q(x_{t-1} | x_0)}{q(x_t | x_0)} \\ &\propto \exp\left(-\frac{1}{2}\left(\frac{(x_t - \sqrt{\alpha_t}x_{t-1})^2}{\beta_t} + \frac{(x_{t-1} - \sqrt{\alpha_{t-1}}x_0)^2}{1 - \bar{\alpha}_{t-1}} - \frac{(x_t - \sqrt{\alpha_t}x_0)^2}{1 - \bar{\alpha}_t}\right)\right) \\ &= \exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1 - \bar{\alpha}_{t-1}}\right)x_{t-1}^2 - \left(\frac{2\sqrt{\alpha_t}}{\beta_t}x_t + \frac{2\sqrt{\alpha_t}}{1 - \bar{\alpha}_t}x_0\right)x_{t-1} + C\right)\right) \end{aligned} \quad (4)$$

where C is a term not involving x_{t-1} . According to the probability density function of the normal distribution, the mean and variance of Equation (4) can be expressed as:

$$\begin{cases} \tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \cdot \beta_t \\ \tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}x_t + \frac{\sqrt{\alpha_{t-1}}\beta_t}{1 - \bar{\alpha}_t}x_0 \end{cases} \quad (5)$$

Then, we transform Equation (1) into the form of $x_0 = \frac{1}{\sqrt{\alpha_t}}(x_t - \sqrt{1 - \bar{\alpha}_t}\varepsilon_t)$ to replace the unknown x_0 in Equation (5), and derive the target mean that only depends on x_t :

$$\tilde{\mu}_t(x_t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}}\varepsilon_t\right) \quad (6)$$

The above derivations reveal that the variance $\tilde{\beta}_t$ relies solely on the noise schedule and, thus, can be pre-computed. The parameter to be approximated (ε_t) exists in $\tilde{\mu}_t$, so we use a neural network to estimate the noise and, consequently, the mean.

3.3. Training a Diffusion Model

Diffusion models adopt the modeling method of noise prediction, where the neural networks take x_t and time step t as input to estimate the noise $\varepsilon_\theta(x_t, t)$. The goal of the training process is to narrow the gap between the actual noise and the predicted one by optimizing the negative log-likelihood using the variational lower bound. The loss term is parameterized as:

$$\begin{aligned} L_t(\theta) &= \mathbb{E}_{x_0, t, \varepsilon} \left[\frac{1}{2\|\Sigma_\theta(x_t, t)\|_2^2} \|\tilde{\mu}_t - \mu_\theta(x_t, t)\|_2^2 \right] \\ &= \mathbb{E}_{x_0, t, \varepsilon} \left[\frac{\beta_t^2}{2\alpha_t(1 - \bar{\alpha}_t)\|\Sigma_\theta\|_2^2} \left\| \varepsilon - \varepsilon_\theta\left(\sqrt{\alpha_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\varepsilon, t\right) \right\|^2 \right] \end{aligned} \quad (7)$$

A few simplifications lead to more stable training:

$$L_{\text{simple}}(\theta) = \mathbb{E}_{x_0, t, \varepsilon} \left[\left\| \varepsilon - \varepsilon_\theta\left(\sqrt{\alpha_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\varepsilon, t\right) \right\|^2 \right] \quad (8)$$

4. Design

4.1. Conditional Diffusion Model as Appliance-Level Data Generator

One of the salient features of NILM, as a generation task, is that, instead of randomly generating power sequences that follow a certain distribution, the generation of each segment of appliance power waveform is conditioned on a segment of aggregated power waveform with the same length. However, the vanilla diffusion model was originally designed for unconditional image generation, which necessitates adaptive modifications to tailor it to the requirements of the NILM task.

Conditional diffusion models have been well-studied in other sequence modeling tasks. For instance, in machine translation the model conditions on the source sentences, and in speech synthesis the model conditions on the mel-spectrogram. The general goal of such algorithms is to model the probability density of $p_\theta(x_0 | x_d)$, where x_d contains

conditioning features relevant to x_0 . For diffusion models, the conditional distribution can be written as:

$$p_{\theta}(x_0 | x_d) := \int p_{\theta}(x_{0:N} | x_d) dx_{1:N} \quad (9)$$

The proposed model takes two conditional inputs: the total power and the encoded temporal features. Traditional NILM algorithms only detect the states of appliances based on the aggregated power sequence, disregarding the regularity and periodicity in the energy consumption patterns of users (for instance, dishwashers are generally used after dinner, and refrigerators operate more frequently during summer). In this study, we present an encoding technique that integrates multi-scale temporal information as supplementary knowledge for energy disaggregation, with reference to the global timestamp representation introduced in [37]. As illustrated in Figure 3, we extract three features from each time tag: hour of day, day of week and month of year, and then linearly encode these three features into values within the interval of $[-0.5, 0.5]$, respectively.

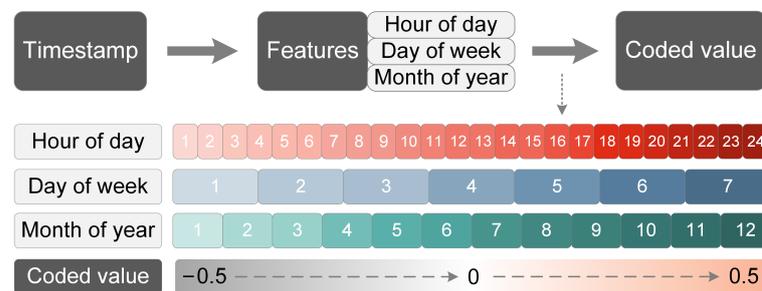


Figure 3. Illustration of temporal data encoding.

