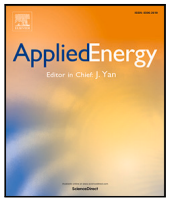
Applied Energy 365 (2024) 123157

Contents lists available at ScienceDirect

Applied Energy

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



ApplianceFilter: Targeted electrical appliance disaggregation with prior knowledge fusion☆

Dong Ding [a], Junhuai Li [b,*], Huaijun Wang [b], Kan Wang [b], Jie Feng [c], Ming Xiao [d]

[a] Xi'an University of Technology, School of Electrical Engineering, NO. 58 Yanxiang Road, Xi'an, 710054, ShaanXi, China
[b] Xi'an University of Technology, School of Computer Science and Engineering, NO. 5 South Jinhua Road, Xi'an, 710048, Shaanxi, China
[c] Xidian University, School of Telecommunications Engineering, No. 2 South Taibai Road, Xi'an, 710071, Shaanxi, China
[d] KTH Royal Institute of Technology, Department of Information Science and Engineering, Malvinas Väg 10, 10044, Stockholm, Sweden

ARTICLE INFO

Keywords:
Non-intrusive load monitoring
Load disaggregation
Prior knowledge
Expert feature
Deep learning

ABSTRACT

In smart home services, non-intrusive load monitoring (NILM) can reveal individual appliances' power consumption from the aggregate power and requires only one measurement point at the entrance by a smart meter. Most of the existing load disaggregation methods are based on deep and complex neural networks, and excessively long input sequences could increase the model disaggregation time. Meanwhile, constructing representative features and designing effective disaggregation model is becoming increasingly important. Therefore, we utilize a gramian summation difference angular field (GASDF) image, taking any two power sample points' temporal correlations as input to our baseline model, to better recognize different appliances from the aggregate power sequence. Then, since GASDF could not provide statistical characteristics, we further build the expert feature encoder (EFE) to realize the multi-dimensional representation of power by encoding both current aggregate power and statistical characteristics from historical data as prior knowledge. Afterwards, a batch-normalization (BN)-based normalization fusion (NF) method is proposed to lower the disaggregation error incurred by the distribution difference between GASDF and prior knowledge. Finally, to verify the proposed method's effectiveness, named ApplianceFilter, experiments are conducted on the UK-DALE and REDD data, showing that load disaggregation is improved using prior knowledge fusion, superior to the existing end-to-end neural network model.

1. Introduction

In recent years, greenhouse gas emissions, especially carbon dioxide, have incurred natural disasters, and thus mitigating the carbon dioxide emission has become a common goal worldwide. In particular, to reduce the carbon emission, appliance load monitoring in residential electricity has gained lots of interests, and appliance-level electricity utilization feedback could save 5%–15% energy consumption [1]. There typically exist two methods in residential appliance load monitoring, i.e., intrusive load monitoring (ILM) and non-intrusive load monitoring (NILM), both of which could improve household energy management efficiency [2–4]. To achieve the fine-grained appliance-level energy consumption management, ILM has to install monitoring sensors in each monitored appliance, while NILM acquires only one measurement point at the entrance by using intelligence algorithms in a smart meter, which is easier to install [5,6]. Meanwhile, smart meters could utilize the fifth-generation communication or Ethernet to facilitate real-time access to electricity consumption information [7]. Therefore, based on the lower cost and simpler installation principle, NILM has captured more attention.

Recent NILM studies have explored many state-of-the-art models in deep learning, e.g., denoising autoencoder (DAE), long short-term memory (LSTM), and convolution neural network (CNN) [8], most of which only use raw electrical data (power, voltage or current), without reconstructing the data for new features. For instance, Kelly et al. in [9] introduced the deep neural network into NILM for the first time, but the input of network was unreconstructed raw power sequence. Although the disaggregation is improved over traditional machine learning methods, e.g., combinatorial optimization [10] and factorial hidden Markov model (FHMM) [11], its feature is limited due to the insufficient exploitation on statistical characteristics. Later, in [12,13], the raw voltage and current data were reconstructed by V-I trajectory, which were further quantified as current span, shape of middle segment,

☆ This work was supported by the National Key R&D Program of China (No. 2018YFB1703000), and the National Natural Science Foundation of China (61801379 and 62102297).

* Corresponding author.
E-mail address: lijunhuai@xaut.edu.cn (J. Li).

https://doi.org/10.1016/j.apenergy.2024.123157
Received 12 September 2023; Received in revised form 19 March 2024; Accepted 31 March 2024
Available online 20 April 2024

area of right and left segments, area enclosed segments, and peak of middle segment, etc. Extracting features from trajectory, nevertheless, is complicated and not robust to data noise [14]. Therefore, in [15], V-I trajectory was treated as a binary image and then fed into the network, superior to only quantifying V-I trajectory features. Yet, such the binary image method could not well recognize different power sequences from distinct appliances with similar V-I trajectories [16].

Recurrence graph (RG) [16] and Markov transition field (MTF) [17] are other typical load features designed to convert current waveform into an image representation. Unlike the V-I trajectory, RG utilizes a recursive graph matrix to represent the current sequence through the similarities between samples in the sequence, which could distinguish appliances the V-I trajectory could not. However, the RG-based approach requires the conversion of the similarity matrix to a recursive graph matrix through threshold binarisation, which leads to information loss of the power data and may affect the performance of load disaggregation model. Since MTF needs to discretize continuous sequences to construct the state transfer matrix, MTF also faces an information loss problem similar to the RG.

Therefore, gramian angular field (GAF) has been proposed for NILM, to better recognize different appliances with similar trajectories [18,19]. GAF includes both gramian summation angular field (GASF) and gramian difference angular field (GADF), obtained by transforming sequence through the polar coordinates of the power data and encoding temporal correlations within different time intervals by considering the trigonometric sum/difference between each point of power sequence [20]. Therefore, based on GAF images containing temporal correlations information, load disaggregation model could obtain correlations feature information when extracting features from the GAF images [21]. Existing works have tried to transform electrical sequence into either GASF (e.g., in [18]) or GADF (e.g., in [19]) image. Yet, both GASF and GADF cannot provide all GAF features, and thus it is advocated to use both of them simultaneously [22], which in turn would enlarge model parameters with double-sized channel numbers, and increase the training time. Pan et al. construct a new GAF in distribution network topology identification named as gramian summation difference angular field (GASDF) [23], containing both GASF and GADF information and keeping the same image channel number as either GASF or GADF.

Inspired by Pan's work, with the image GASDF as the input, we construct an image baseline model. Although GASDF could well recognize different appliances with similar V-I trajectories, some important statistical characteristics (e.g., mean, standard deviation, maximum or minimum element) cannot be extracted from GASDF. Meanwhile, Zhang et al. in [24] showed that injecting prior knowledge into neural network could improve the performance in fault diagnosis, which motivates us to use prior knowledge to include statistical characteristics to improve the disaggregation. Thus, we next try to build a NILM model integrating GASDF with fused prior knowledge, to fully use statistical characteristics to disaggregate the aggregate power sequence into each appliance's sequence, while keeping the image channel number unchanged as compared to original GAF. The main contributions are as follows:

1. We build a non-intrusive load disaggregation model based on GASDF and prior knowledge fusion, to better disaggregate different appliances from the aggregate power sequence. Particularly, the model could increase the data description of each appliance's features and ensure disaggregation accuracy without increasing model depth.
2. To inject prior knowledge into the proposed baseline model, we next use DAE to build the expert feature encoder (EFE). In particular, the current aggregate power and statistical characteristics based on historical data from each appliance are mixed and then encoded as the expert feature. The encoder structure of EFE could pre-extract spatio-temporal features from the time series that are different from the ones extracted from the GASDF images, and such injection of expert features improves the effectiveness of the disaggregation model.
3. The GASDF's feature map and expert feature are with different orders of magnitude, and it is shown that such mismatch would incur the covariance shift in the forward propagation, thereby impeding the network training. We then develop a batch-normalization (BN)-based normalization fusion (NF) method to lower the disaggregation error caused by the distribution difference.
4. Extensive numerical experiments are executed to show the advantage of integrating GASDF with prior knowledge. First, in the baseline model, GASDF is superior to both GADF and GASF regarding mean absolute error (MAE) and root mean square error (RMSE) of each appliance's power sequence. Then, our proposed model, named ApplianceFilter, could improve the baseline one on the UK-DALE and REDD Dataset by 16.62% and 14.51% in the MAE, and by 16.27% and 15.72% in the RMSE, respectively.