Moreover, the continuous noise level is adopted in this paper, as opposed to discrete noise level, where we sample $t \sim \text{Uniform}(\{1, 2, \dots, T\})$ and reach for the corresponding α_t in the predefined linear schedule. The proposed diffusion model conditions on the continuous noise level $\sqrt{\bar{\alpha}}$ instead of time step t and $\sqrt{\bar{\alpha}}$ is randomly chosen between two adjacent discrete noise levels:

$$\sqrt{\bar{\alpha}} \sim \text{Uniform}(\sqrt{\bar{\alpha}_t}, \sqrt{\bar{\alpha}_{t-1}}) \quad (10)$$

For our task, as depicted in Figure 4, the neural network takes three inputs: the noisy appliance-level power data $x_{\bar{\alpha}}$, the corresponding noise level $\sqrt{\bar{\alpha}}$, the conditional aggregated power data x_{aggre} and embedded time tags x_{time} , and outputs the approximated noise $\varepsilon_{\theta}(x_{\bar{\alpha}}, \sqrt{\bar{\alpha}}, x_d)$.

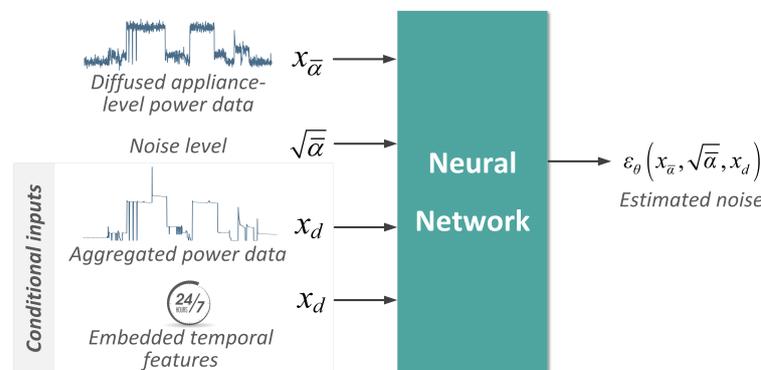


Figure 4. Conditional diffusion model for NILM task. The neural network takes diffused data $x_{\bar{\alpha}}$, noise level $\sqrt{\bar{\alpha}}$ and conditional data x_{aggre} and x_{times} as inputs to estimate the corresponding noise ε_{θ} .

4.2. Network Architecture

This section details the implementation of a neural network for noise prediction, with an architecture inspired by NU-Wave [38] and DiffWave [4], which are two diffusion-based neural vocoders.

As revealed in Figure 5, 1D convolutional layers were used to increase the number of channels of the input sequences $x_{\bar{\alpha}}$, x_{aggre} and x_{time} to C , and the Sigmoid Linear Unit (SiLU) activation was adopted:

$$SiLu(x) = x * Sigmoid(x) \tag{11}$$

Similar to the positional embedding method, proposed in Transformer [30], the sinusoidal encoding formula is applied to embed the noise level $\sqrt{\bar{\alpha}}$:

$$Embed(\sqrt{\bar{\alpha}}) = \left[\sin\left(10^{-\frac{[0:63]}{16}} \times 50,000 \times \sqrt{\bar{\alpha}}\right), \cos\left(10^{-\frac{[0:63]}{16}} \times 50,000 \times \sqrt{\bar{\alpha}}\right) \right] \tag{12}$$

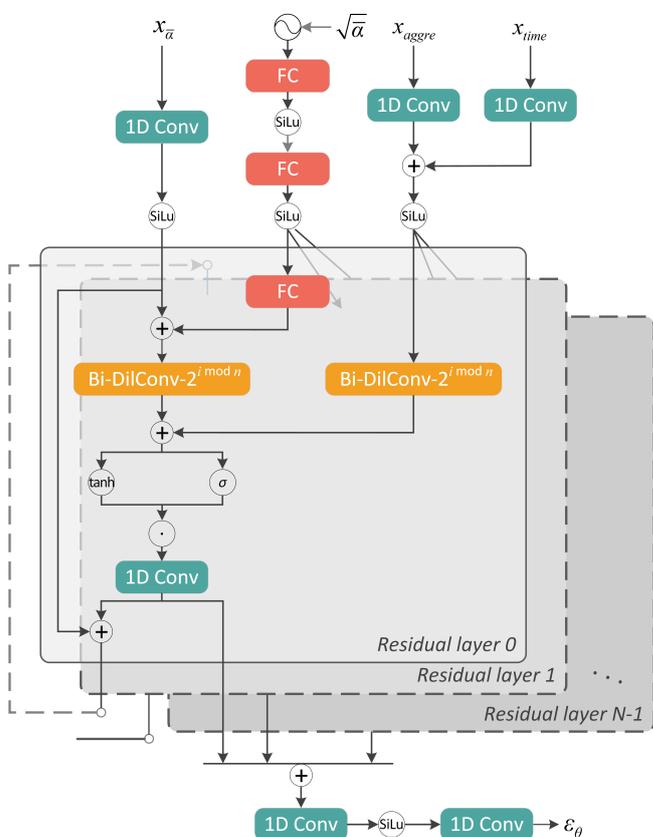


Figure 5. The neural network architecture for noise prediction.

Then we use two shared SiLu-activated Fully Connected (FC) layers and one residual-layer-specific FC layer to project the encoded noise level to a C -dimensional vector, and add it to the convoluted $x_{\bar{\alpha}}$ as a bias term.

The main body of the model consists of N conformably-structured layers connected in residual manner to enable the direct delivery of input information to the final layers. In each residual layer, we used Bi-directional Dilated Convolution (Bi-DilConv) to deal with the inputs for an exponential growth in the receptive field, and in the i -th residual layer, the spacing between the kernel points was set to $2^{i \bmod n}$. Gated Units (GU) are applied to activate the summation of the processed noisy signals and conditional signals. Then, the convoluted vector is split in two and passed on as residual output and skip output, respectively. Finally, we sum all the skip connections and use two convolutional layers to gain the noise vector ϵ_{θ} in the same shape as $x_{\bar{\alpha}}$ and x_{aggre} .

4.3. Training and Sampling Procedures

The training and sampling procedures of the diffusion model are shown in Algorithms 1 and 2. In the training procedure, after extracting data from the dataset, we sample an iteration index t and obtain a corresponding continuous noise level $\sqrt{\bar{\alpha}}$ to determine the extent of whitening applied to the original waveform. As mentioned in Section 3.3, the deep learning model updates its parameters with the purpose of minimizing the distance between the sampled noise ε and the predicted noise ε_θ . Instead of using common loss functions, such as MSE and L_1 norm, we found that log-norm improves convergence speed and leads to improved empirical outcomes:

$$\log \left\| \varepsilon - \varepsilon_\theta \left(x_{\bar{\alpha}}, \sqrt{\bar{\alpha}}, x_d \right) \right\|_1 \tag{13}$$

In the sampling algorithm, we adopted a fast sampling method where much fewer inference steps are used. Instead of traversing the reverse process step by step with $t = T, T - 1, \dots, 1$, we define an inference schedule with only T_{infer} noise levels ($T_{infer} \ll T$). The test results demonstrated that the fast sampling trick greatly accelerated the inference procedure without degrading generational quality. In each inference step, we calculate the predicted variance $\tilde{\beta}_t$ and mean $\mu_\theta(x_{\bar{\alpha}}, \sqrt{\bar{\alpha}}, x_d)$ to estimate the previous term.

Algorithm 1: Training.

- 1: **repeat**
 - 2: $x_0 \sim q(x_0)$
 - 3: $t \sim \text{Uniform}(\{1, 2, \dots, T\})$
 - 4: $\sqrt{\bar{\alpha}} \sim \text{Uniform}(\sqrt{\bar{\alpha}_t}, \sqrt{\bar{\alpha}_{t-1}})$
 - 5: $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
 - 6: Take gradient descent step on $\nabla_\theta \log \left\| \varepsilon - \varepsilon_\theta \left(x_{\bar{\alpha}}, \sqrt{\bar{\alpha}}, x_d \right) \right\|_1$
 - 7: **until** converged
-

Algorithm 2: Sampling.

- 1: $x_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
 - 2: **for** $t = T_{infer}, T_{infer} - 1, \dots, 1$ **do**
 - 3: $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
 - 4: $\tilde{\beta}_t = \sqrt{\frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}} \beta_t$
 - 5: $\mu_\theta(x_{\bar{\alpha}}, \sqrt{\bar{\alpha}}, x_d) = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \varepsilon_\theta(x_{\bar{\alpha}}, \sqrt{\bar{\alpha}}, x_d) \right)$
 - 6: $x_{t-1} = \mu_\theta(x_t, t) + \tilde{\beta}_t z$
 - 7: **end for**
 - 8: **return** x_0
-

5. Experiments

We carried out an experiment to test the proposed model. The workflow, as illustrated in Figure 6, involved pre-processing data, splitting the dataset, training a neural network using the training set, and evaluating its performance on the testing set.

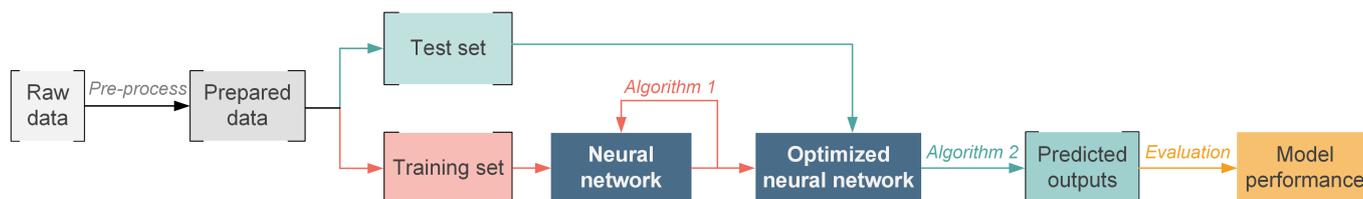


Figure 6. The workflow of the experiment conducted.

5.1. Dataset

This study employed low-frequency active power data from the REDD and UKDALE datasets to train and test the proposed model. REDD is the most widely-used dataset in the domain of NILM, comprising the mains and submeter power data of six residential homes in the United States, recorded over a period of approximately four months. The UKDALE dataset, on the other hand, was published by Imperial College London, in 2014, and contains power consumption information from House 1 collected for up to three years, while the data for the other four houses were recorded for several months.