The remainder is arranged as follows. Section 2 overviews the related work. Section 3 introduces the load disaggregation problem, the image-based load disaggregation model, and baseline model with GASDF. Section 4 introduces the prior knowledge fusion-based load disaggregation model. Section 5 presents the evaluation criteria, experimental results and experimental analysis, and this work is concluded in Section 6 with future work. *Notation*: Bold upper and lower case letters respectively denote tensors and vectors, respectively, where matrix is a special tensor; $[\cdot]^\top$ denotes the vector transpose.

2. Related work

Appliance load monitoring (ALM) includes both ILM and NILM [25]. ILM monitors each appliance via sensors, whereas NILM requires only a meter at the entrance of a residential house. Thus, NILM is a more practical solution that facilitates the spreading of load monitoring, and improvement of energy efficiency, due to its lower cost.

Hart et al. in [10] first studied NILM, which has attracted a lot of interests and become critical in the smart grid. NILM was original based on classical machine learning methods, e.g, k-nearest neighbor (KNN) [26], support vector machine (SVM) [27], and FHMM [11]. However, it poses challenges in the real-world residential electricity. For example, when the appliance number increases, the state space size and computational complexity in FHMM would grow exponentially. Moreover, limited by the model learning ability, these methods cannot process large volumes of data with complicated states.

Motivated by recent advances in deep learning and its powerful feature extraction ability, NILM is increasingly supported by deep learning methods. Ciancetta et al. in [28] used the temporal-frequency features and CNN to detect the appliance events. Ding et al. in [29] proposed to utilize multiscale-based convolution kernel to improve CNN in load disaggregation. Antoine et al. in [30,31] proposed a more efficient energy disaggregation approach based on variational autoencoder framework (VAE) with Unet structure, which base layer is convolution layer. The model uses the regularized latent space of the VAE to encode the features of the aggregated sequence, which improves the feature extraction. Meanwhile, the skip connection of Unet could further help the model reconstruct the sequence of single appliances from the aggregated sequence. Since CNN cannot extract temporal features, LSTM was used to treat long power sequences in [32]. Thus, probabilistic neural network (PNN) and LSTM were combined to identify the states of household appliances in [33], and CNN-LSTM was proposed to recognize appliance load status (either ON or OFF) using only two appliances in one house [34]. However, LSTM could cause problems such as vanishing gradients when processing long-time power sequences.

Recently, Transformer has become a popular method for processing sequences and has been applied to NILM [35,36]. Transformer-based

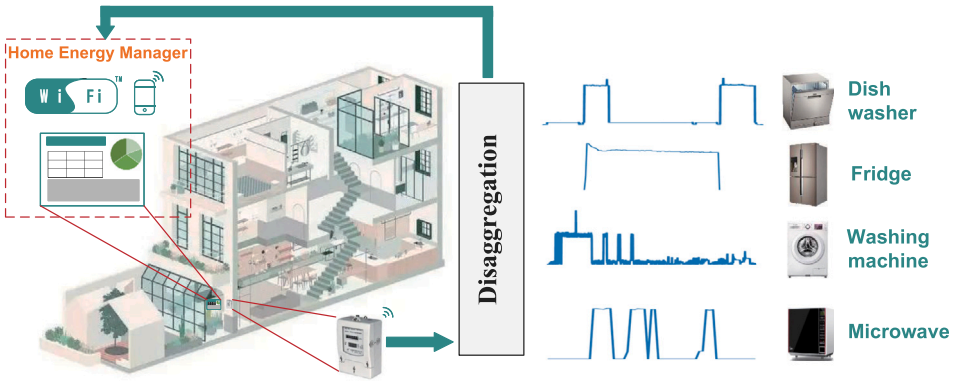


Fig. 1. Schematic diagram of NILM.

NILM methods use a self-attention mechanism to establish dependencies between positions in the input power sequences to better capture long-range dependencies, and related methods have yielded better results. However, Transformer models usually require many parameters to achieve their powerful modeling capabilities, leading to high storage and computational costs, especially for resource-constrained devices and environments [5]. Meanwhile, Transformer models usually require much training data to achieve good performance. For some low-resource languages or domain-specific tasks, obtaining sufficient training data may be challenging.

On the other hand, to find more identifiable features, high-frequency data (e.g., sampled in 44.1 kHz, 16 kHz, and 2 kHz) is favored [37], yet at the expense of larger storage space, wider bandwidth, and stronger processing capabilities. In addition to raw power sequence, other load data has also been examined, for instance, Lam et al. in [38] first studied the V-I trajectory, which is the reconstructed feature from current and voltage data. It should be noted that V-I methods use binary images to construct features, but could not distinguish appliances with similar V-I trajectories due to the normalized voltage and current data [14,15]. The authors in [18,19], and [21] have proposed a GAF image-based load recognition method to capture more features, yet at the cost of larger depth in neural network.

Almost all aforementioned works are built on end-to-end neural networks with only one sequence as the input, which learn the best layer parameters to improve the disaggregation. Nevertheless, in such networks, all values in the sequence must be processed at one time, and extracted features are probably adverse. The good and adverse features are learned by the model probabilistically. Thus, we try to design a multi-input load disaggregation model based on prior knowledge fusion, trained with low-frequency steady-state power data.

3. Preliminary

Nowadays, NILM is increasingly prevailing in energy management, bringing various approaches to improve disaggregation accuracy, and composed of two parts, i.e., data sampling and disaggregating, as shown in Fig. 1. More precisely, the smart meter would get its associated power values of each electrical appliance with the disaggregation method from the aggregate power, introduced as follows.

At moment t, let x_t, y_t^n and $a_t^n \in \{0,1\}$ be the aggregate power, nth appliance's power and running state of the nth appliance, respectively. If the nth appliance is running at time t, then $a_t^n = 1$; otherwise, $a_t^n = 0$. Let N be the total number of appliances, and thus the aggregate power at t becomes as

$$x_t = \sum_{n=1}^{N} a_t^n y_t^n + \varsigma_t, \tag{1}$$

with ς_t as the additive noise or measurement error.

Further, let $\mathbf{y}^n$ be the power sequence of nth appliance in period T, i.e., $\mathbf{y}^n = \left[y_1^n, y_2^n, \ldots, y_T^n\right]^\top$. If the aggregate power sequence $\mathbf{x} = \left[x_1, x_2, \ldots, x_T\right]^\top$ in period T is available, then we try to disaggregate $\mathbf{x}$ and recover the power sequence $\mathbf{y}^n$ for each appliance. That is, power disaggregation can be modeled as an optimization problem, finding the optimal N-dimensional vector $\hat{\boldsymbol{a}}_t^* = [a_t^{1*}, a_t^{2*}, \ldots, a_t^{N*}]^\top$ miniming the estimation error at moment t, i.e.,

$$\hat{\boldsymbol{a}}_t^* = \underset{\{a_t^n\}_{n=1}^N \in \{0,1\}^N}{\arg\min} \left| x_t - \sum_{n=1}^{N} a_t^n y_t^n \right|. \tag{2}$$

In particular, in deep learning, many mapping methods have emerged to solve (2), e.g., DAE, generative adversarial networks (GAN), LSTM, CNN, hidden Markov model (HMM), and its variants. The convolution layer of CNN and GAN, nevertheless, only focuses on local features and thus could not process long-term power sequences. On the contrary, LSTM could well process long-term power sequences, but is insensitive to local features [20].

To well extract the local features and simultaneously process long-term power sequence, the authors in [18,19,21] have proposed one end-to-end load disaggregation model using either GASF or GADF image, encoding one-dimensional power sequence into two-dimensional images with cosine/sine correlation and retaining correlation of power sequence sample points at different time intervals, thereby effectively retaining the power distribution law and providing correlation information [20]. Further, some works have emerged to integrate GASF with GADF to provide more features [20,22], but would incur overlarge model parameters with the double-sized channel number. Note that both GASF and GADF are symmetric matrices, and the information can be represented using either upper or lower triangular matrices. In distribution network topology identification, Pan et al. construct a new GAF, named as GASDF [23], containing both GASF and GADF information and keeping the same image channel number as either GASF or GADF. Motivated by Pan's work, we construct a GASDF for NILM with halves of both GASF and GADF, an entire feature is included, but without expanding model parameters. Thus, we advocate using the two-dimensional image GASDF (including both GASF and GADF), in which the upper-triangular and lower-triangular elements are cosine of two angle sum and sine of two angle difference [20], respectively. However, when the length of the input data is too long (i.e., the input length is T), the conversion to GAF would become an $T \times T$ sample, resulting in the cost of the model computation increasing. To address the disadvantage, we use a piecewise aggregate approximation (PAA), which could be viewed as a down-sampling approach, to make a short but representative abstraction for a long sequence. Let the length of the compressed sequence $\bar{\mathbf{x}} = \left[\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_T\right]^\top$ be G, where $G \leq T$. Then, each sampling point in $\bar{\mathbf{x}}$ could be calculated as follows:

$$\bar{x}_i = \frac{1}{k} \sum_{j=k(i-1)+1}^{k\times i} x_j, i = 1, 2, 3, \ldots, G, \tag{3}$$

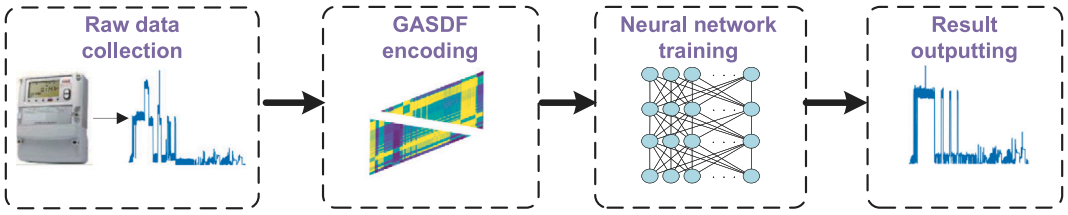


Fig. 2. Structure of proposed baseline model. The GASDF image encoding from aggregated power is the input of baseline model, and its output is disaggregated result.

Table 1
Parameters of the baseline model.

Layer	Kernel_size	Dilation_rate	Fileters/Nodes
Conv. layer 1	3	2	16
Conv. layer 2	3	2	16
Conv. layer 3	3	2	16
Conv. layer 4	3	1	16
Conv. layer 5	3	1	8
Conv. layer 6	3	1	8
Dense layer 1	–	–	T
Dense layer 2	–	–	T

where $k = \frac{T}{k}$ is the compression ratio. The calculation of dimension G will be set after EFE. Since the dimension G is related to the input sequence length T of the EFE, the detailed computation of G will be presented after the EFE in Section 4.1.2.

Next, the end-to-end load disaggregation model is constructed as the baseline one, as shown in Fig. 2. More precisely, four parts exist in the baseline model, used to collect the aggregate power data by meters, transform the aggregate power into GASDF image, construct a disaggregation neural network, and output the disaggregation result, respectively. Table 1 provides the parameters of baseline model, in which the activations of each two-dimensional convolution layer is LeakyReLU, dense layer 1's activation is Tanh, dense layer 2's activation is linear, and the first three two-dimensional convolution layers are dilated [39], thereby capturing more low-level features with larger receptive field. Dilated two-dimensional convolution could increase the receptive field but suffer from the "gridding" effect described in [40]. Typically, in deeper layers, sparse convolution kernels would be inefficient at covering the local information when the receptive field increases [41]. Therefore, the last three two-dimensional convolution layers are standard. Meanwhile, the input of baseline model is $\mathbf{x}$ visualized by GASDF.

Fig. 3 illustrates the encoding of GASDF from the aggregate power sequence. More precisely, the GASDF construction involves three steps, i.e., scaling, polar coordinate representation, and gramian matrix encoding. First, power sequence $\mathbf{x}$ needs to be normalized between 0 and 1 as follows:

$$\tilde{x}_t = \frac{x_t - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})}, \tag{4}$$

where $\min(\cdot)$ and $\max(\cdot)$ denote the minimum and maximum elements in sequence $\mathbf{x}$, respectively. For simplicity, define $\tilde{\mathbf{x}} = \left[\tilde{x}_1, \ldots, \tilde{x}_T\right]^\top$ as the normalized power sequence, which is next transformed into polar coordinate by encoding the value as angular cosine, i.e.,

$$\phi_t = \arccos(\tilde{x}_t). \tag{5}$$

Meanwhile, for $\tilde{x}_{t_1}, \tilde{x}_{t_2}$, when using the cosine of two angle sum, the quasi-inner product is defined as

$$\begin{aligned}\left\langle \tilde{x}_{t_1}, \tilde{x}_{t_2} \right\rangle &= \cos(\phi_{t_1} + \phi_{t_2}) \\ &= \cos(\phi_{t_1})\cos(\phi_{t_2}) - \sin(\phi_{t_1})\sin(\phi_{t_2}) \\ &= \cos(\phi_{t_1})\cos(\phi_{t_2}) - \sqrt{1-\cos^2(\phi_{t_1})}\sqrt{1-\cos^2(\phi_{t_2})} \\ &= \tilde{x}_{t_1}\tilde{x}_{t_2} - \sqrt{1-\tilde{x}_{t_1}^2}\sqrt{1-\tilde{x}_{t_2}^2}\end{aligned}. \tag{6}$$

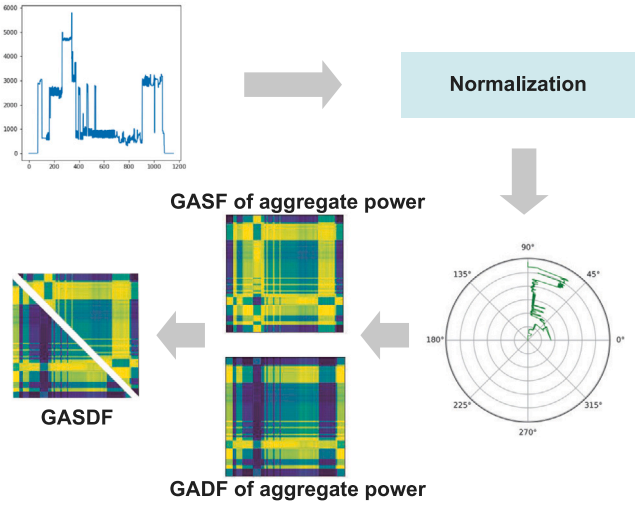


Fig. 3. Process of GASDF image encoding from aggregate power. The aggregated power is first encoded as GASF and GADF, then construct the GASDF by combining both GASF and GADF.