We pre-processed the original power data according to the following procedure:

Step 1: Merge the data of split-phase mains meter. Two-phase power supply is commonly-used in North American households, so, for REDD, we calculated the sum of each mains meter to obtain the actual aggregated power data.

Step 2: Resample the power data at a fixed interval of 6 s.

Step 3: Fill data gaps shorter than 3 min by forward-filling, and fill those longer than 3 min with zeros.

Step 4: Attach status labels to the datasets. An appliance is classified as being in an ‘on’ state at a particular time point and assigned a status label of 1, provided that its power consumption falls within the acceptable ‘on’ power range and its operation time exceeds the minimum duration specified in Table 1. Otherwise, a status label of 0 is assigned.

Step 5: Standardize the power data according to Formula (14) to enhance the accuracy of the model and convergence speed.

$$x^* = \frac{x - \mu}{\sigma} \quad (14)$$

Table 1. Basic parameter settings of appliances.

Appliance	Reasonable ‘on’ Power Range (W)	Minimum Duration of Operation (s)
Microwave	200~1800	12
Washer	40~3500	1800
Dish washer	50~1200	1800
Refrigerator	50~400	60

Following the pre-processing of the power data, overlapping sliding windows were utilized to extract sequences of processable length.

5.2. Evaluation Metrics

The selection of suitable metrics is important in appraising the algorithm’s performance. As NILM can be formulated as either a binary classification problem (to detect the on/off states of the target appliance) or a regression problem (to estimate the numeric value of power consumption), the evaluation incorporated both classification and regression metrics to ensure a comprehensive assessment.

5.2.1. Classification Metrics

We used classification metrics in Equation (16) to evaluate the ability of the algorithm to identify the on/off states, where TP , FP and FN , respectively, represent the number of TP (True Positive), FP (False Positive) and FN (False Negative) results, and P and N , respectively, represent the number of points where the appliance is switched on and off in ground truth.

$$\begin{aligned}
 accuracy &= \frac{TP + TN}{P + N} \\
 recall &= \frac{TP}{TP + FN} \\
 precision &= \frac{TP}{TP + FP} \\
 F\text{-score} &= (1 + \beta^2) \times \frac{precision \times recall}{\beta^2 \times precision + recall} \\
 F1\text{-score} &= 2 \times \frac{precision \times recall}{precision + recall}
 \end{aligned} \tag{15}$$

While *accuracy* is an intuitive classification metric, its applicability is restricted in datasets that are unbalanced, where the ‘on’ states of appliances constitute a small fraction of the entire sequence. In such cases, the *F-score* index serves as an effective approach to address the imbalance issue. The *F-score* comprehensively incorporates both *precision* and *recall*, and varying weights can be assigned to them by adjusting the β value, thereby enabling an evaluation of the quality of NILM algorithms under diverse application scenarios. Given that *precision* and *recall* are usually deemed equally important, the value of β was set to 1, and *F-score* was calculated as the harmonic average of the two, termed as *F1-score*.

5.2.2. Regression Metrics

To evaluate the performance of the model to reconstruct the power profiles of the target appliance, two commonly-used regression metrics, Mean Absolute Error (*MAE*) and Mean Relative Error (*MRE*), were adopted:

$$\begin{aligned}
 MAE &= \frac{1}{T} \sum_{t=1}^T |\hat{x}_t - x_t| \\
 MRE &= \frac{1}{T} \sum_{t=1}^T \frac{\hat{x}_t - x_t}{\max(\hat{x}_t, x_t)}
 \end{aligned} \tag{16}$$

where \hat{x}_t and x_t , respectively, represent the appliance’s estimated and actual power at time t , and T is the total number of points in the sequence.

5.3. Implementation Details

The NILM project was conducted on a 64-bit computer equipped with Intel(R) Core™ i7-12700 CPU@ 3.61GHz, 32GB memory, and NVIDIA GeForce RTX 3080Ti. The Pytorch framework was employed to train and test the diffusion model.

During the training phase, the model was trained until convergence at a learning rate of 3×10^{-5} . To accelerate gradient descent, the Adam optimizer was utilized, where the hyperparameters β_1 and β_2 were set to 0.5 and 0.999, respectively.

The hyperparameters of the diffusion model are shown in Table 2.

Table 2. Hyperparameters of the model.

Symbol	Description	Value
L	Length of the input and output power sequences	480
T	Maximum diffusion step	1000
$\beta_{1:T}$	Noise schedule	Linear(1×10^{-6} , 0.006, 1000)
T_{infer}	Inference step	8
$\beta_{1:T_{infer}}$	Inference noise schedule	$1 \times 10^{-6}, 2 \times 10^{-6}, 1 \times 10^{-5}, 1 \times 10^{-4}, 1 \times 10^{-3}, 1 \times 10^{-2}, 1 \times 10^{-1}, 9 \times 10^{-1}$
N	Number of residual layers	30
C	Number of residual channels	128
n	Length of the dilation cycle	10

5.4. Results

DiffNILM was evaluated against four state-of-the-art NILM models, including the bi-directional LSTM [21], CNN [23], BERT4NILM [31] and cGAN [35]. The objectivity of the comparative experiments was ensured by adopting the same data processing method, and all the baseline models were trained to convergence. The performance indicators of the five disaggregation models on REDD and UKDALE datasets are shown in Tables 3 and 4. Output sample curves generated by DiffNILM, BERT4NILM, and cGAN models are displayed in Figures 7 and 8, where two relatively underperforming methods were excluded to avoid clutter.