Besides, when using the sine of the difference of two angles, the quasi-inner product is defined as

$$\begin{aligned}\left\langle \tilde{x}_{t_1}, \tilde{x}_{t_2} \right\rangle &= \sin(\phi_{t_1} - \phi_{t_2}) \\ &= \sin(\phi_{t_1})\cos(\phi_{t_2}) - \cos(\phi_{t_1})\sin(\phi_{t_2}) \\ &= \sqrt{1-\cos^2(\phi_{t_1})}\cos(\phi_{t_2}) - \cos(\phi_{t_1})\sqrt{1-\cos^2(\phi_{t_2})} \\ &= \sqrt{1-\tilde{x}_{t_1}^2}\tilde{x}_{t_2} - \tilde{x}_{t_1}\sqrt{1-\tilde{x}_{t_2}^2}\end{aligned}. \tag{7}$$

Till now, the GASF and GADF encoding are respectively encoded as

$$\begin{aligned}\mathrm{GASF} &= \left[\cos(\phi_{t_1} + \phi_{t_2})\right]_{t_1,t_2=1}^{T} \\ &= \begin{pmatrix} \cos(\phi_1+\phi_1) & \cos(\phi_1+\phi_2) & \ldots & \cos(\phi_1+\phi_T) \\ \cos(\phi_2+\phi_1) & \cos(\phi_2+\phi_2) & \ldots & \cos(\phi_2+\phi_T) \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\phi_T+\phi_1) & \cos(\phi_T+\phi_2) & \ldots & \cos(\phi_T+\phi_T) \end{pmatrix} \\ &= \tilde{\mathbf{x}}\tilde{\mathbf{x}}^\top - \mathbf{s}\mathbf{s}^\top\end{aligned}, \tag{8}$$

and

$$\begin{aligned}\mathrm{GADF} &= \left[\sin(\phi_{t_1} - \phi_{t_2})\right]_{t_1,t_2=1}^{T} \\ &= \begin{pmatrix} \sin(\phi_1-\phi_1) & \sin(\phi_1-\phi_2) & \ldots & \sin(\phi_1-\phi_T) \\ \sin(\phi_2-\phi_1) & \sin(\phi_2-\phi_2) & \ldots & \sin(\phi_2-\phi_T) \\ \vdots & \vdots & \ddots & \vdots \\ \sin(\phi_T-\phi_1) & \sin(\phi_T-\phi_2) & \ldots & \sin(\phi_T-\phi_T) \end{pmatrix} \\ &= \mathbf{s}\tilde{\mathbf{x}}^\top - \tilde{\mathbf{x}}\mathbf{s}^\top\end{aligned}, \tag{9}$$

with $\mathbf{s} = \left[\sqrt{1-\tilde{x}_1^2}, \sqrt{1-\tilde{x}_2^2}, \ldots, \sqrt{1-\tilde{x}_T^2}\right]^\top$.

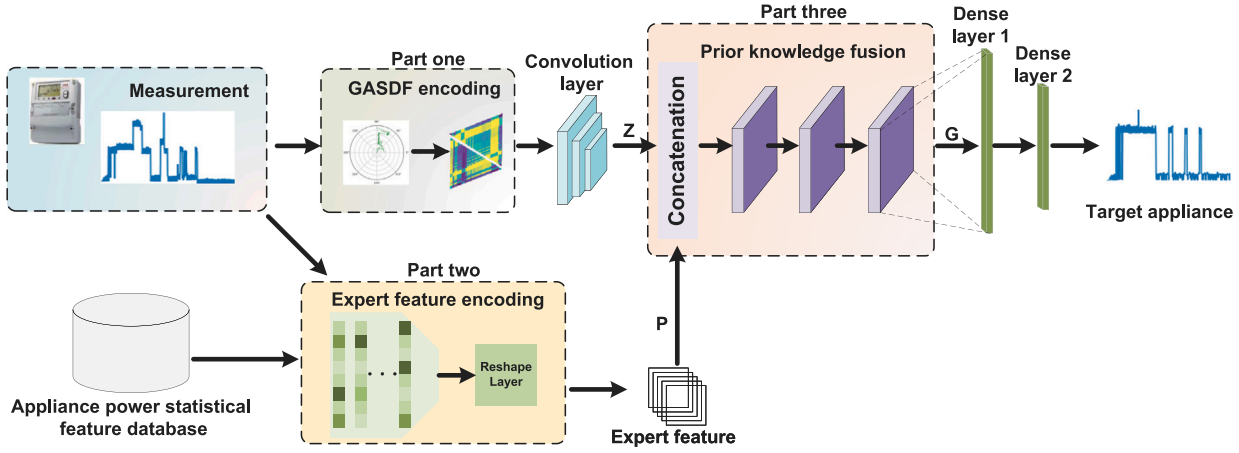


Fig. 4. Architecture of ApplianceFilter. Part one: This part is a GASDF encoding to process aggregated power, and then the GASDF feature map **Z** is got through the convolution layer. Part two: This part constructs an expert feature **P** from aggregated power and appliance statistical feature. Part three: This part fuses the **P** and **Z**. The proposed baseline model is constructed by part one, the convolution layer, part three, and two dense layers, as shown in Fig. 2.

Since both GASD and GADF are symmetric, to involve the information of both matrices with the least amount of data, we only need to retain half of both. Besides, since the diagonal elements in GADF are all zero, we have to take those in GASF as the diagonal of GASDF. Till now, the GASDF is constructed with dimension $G \times G$ as follows.

$$\left\langle \tilde{x}_{t_1}, \tilde{x}_{t_2} \right\rangle = \begin{cases} \tilde{x}_{t_1}\tilde{x}_{t_2} - \sqrt{1-\tilde{x}_{t_1}^2}\sqrt{1-\tilde{x}_{t_2}^2}, & \forall t_1 \le t_2 \\ \sqrt{1-\tilde{x}_{t_1}^2}\tilde{x}_{t_2} - \tilde{x}_{t_1}\sqrt{1-\tilde{x}_{t_2}^2}, & \forall t_1 > t_2 \end{cases} = \begin{cases} \cos(\phi_{t_1} + \phi_{t_2}), & \forall t_1 \le t_2 \\ \sin(\phi_{t_1} - \phi_{t_2}), & \forall t_1 > t_2 \end{cases} \tag{10}$$

4. Prior knowledge-based load disaggregation model

To get the disaggregated power for individual appliances from the aggregate power, a prior knowledge load disaggregation network, named as ApplianceFilter is proposed in this section, which includes GASDF encoding, prior knowledge encoding, and prior knowledge fusion.

4.1. Framework

4.1.1. Framework overview

Proposed ApplianceFilter is shown in Fig. 4, where the raw aggregate power **x** is sampled by a meter at the entrance of a residential house, and the nth target appliance's power sequence $\mathbf{y}^n$ is got from historical data. First, in part one, we encode **x** to GASDF as baseline model. Second, in part two, **x** and $\mathbf{y}^n$'s statistical characteristics are imported into the prior encoder to pre-extract the embedding vector E, taken as the prior knowledge, and next reshaped as an expert feature. Third, in part three, a fusion module is used to merge expert feature into the baseline model. Finally, the fusion feature is transformed into two dense layers.

4.1.2. Expert feature encoder

DAE is a neural network-based load disaggregation model, reconstructing a target appliance from a noisy input (taken as the background noise from other appliances) [9]. However, the raw power's features collected by meters are limited. Fortunately, statistical characteristics and pre-extracted features of current aggregate power could be fused into fusion features as prior knowledge, and imported into the baseline model, improving the disaggregating ability [24]. In this work, we try to build EFE, including pre-trained DAE's encoder and reshape layer. EFE is responsible for encoding **x**, and statistical characteristics of $\mathbf{y}^n$ as the expert feature, and obtaining prior knowledge to realize the multidimensional representation of power [9,42].

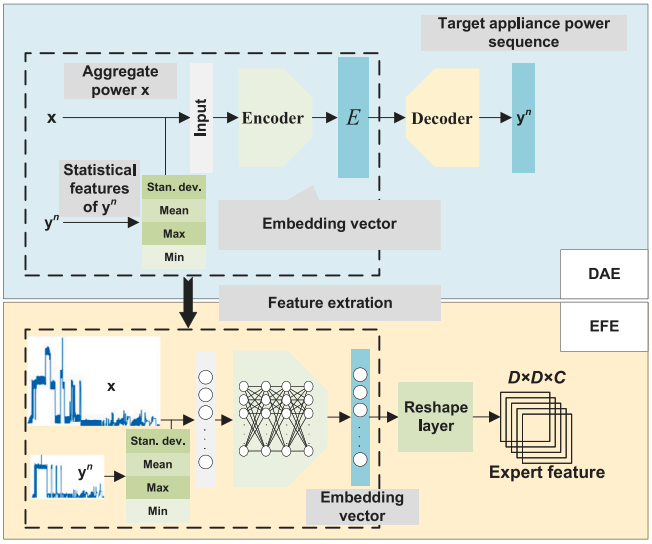


Fig. 5. Training process of expert feature encoder. The pre-trained DAE consists of CNN, BiLSTM, and dense layers. Depending on the encoder of DAE and reshape layer, EFE is constructed to get the expert feature.

More precisely, EFE includes two steps, as shown in Fig. 5. First, different from traditional DAE [9] respectively using **x** and $\mathbf{y}^n$ as input and output, we pre-trained DAE by using **x** and statistical characteristics of $\mathbf{y}^n$ as input. Meanwhile, unlike the convolution-based encoder, we construct the encoder using one CNN, one BiLSTM, and two dense layers. This structured encoder could extract spatio-temporal features, which solves the problem that the CNN layer-based encoder can only extract local spatial features in the sequences, and provides richer information for expert features. Second, the reshape layer (to change the dimension of E) follows pre-trained DAE's encoder (to extract E) to achieve the EFE.

In the model, the size of GASDF is related to the input length of EFE. Let the dimension of E be $T \times C$, with C as the filter number. Recall that the GASDF's dimension is $G \times G$. After going through three convolution layers of the Convolution layer module, the dimension of GASDF's feature map **Z** becomes $D \times D \times C$, with $D = G/8$. Meanwhile, to minimize the information loss of prior knowledge fusion, set $D = \sqrt{T}$. Till now, the expert feature **P** with dimension $\sqrt{T} \times \sqrt{T} \times C$ is obtained

after the reshape layer, with $\mathbf{P}$ as the output of EFE. Therefore, the G will be set as $8 * \sqrt{T}$.

4.1.3. Prior knowledge fusion

Our propsed baseline model, as shown in Fig. 2, can be composed of five parts in Fig. 4: Part one, Convolution layer, Part three, Dense layer 1 and 2. Then, joining Part two on the basis of baseline model forms the ApplianceFileter model, as shown in Fig. 4. ApplianceFileter uses the $\mathbf{P} \in \mathbb{R}^{\sqrt{T}\times\sqrt{T}\times C}$ and $\mathbf{Z} \in \mathbb{R}^{\sqrt{T}\times\sqrt{T}\times C}$ rather than baseline model only using $\mathbf{Z}$ to train the Part three and it subsequent layers. As such, if the disaggregation model is without prior knowledge, then the whole model is only a baseline model with the input GASDF that we proposed.

The fusion module is shown in Fig. 6, whcih is the Part three of Fig. 4. Suppose that the new high-dimensional feature $\mathbf{Q} := [\mathbf{Z}, \mathbf{P}]$ consists of a concatenation of $\mathbf{P}$ and $\mathbf{Z}$. To fuse the features of $\mathbf{P}$ and $\mathbf{Z}$, $\mathbf{Q}$ would be fed into the fusion module composed of three convolution layers, i.e. $\mathbf{M}(\mathbf{Q}) = \mathbf{F}_3(\mathbf{F}_2(\mathbf{F}_1(\mathbf{Q})))$, with $\mathbf{F}_i(\cdot)$ as the ith ($i = 1, 2, 3$) convolution layer. Note that, to distinguish from the Convolution layer module in Fig. 4, we call the convolution layer in Part three the weight layer.

Therefore, the calculation process of the fusion module without adding normalization fusion (NF) is as follows. Let the activation function and output of weight layer i be $\sigma_i(\cdot)$ and $\mathbf{O}^i = [\mathbf{O}^i_1, \ldots, \mathbf{O}^i_n, \ldots, \mathbf{O}^i_N]$, respectively, where $\mathbf{O}^i_n \in \mathbb{R}^{\sqrt{T}\times\sqrt{T}}$ is the nth channel of $\mathbf{O}^i$, and $\sigma_i(\cdot)$'s output is $\mathbf{A}^i = \sigma_i(\mathbf{O}^i)$. Meanwhile, let $\mathbf{W}^i \in \mathbb{R}^{3\times3\times M_i\times N_i}$ is weight layer i's parameters, where M_i and N_i are channel and kernel numbers, respectively. Each channel is a two-dimensional sub-matrix $\mathbf{W}^i_{m,n}$ of the kernel $W^i_n \in \mathbb{R}^{3\times3\times M_i}$, $1 < m < M_i$ and $1 < n < N_i$. Taking weight layer 1 as one instance, its kernel convolution output is

$$\mathbf{O}^1_n = \sum_{m=1}^{M_i} \mathbf{Q}_m * \mathbf{W}^1_{m,n} + b, \tag{11}$$

where $*$ is the convolution operation, b is bias, and $\mathbf{Q}$ is one three-dimensional tensor concatenated by $\mathbf{Z}$ and $\mathbf{P}$, i.e., $\mathbf{Q} = [\mathbf{Q}_1, \mathbf{Q}_2, \ldots, \mathbf{Q}_m, \ldots, \mathbf{Q}_{M_1}] \in \mathbb{R}^{\sqrt{T}\times\sqrt{T}\times 2C}$. Then, $\mathbf{W}^1$ is trained through each kernel's sub-matrix as follows [43].

$$\mathbf{W}^1_{m,n} = \mathbf{W}^1_{m,n} - \alpha \frac{\partial L}{\partial \mathbf{W}^1_{m,n}}, \tag{12}$$

where L is the ApplianceFilter's loss function, and α is the learning rate. More especially, $\frac{\partial L}{\partial \mathbf{W}^1_{m,n}}$ could be calculated as

$$\frac{\partial L}{\partial \mathbf{W}^1_{m,n}} = \frac{\partial L}{\partial \mathbf{O}^1_n} * \mathbf{Q}_m. \tag{13}$$