For starters, we examined the performance of DiffNILM on microwaves and kettles, which are characterized by infrequent usage and relatively short running periods. The results from the tables indicate that the proposed algorithm outperformed other methods on several indicators, particularly the MAE and MRE. The output signals further reveal that the model effectively captured most of the activations and the predicted power values aligned well with the ground truth values. However, a few exceptional cases were identified where the power signatures were not exactly typical, notably in the first activation of the microwave, depicted in Figure 8, which exhibited a longer turn-on time than other instances, and was subject to relatively strong interference from background noise.

Table 3. Model performances on REDD.

Appliance	Model	Accuracy ↑	F1-Score ↑	MAE ↓	MRE ↓
Microwave	Bi-LSTM	0.989	0.604	17.39	0.058
	CNN	0.986	0.378	18.59	0.060
	BERT4NILM	0.989	0.476	17.58	0.057
	cGAN	0.989	0.415	18.15	0.058
	DiffNILM	0.989	0.430	17.13	0.057
Washer	Bi-LSTM	0.989	0.125	35.73	0.020
	CNN	0.970	0.274	36.12	0.042
	BERT4NILM	0.991	0.559	34.96	0.022
	cGAN	0.990	0.478	29.67	0.025
	DiffNILM	0.988	0.569	26.44	0.019
Dish washer	Bi-LSTM	0.956	0.421	25.25	0.056
	CNN	0.953	0.298	25.29	0.053
	BERT4NILM	0.969	0.523	20.49	0.039
	cGAN	0.951	0.295	24.80	0.055
	DiffNILM	0.971	0.593	18.16	0.037
Refrigerator	Bi-LSTM	0.789	0.709	44.82	0.841
	CNN	0.796	0.689	35.69	0.822
	BERT4NILM	0.841	0.756	32.35	0.806
	cGAN	0.811	0.732	33.83	0.820
	DiffNILM	0.868	0.794	33.58	0.808
Average	Bi-LSTM	0.931	0.465	30.80	0.244
	CNN	0.926	0.410	28.92	0.244
	BERT4NILM	0.948	0.579	26.35	0.231
	cGAN	0.935	0.498	26.70	0.240
	DiffNILM	0.954	0.597	23.70	0.230

↑ indicates that a higher value of the metric is better, while ↓ indicates that a lower value of the metric is better. The bold item in each column represents the optimal index for that particular metric in all the models.

Table 4. Model performances on UKDALE.

Appliance	Model	Accuracy ↑	F1-Score ↑	MAE ↓	MRE ↓
Microwave	Bi-LSTM	0.995	0.060	6.55	0.014
	CNN	0.995	0.341	6.36	0.014
	BERT4NILM	0.995	0.014	6.57	0.014
	cGAN	0.996	0.474	5.98	0.012
	DiffNILM	0.996	0.501	4.54	0.012
Washer	Bi-LSTM	0.938	0.150	15.66	0.067
	CNN	0.913	0.173	11.90	0.094
	BERT4NILM	0.966	0.325	6.98	0.040
	cGAN	0.959	0.376	10.84	0.062
	DiffNILM	0.986	0.390	5.74	0.058
Dish washer	Bi-LSTM	0.976	0.605	36.36	0.033
	CNN	0.947	0.560	25.45	0.069
	BERT4NILM	0.966	0.667	16.18	0.049
	cGAN	0.961	0.646	13.89	0.042
	DiffNILM	0.980	0.662	19.58	0.030
Refrigerator	Bi-LSTM	0.573	0.174	43.74	0.956
	CNN	0.772	0.718	29.29	0.758
	BERT4NILM	0.813	0.766	25.47	0.732
	cGAN	0.818	0.801	25.11	0.730
	DiffNILM	0.857	0.816	22.82	0.699
Kettle	Bi-LSTM	0.994	0.531	21.26	0.007
	CNN	0.997	0.850	9.64	0.003
	BERT4NILM	0.998	0.907	6.82	0.002
	cGAN	0.998	0.911	7.09	0.002
	DiffNILM	0.999	0.918	4.59	0.002
Average	Bi-LSTM	0.875	0.229	21.80	0.261
	CNN	0.919	0.521	14.28	0.261
	BERT4NILM	0.943	0.503	11.47	0.194
	cGAN	0.946	0.642	14.58	0.170
	DiffNILM	0.964	0.657	11.45	0.164

↑ indicates that a higher value of the metric is better, while ↓ indicates that a lower value of the metric is better. The bold item in each column represents the optimal index for that particular metric in all the models.

Washers and dishwashers are a type of household appliances that exhibit infrequent use but extended operation per use. The consumption patterns are intricate, due to the frequent start-and-stop events and mode switching during operation. In the REDD dataset, Washers maintained a constant power level during the ‘on’ mode and the waveforms were effectively rebuilt by DiffNILM, despite the slightly elevated power values. Washers in the UK have operating patterns that are distinct from their US counterparts, with evident power oscillations, which the proposed algorithm effectively reconstructed. Dishwashers present more complex operational characteristics with multiple modes, such as pre-rinse, steam wash and dry, which require a more advanced model generation capacity. Although DiffNILM’s output in low power consumption mode was not entirely consistent with the ground truth signal, it exhibited good overall power estimation performance.

The refrigerator operates based on automatic temperature regulation requirements, with frequent start and stop events and prominent periodicity. Based on the evaluation metrics and sample waveforms, DiffNILM exhibited satisfactory performance in disaggregating the refrigerator load. The algorithm could accurately detect each activation event, and the power prediction accuracy was only compromised when there was significant background power interference.

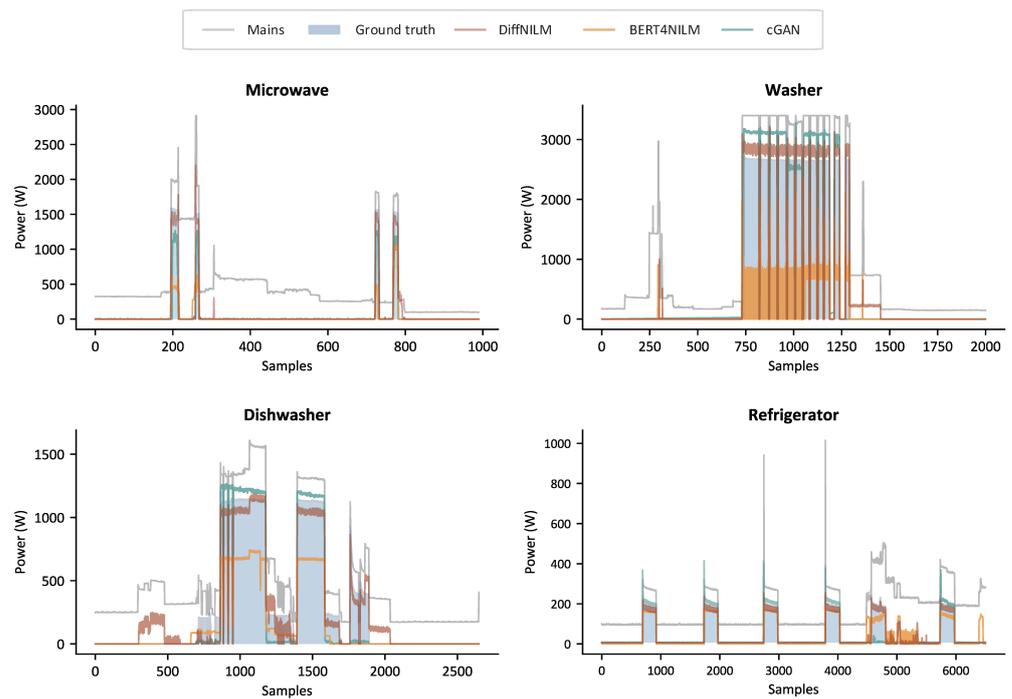


Figure 7. Sample outputs of microwave, Washer, dishwasher and refrigerator on REDD.

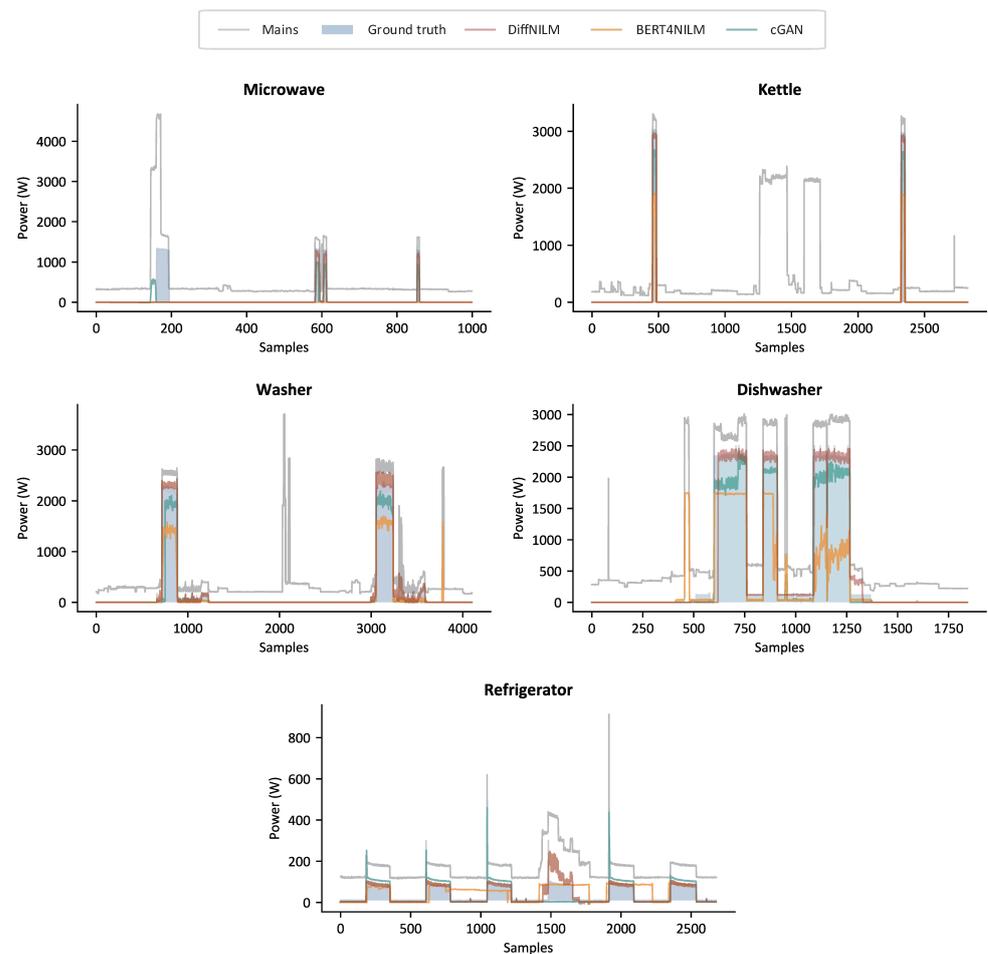


Figure 8. Sample outputs of microwave, kettle, Washer, dishwasher and refrigerator on UKDALE.