Note that, to obtain $\frac{\partial L}{\partial \mathbf{A}^1_n}$, we have to get the derivative of L w.r.t. $\mathbf{A}^1_n$ $\left(\mathbf{A}^1_n = \sigma_1\left(\mathbf{O}^1_n\right)\right)$ as follows:

$$\frac{\partial L}{\partial \mathbf{A}^1_n} = \sum_{n=1}^{N_i} \left(\mathrm{rot180}\left(\mathbf{W}^2_{m,n}\right) * \frac{\partial L}{\partial \mathbf{O}^2_n} \right), \tag{14}$$

where rot180 means to make matrix flip up and down first, and then rotate left and right. Take the derivative of each element in $\mathbf{A}^1_n$ w.r.t. its associated element in $\mathbf{O}^1_n$, and we define the matrix of derivative as $\sigma_1'(\mathbf{O}^1_n) := \left[\frac{\mathrm{d}\left(\mathbf{A}^1_n\right)_{t_1,t_2}}{\mathrm{d}\left(\mathbf{O}^1_n\right)_{t_1,t_2}} \right]_{t_1,t_2=1}^{\sqrt{T}\times\sqrt{T}}$. Then the derivative of L w.r.t. $\mathbf{O}^1_n$ is

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{O}^1_n} &= \sigma_1'(\mathbf{O}^1_n) \odot \frac{\partial L}{\partial \mathbf{A}^1_n} \\ &= \sigma_1'\left(\mathbf{O}^1_n\right) \odot \sum_{n=1}^{N_i} \left(\mathrm{rot180}\left(\mathbf{W}^2_{m,n}\right) * \frac{\partial L}{\partial \mathbf{O}^2_n} \right), \end{aligned} \tag{15}$$

with $\odot$ as the Hadamard product. And for each convolution weight layer, the derivatives of L w.r.t. $\mathbf{O}^i_n$ is

$$\frac{\partial L}{\partial \mathbf{O}^i_n} = \sigma_i'\left(\mathbf{O}^i_n\right) \odot \sum_{n=1}^{N_i} \left(\mathrm{rot180}\left(\mathbf{W}^{i+1}_{m,n}\right) * \frac{\partial L}{\partial \mathbf{O}^{i+1}_n} \right). \tag{16}$$

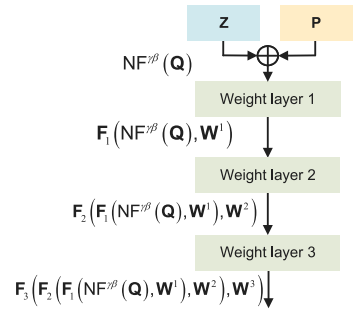


Fig. 6. Prior knowledge fusion module. This module is the part three of Fig. 4.

Finally, $\frac{\partial L}{\partial \mathbf{W}^1_{m,n}}$ can be computed through (13) and (16), as shown in (17).

$$\frac{\partial L}{\partial \mathbf{W}^1_{m,n}} = \left(\sigma_1'\left(\mathbf{O}^1_n\right) \odot \sum_{n=1}^{N_i} \left(\mathrm{rot180}\left(\mathbf{W}^2_{m,n}\right) * \frac{\partial L}{\partial \mathbf{O}^2_n} \right) \right) * \mathbf{Q}_m. \tag{17}$$

Nevertheless, $\mathbf{P} \in \mathbb{R}^{\sqrt{T}\times\sqrt{T}\times C}$ and $\mathbf{Z} \in \mathbb{R}^{\sqrt{T}\times\sqrt{T}\times C}$ are typically output by different convolution layers, and thus the elements of $\mathbf{P}$ and $\mathbf{Z}$ are with diverse orders of magnitude and different distributions. From (12) and (17), when training, $\mathbf{Q}$ is input into weight layer 1, which would affect $\mathbf{W}^1$ and L by $\mathbf{Q}$'s diverse orders of magnitude and distributions. If $\mathbf{Q}$ is not normalized, then the "covariate shift" would occur [44], thereby affecting the disaggregation model accuracy. Therefore, based on batch normalization (BN), we propose a normalization fusion (NF) method to lower the disaggregation error caused by the distribution difference. Assume q_{ijk}, $\mu_{\mathbf{Q}}$ and $\delta^2_{\mathbf{Q}}$ are the (i, j, k)th element, sample mean, and sample variance of $\mathbf{Q}$, respectively.

First, the $\mathbf{Q}$ sample mean is reached by

$$\mu_{\mathbf{Q}} = \frac{1}{2TC} \sum_{i,j,k=1}^{\sqrt{T},\sqrt{T},2C} q_{ijk}. \tag{18}$$

Then, the $\mathbf{Q}$ fusion sample variance is met by

$$\delta^2_{\mathbf{Q}} = \frac{1}{2TC - 1} \sum_{i,j,k=1}^{\sqrt{T},\sqrt{T},2C} \left(q_{ijk} - \mu_{\mathbf{Q}}\right)^2. \tag{19}$$

Afterwards, the $\mathbf{Q}$ fusion normalization is got by

$$\hat{q}_{ijk} = \frac{q_{ijk} - \mu_{\mathbf{Q}}}{\sqrt{\delta^2_{\mathbf{Q}} + \varepsilon}}, \tag{20}$$

where ε is a constant added to the variance for numerical stability. However, simply normalizing $\mathbf{Q}$ to the standard normal distribution would change $\mathbf{Q}$'s original distribution and lose the information [44]. Therefore, the normalized data needs to be scaled and shifted appropriately in the fusion module, by introducing a pair of parameters γ_{ijk} and β_{ijk} as

$$\hat{o}_{ijk} = \gamma_{ijk}\hat{q}_{ijk} + \beta_{ijk}. \tag{21}$$

Till now, we refer the NF concatenating as

$$\mathrm{NF}^{\gamma\beta}_{ijk}\left(q_{ijk}\right) := \gamma_{ijk}\hat{q}_{ijk} + \beta_{ijk}. \tag{22}$$

Yet, when scaled and shifted, we need to calculate the value of γ_{ijk} and β_{ijk} for each element, thus incurring a huge computation. Thus, a sharing pair of γ and β is preferred, rather than per-element pair [44], and $\hat{\mathbf{Q}}$'s scaled and shifted result is shown as

$$\mathrm{NF}^{\gamma\beta}(\mathbf{Q}) := \gamma\hat{\mathbf{Q}} + \beta, \tag{23}$$

with $\hat{\mathbf{Q}} = \left[\hat{q}_{ijk}\right]^{\sqrt{T},\sqrt{T},2C}_{i,j,k}$.

Finally, note that each weight layer is a four-dimensional tensor composed of N_i convolution kernels. From Table 1, we can get $\mathbf{W}^2 \in \mathbb{R}^{3\times3\times C\times C/2}$ and $\mathbf{W}^3 \in \mathbb{R}^{3\times3\times C/2\times C/2}$, and the fused feature is generated from fusion module as follows:

$$\mathbf{G} = \mathbf{F}_3\left(\mathbf{F}_2\left(\mathbf{F}_1\left(\mathrm{NF}^{\gamma\beta}(\mathbf{Q}), \mathbf{W}^1\right), \mathbf{W}^2\right), \mathbf{W}^3\right), \tag{24}$$

which is input into the dense layer, as shown in Fig. 4.

4.2. Advantages

Prior knowledge-based load disaggregation model is an improved baseline one by incorporating the expert feature.

4.2.1. Comparison between ApplianceFilter and DAE

The load disaggregation model aims to recover the target appliance's power sequence from aggregate power. The disaggregation error of DAE largely depends on the input power sequence length. The baseline model's GASDF image encoding obtains a power sequence's two-dimensional image with temporal correlation information, thus recognizing temporal correlation between any two samples far apart in long-term power sequences. Meanwhile, in ApplianceFilter, the expert feature is imported into baseline model by encoding statistical characteristics and current aggregate power, where statistical characteristics include mean, standard deviation, maximum and minimum elements of power sequence.