Through horizontal comparison of the results on the two datasets, it is interesting to notice that the metrics of the two generative models exhibited more enhanced performance on UKDALE than REDD. A plausible reason for this outcome is that deep generative models typically necessitate larger amounts of training data. Specifically, the smaller REDD dataset might fail to meet the data requirements of cGAN and DiffNILM, which, in turn, would hinder their performances on this dataset. In contrast, the larger UKDALE dataset facilitated better performance, reflected in the significant improvement of the metrics of the generative models.

Overall, the proposed algorithm outperformed the baseline models on most metrics and yielded better results than the previous methods concerning the mean values of the four metrics on both datasets. Meanwhile, DiffNILM demonstrated a satisfactory fitting effect on the consumption signals of various electrical appliances, and was capable of handling complex load patterns. Nonetheless, due to the unique nature of ‘diffusion’, the predicted power curve was not always smooth. Additionally, in cases where the background power was complex, the disaggregated curve might experience distortion following the total power, although the impact remained within an acceptable range.

6. Conclusions

In this paper, we introduce DiffNILM, a novel framework for energy disaggregation that utilizes the diffusion probabilistic model. The key innovation of our approach is the conditional diffusion model which takes both the total active power and embedded time tags as inputs and generates the appliance power waveforms. Additionally, we propose an encoding method for multi-scale temporal features which captures the periodicity of power consumption behaviors. The proposed method was applied and assessed on two open-access datasets, REDD and UKDALE. Averaging across all appliances, DiffNILM displayed an improvement in all four metrics on both datasets. The results also highlight the potential of the proposed DiffNILM algorithm in reconstructing complex load patterns, despite the fact that DiffNILM exhibits certain issues, such as generating power waveforms that are not sufficiently smooth and may experience distortion.

Meanwhile, we would like to clarify that the algorithm was developed with accuracy as the primary objective, and we did not explicitly consider the computational burden of the proposed implementation. Going forward, we are committed to developing a light-weight version of the algorithm that balances both accuracy and computational efficiency. This will enable the approach to be deployed in real-world settings with limited computational resources.

Furthermore, when analyzing the results, the significance of dataset size in achieving optimal performance was noted. However, acquiring large-scale appliance-level data through field sampling in numerous households can be a formidable task. In forthcoming research, we aim to explore a method of synthesizing appliance power signatures as a means of augmenting the existing NILM datasets, which can also be realized with diffusion models.

Author Contributions: Conceptualization, R.S. and K.D.; Methodology, R.S.; Software, R.S.; Validation, R.S.; Writing—original draft, R.S.; Writing—review & editing, K.D.; Supervision, K.D. and J.Z.; Project administration, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: To access the public dataset used in this paper, please follow the links provided: <http://redd.csail.mit.edu/>, accessed on 1 March 2023, for REDD and <https://jack-kelly.com/data/forUKDALE>, accessed on 1 March 2023.