4.2.2. Comparison between ApplianceFilter and image-based model

The baseline model is an image-based one, which fails to obtain deep information due to its shallow model with finite layers to extract deep features. Most image-based models are deep, e.g., AlexNet in [21]. Yet, it poses challenges to regulate overfitting. Thus, based on the pre-trained DAE, we construct the EFE to obtain a multi-dimensional as prior knowledge. As such, ApplianceFilter not only obtains the features contained in the GASDF image, but also uses prior knowledge extracted by EFE to improve the shallow network's feature extraction ability.

5. Experiments and results

In this section, we first show the used dataset, then present the metrics for experimental evaluation, and finally compare prosed ApplianceFilter with other methods.

5.1. Data

We use two public available data set REDD [45] and UK-DALE [46] to validate the proposed method. REDD records six U.S. residential houses' power consumption from the main meter as well as several submeters. The sampling rates of main consumption and individual appliances consumption are 1 Hz and 1/3 Hz, respectively. To align the aggregated and individual power consumption data of REDD, we downsample the power consumption data to 1/3 Hz. UK-DALE records five U.K. households' power consumption data from main meter and submeters. Each appliance's power is sampled every 6 s, while the aggregate power is sampled every 1 s. Thus, we need to downsample all power data to 6 s rate. Both the two data set use the aggregate consumption power and target individual appliance consumption power as training data and label, respectively.

REDD and UK-DALE datasets include various appliances with different power levels. To verify the proposed method, both REDD and UK-DALE datasets are divided into training and testing data, with the details shown in Table 2. In REDD, we choose three types of appliances: microwave, fridge, and dishwasher in households 1–5 as the training data. All three appliances' data in household 6 are used as the testing data. In UK-DALE, we choose four types of appliances, i.e., dishwasher (DW), fridge (FG), microwave (MW), and washing machine (WM). All four appliances' data in household 5 are used as the testing data.

Table 2
Households used for training and testing.

Appliances	UKDALE		REDD	
	Training household	Testing household	Training household	Testing household
DW	1,2	5	1,2,3,4,5	6
FG	1,2,4	5	1,2,3,4,5	6
WM	1,2	5	–	–
MW	1,2	5	1,2,3,4,5	6

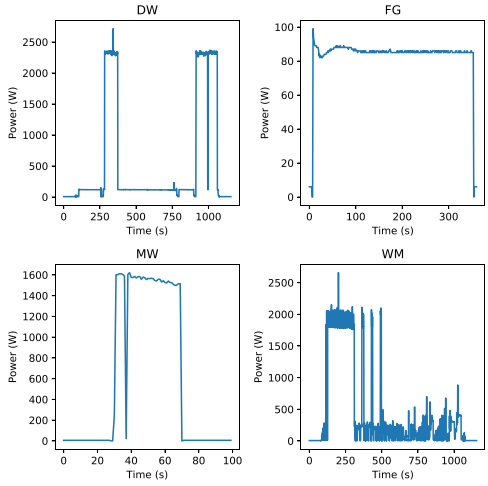


Fig. 7. Activation of four appliances in UK-DALE dataset.

Table 3
Window length used for different appliances in REDD and UK-DALE dataset.

Appliance	UK-DALE	REDD
DW	1156	1296
FG	361	400
MW	100	144
WM	1156	–

Moreover, UK-DALE metadata shows that WM and MW share a single meter in household 4, DW, FG, MW, and WM do not have a record in household 3, and household 4 does not have a record of DW. Therefore, DW, FG, MW, and WM of household 3 and DW, WM, and MW of household 4 would not be involved in training. Meanwhile, the same appliance varies in power demands in different houses, and we only study common states of appliances.

To obtain the samples and labels, we need to first locate the "on" period of an appliance. And each appliances' "on" period (appliance activations) are extracted by using the *get_activations*() function of NILM toolkit (NILMTK) [47]. Taking the four appliances of UK-DALE as an example, the activations of the appliances are shown in Fig. 7. Then, the target appliance activation power sequence $\mathbf{y}^n = \left[y_1^n, y_2^n, \ldots, y_T^n\right]^\top$ is used as a label. And the aggregate power sequence $\mathbf{x} = \left[x_1, x_2, \ldots, x_T\right]^\top$ and the target appliance activation power sequence are paired as training data during the same time index. Since the activation power sequence lengths are unequal, we need to tailor them to a fixed window length. Therefore, we use the same window length setting method as [9,48], and the window length set for UK-DALE and REDD are shown in Table 3.

Before starting to train the model, we use the max–min normalization to preprocess each pair of training data and label to facilitate model training.

Table 4
Hyperparameter value.

	Appliance	LR	Optimizer	BS	Epoch	LF
UK-DALE	DW	0.001	Adam	20	100	MAE
	FG	0.0001	SGD	25	100	MAE
	MW	0.01	SGD	30	90	MSE
	WM	0.0001	SGD	30	100	MAE
REDD	DW	0.0001	SGD	30	100	MAE
	FG	0.001	Adam	20	90	MAE
	MW	0.00001	Adam	30	100	MAE

5.2. Evaluation

To evaluate the performance of ApplianceFilter, we adopt RMSE and MAE as metrics, which have been widely used to evaluate NILM methods [16].

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(y_t^n - \hat{y}_t^n\right)^2}, \tag{25}$$

and

$$\mathrm{MAE} = \frac{1}{T}\sum_{t=1}^{T}\left|y_t^n - \hat{y}_t^n\right|, \tag{26}$$

where y_t^n and $\hat{y}_t^n$ represent the raw target power and disaggregate power of the nth appliance, respectively at time t, and T denotes the length of power sequence.

5.3. Hyperparameter optimization

The manually set framework parameters in the machine learning model are called hyperparameters. Setting appropriate hyperparameters could effectively reduce the model's error. Therefore, the model needs to use hyperparameter optimization methods to select an optimal subset of hyperparameters. Optuna [49] is an automated hyperparameter optimization framework designed especially for machine learning, with various optimization methods to choose from, such as grid search, stochastic search, and Bayesian optimization search. This paper uses Optuna-based tree-structured Bayesian optimization search TPE (Tree-structured Parzen Estimator) to search hyperparameters during hyperparameter optimization. Owing to the design of the model structure, the number of model layers has been fixed in this paper. Therefore, for each appliance model, we search for the primary hyperparameters, learning rate (LR), optimizer, batch size (BS), epoch, and loss function (LF). The primary hyperparameters and their importance rate obtained according to the Optuna optimization framework w.r.t each appliance's model are shown in Table 4, Figs. 8, and 9.

Figs. 8 and 9 illustrate that LR and the optimizer are the main parameters influencing the appliance model. The SGD optimizer is chosen for most appliance models during the optimization process. Although Adam uses the average of the historical gradients to accelerate convergence, gradient averaging makes Adam more prone to overfitting than SGD when the data changes significantly. The microwave requires different LRs, optimizers, and LFs when constructing the model on different datasets. The diverse operating states of microwaves lead to variable microwave data sequences in different datasets. Therefore, the hyperparameters of the model vary significantly during the training process. To converge the microwave model for different datasets, the LR needs to be tuned to control the degree of gradient influence to prevent model divergence. Load disaggregation model is a typical regression model in which LF is mainly chosen as MSE or MAE. However, when the data contains outliers, i.e. large or small values distant from the mean, the MAE is more suitable as a LF than the MSE. Therefore, further analysis of hyperparameters' effects on different home appliance models is needed in future studies.