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

References

1. Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6840–6851.
2. Sehwag, V.; Hazirbas, C.; Gordo, A.; Ozgenel, F.; Canton, C. Generating High Fidelity Data from Low-density Regions using Diffusion Models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11492–11501.
3. Wolleb, J.; Sandkühler, R.; Bieder, F.; Valmaggia, P.; Cattin, P.C. Diffusion Models for Implicit Image Segmentation Ensembles. *arXiv* **2021**, arXiv:2112.03145.
4. Kong, Z.; Ping, W.; Huang, J.; Zhao, K.; Catanzaro, B. Diffwave: A versatile diffusion model for audio synthesis. *arXiv* **2020**, arXiv:2009.09761.
5. Luo, S.; Hu, W. Diffusion probabilistic models for 3d point cloud generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–20 June 2021; pp. 2837–2845.
6. Hart, G.W. Nonintrusive appliance load monitoring. *Proc. IEEE* **1992**, *80*, 1870–1891. [[CrossRef](#)]
7. Mukaroh, A.; Le, T.T.H.; Kim, H. Background load denoising across complex load based on generative adversarial network to enhance load identification. *Sensors* **2020**, *20*, 5674. [[CrossRef](#)] [[PubMed](#)]
8. Kang, H.; Kim, H. Household appliance classification using lower odd-numbered harmonics and the bagging decision tree. *IEEE Access* **2020**, *8*, 55937–55952.
9. Gillis, J.M.; Alshareef, S.M.; Morsi, W.G. Nonintrusive load monitoring using wavelet design and machine learning. *IEEE Trans. Smart Grid* **2015**, *7*, 320–328. [[CrossRef](#)]
10. Heo, S.; Kim, H. Toward load identification based on the Hilbert transform and sequence to sequence long short-term memory. *IEEE Trans. Smart Grid* **2021**, *12*, 3252–3264.
11. Wang, A.L.; Chen, B.X.; Wang, C.G.; Hua, D. Non-intrusive load monitoring algorithm based on features of V-I trajectory. *Electr. Power Syst. Res.* **2018**, *157*, 134–144. [[CrossRef](#)]
12. Hassan, T.; Javed, F.; Arshad, N. An empirical investigation of VI trajectory based load signatures for non-intrusive load monitoring. *IEEE Trans. Smart Grid* **2013**, *5*, 870–878. [[CrossRef](#)]
13. 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 August 2011; Volume 25, pp. 59–62.
14. Kim, H.; Marwah, M.; Arlitt, M.; Lyon, G.; Han, J. Unsupervised disaggregation of low frequency power measurements. In Proceedings of the 2011 SIAM International Conference on Data Mining, Mesa, AZ, USA, 28–30 April 2011; pp. 747–758.
15. Kolter, J.Z.; Jaakkola, T. Approximate inference in additive factorial hmms with application to energy disaggregation. In Proceedings of the Artificial Intelligence and Statistics, La Palma, Spain, 21–23 April 2012; pp. 1472–1482.
16. Mao, Y.; Dong, K.; Zhao, J. Non-intrusive load decomposition technology based on CRF model. In Proceedings of the 2021 IEEE Sustainable Power and Energy Conference (iSPEC), Nanjing, China, 23–25 December 2021; pp. 3909–3915.
17. Gong, F.; Han, N.; Zhou, Y.; Chen, S.; Li, D.; Tian, S. A svm optimized by particle swarm optimization approach to load disaggregation in non-intrusive load monitoring in smart homes. In Proceedings of the 2019 IEEE 3rd Conference on Energy Internet and Energy System Integration (EI2), Changsha, China, 8–10 November 2019; pp. 1793–1797.
18. Piga, D.; Cominola, A.; Giuliani, M.; Castelletti, A.; Rizzoli, A.E. Sparse optimization for automated energy end use disaggregation. *IEEE Trans. Control Syst. Technol.* **2015**, *24*, 1044–1051. [[CrossRef](#)]
19. 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.
20. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [[CrossRef](#)]
21. Xia, M.; Wang, K.; Song, W.; Chen, C.; Li, Y. Non-intrusive load disaggregation based on composite deep long short-term memory network. *Expert Syst. Appl.* **2020**, *160*, 113669. [[CrossRef](#)]
22. LeCun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. In *The Handbook of Brain Theory and Neural Networks*; MIT Press: Cambridge, MA, USA, 1995; Volume 3361, p. 1995.
23. 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, LO, USA, 2–7 February 2018; Volume 32.
24. He, D.; Lin, W.; Liu, N.; Harley, R.G.; Habetler, T.G. Incorporating non-intrusive load monitoring into building level demand response. *IEEE Trans. Smart Grid* **2013**, *4*, 1870–1877.
25. Athanasiadis, C.L.; Papadopoulos, T.A.; Doukas, D.I. Real-time non-intrusive load monitoring: A light-weight and scalable approach. *Energy Build.* **2021**, *253*, 111523. [[CrossRef](#)]
26. Makonin, S.; Popowich, F.; Bajić, I.V.; Gill, B.; Bartram, L. Exploiting HMM sparsity to perform online real-time nonintrusive load monitoring. *IEEE Trans. Smart Grid* **2015**, *7*, 2575–2585. [[CrossRef](#)]
27. Rafiq, H.; Zhang, H.; Li, H.; Ochani, M.K. Regularized LSTM based deep learning model: First step towards real-time non-intrusive load monitoring. In Proceedings of the 2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE), Oshawa, ON, Canada, 12–15 August 2018; pp. 234–239.
28. Krystalakos, O.; Nalmpantis, C.; Vrakas, D. Sliding window approach for online energy disaggregation using artificial neural networks. In Proceedings of the 10th Hellenic Conference on Artificial Intelligence, Patras, Greece, 9–12 July 2018; pp. 1–6.

29. Hu, M.; Tao, S.; Fan, H.; Li, X.; Sun, Y.; Sun, J. Non-intrusive load monitoring for residential appliances with ultra-sparse sample and real-time computation. *Sensors* **2021**, *21*, 5366. [[CrossRef](#)]
30. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
31. Yue, Z.; Witzig, C.R.; Jorde, D.; Jacobsen, H.A. Bert4nilm: A bidirectional transformer model for non-intrusive load monitoring. In Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring, Virtual, 18 November 2020; pp. 89–93.
32. Sykiotis, S.; Kaselimi, M.; Doulamis, A.; Doulamis, N. Electricity: An efficient transformer for non-intrusive load monitoring. *Sensors* **2022**, *22*, 2926. [[CrossRef](#)]
33. Bejarano, G.; DeFazio, D.; Ramesh, A. Deep latent generative models for energy disaggregation. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 850–857.
34. Langevin, A.; Carbonneau, M.A.; Cheriet, M.; Gagnon, G. Energy disaggregation using variational autoencoders. *Energy Build.* **2022**, *254*, 111623. [[CrossRef](#)]
35. Pan, Y.; Liu, K.; Shen, Z.; Cai, X.; Jia, Z. Sequence-to-subsequence learning with conditional gan for power disaggregation. In Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 3202–3206.
36. Kaselimi, M.; Doulamis, N.; Voulodimos, A.; Doulamis, A.; Protopapadakis, E. EnerGAN++: A generative adversarial gated recurrent network for robust energy disaggregation. *IEEE Open J. Signal Process.* **2020**, *2*, 1–16. [[CrossRef](#)]
37. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 22 February 22–1 March 2022; Volume 35, pp. 11106–11115.
38. Lee, J.; Han, S. Nu-wave: A diffusion probabilistic model for neural audio upsampling. *arXiv* **2021**, arXiv:2104.02321.

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.