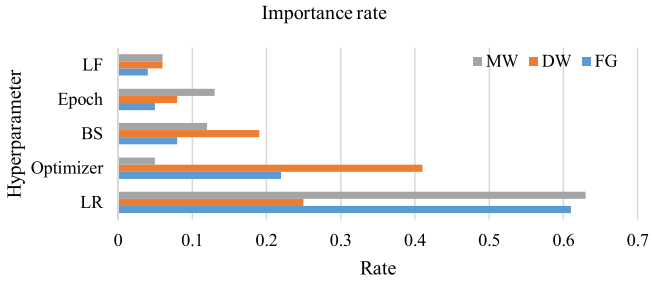


Fig. 8. Hyperparameter importance with REDD dataset.

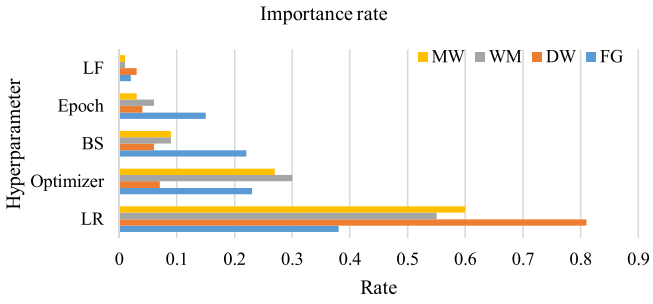


Fig. 9. Hyperparameter importance with UK-DALE dataset.

Table 5
Experiment environment.

Hardware environment		Software environment	
CPU	Intel (R) Core (TM) I5-10400F-CPU @ 2.90 ghz	TensorFlow	2.2.0
GPU	NVIDIA GeForce RTX 2060	NILMTK	0.4.3

Table 6
Comparison between GASF/GADF/GASDF on proposed baseline model in metrics with REDD dataset.

	Method	DW	FG	MW
MAE	GASF	16.31	25.50	179.21
	GADF	14.58	23.91	174.06
	GASDF	**14.18**	**22.93**	**171.03**
RMSE	GASF	52.32	51.43	375.47
	GADF	46.46	48.44	368.93
	GASDF	**44.45**	**48.05**	**353.47**

5.4. Comparison

We first verify the influence of different image features on the proposed baseline model, second compares baseline model with other models, then verify the disaggregation effect of ApplianceFilter on each target appliance, and implement an ablation experiment. At last, the computational complexity of all the models is compared. The experiment environment is summarized in Table 5, and results are listed in Tables 6–14, with best shown in bold.

Since GASF, GADF, and GASDF images contain different information, the disaggregation results of baseline model using different images are not same. The groups of results from three different images are shown in Tables 6 and 7. GASDF-based baseline model has less MAE and RMSE in all appliances, and this is because GASDF contains both GASF and GADF features, but without expanded model parameters, thereby saving the training time. Therefore, GASDF suits more for load disaggregation.

Next, the baseline model is compared with four existing methods (DAE in [50], BiLSTM in [51], Unet based VAE [30] and MTT [35]),

Table 7
Comparison between GASF/GADF/GASDF on proposed baseline model in metrics with UK-DALE dataset.

	Method	DW	FG	MW	WM
MAE	GASF	68.11	25.46	303.64	154.53
	GADF	63.84	23.24	354.39	147.20
	GASDF	**61.99**	**19.98**	**270.11**	**145.54**
RMSE	GASF	195.19	48.14	460.92	280.28
	GADF	187.75	42.32	525.52	271.98
	GASDF	**184.29**	**36.20**	**399.84**	**260.80**

Table 8
Comparison between Baseline, DAE, BiLTSM, MTT and Unet based-VAE models with REDD dataset.

	Method	DW	FG	MW	Overall
MAE	DAE [50]	36.28	27.35	235.54	299.17
	BiLSTM [51]	25.78	35.92	**76.28**	**136.98**
	Unet based-VAE [30]	84.49	32.83	443.88	561.2
	MTT [35]	101.35	100.35	439.27	640.97
	Baseline model	14.18	22.93	171.03	208.14
	ApplianceFilter	**12.99**	**22.59**	142.36	177.94
RMSE	DAE [50]	111.91	59.07	488.66	659.64
	BiLSTM [51]	83.83	62.65	**204.12**	**350.6**
	Unet based-VAE [30]	196.50	54.81	730.19	981.5
	MTT [35]	285.19	128.10	845.80	1259.09
	Baseline model	44.45	48.05	353.47	445.97
	ApplianceFilter	**38.16**	**46.64**	291.06	375.86

with results shown in Tables 8 and 9. DAE outperforms BiLSTM on DW and FG, but not on MW and WM. This is because the inputs to the model are temporally correlated power sequences generated by MW and WM, which undergo multiple state changes during their activation. As a CNN-based one, nevertheless, DAE could only extract local features, and thus could not process long-term power sequences and recognize the temporal correlation between any two states far apart. On the contrary, BiLSTM could better process temporal information with the larger memory, but could not extract local features. Though also based on CNN, the baseline model gets least MAE and RMSE in three appliances (i.e., DW, FG, and WM), only inferior to BiLSTM in MW, which could be explained as follows. Constructed by both GASF and GADF, GASDF could use trigonometric sum/difference between any two sample points to recognize the temporal correlations [20]. Overall, in most appliances, our baseline model outperforms the four most popular methods in NILM without prior knowledge.

Next, we further improve the baseline model by means of prior knowledge fusion, to also get the least metrics of the model overall. The results of ApplianceFilter (w. prior knowledge) and baseline model (w.o. prior knowledge) are both shown in Tables 8 and 9. After prior knowledge fusion, ApplianceFilter is superior to the baseline model overall, reducing MAE by 16.62% and RMSE by 16.27% in UK-DALE dataset, and reduceing MAE by 14.51% and RMSE by 15.72% in REDD dataset. Then, to further verify the effectiveness of EFE, we added it for each of the compared methods and tested them on REDD and UK-DALE, respectively, and the results are shown in Tables 10 and 11. Compared with the results without EFE in Tables 8 and 9, adding EFE effectively improves the disaggregation performance of most methods and reduces the error generated by the disaggregation. Afterwards, the disaggregate power sequences of DW, WM, FG, and MW are separately shown in Figs. 10–13.

As shown in Fig. 10, DW includes two states (cleaning and drying), and its state switching is procedural during an activation. With Tables 8 and 9, the MAE of ApplianceFilter is close to the baseline model, and the RMSE outperforms the baseline model. MAE is un-hypersensitive to high disaggregation bias values, while RMSE is the opposite. Furthermore, the RMSE would be significant when the disaggregation results have a much high bias value, while the MAE would be small due to the tolerance on high bias values. Therefore, Tables 8, 9 and Fig. 10

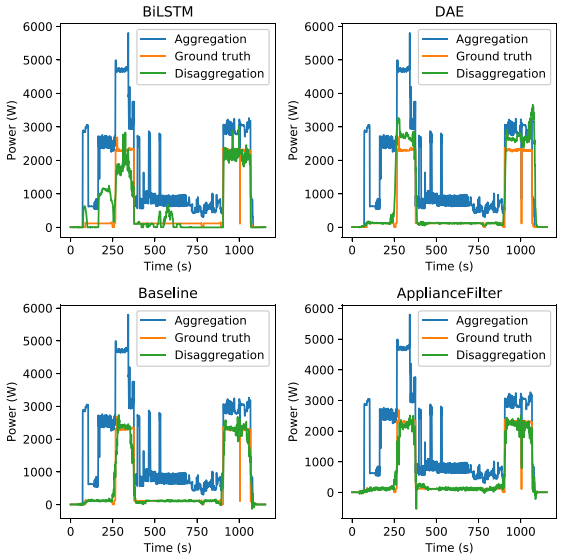


Fig. 10. DW's disaggregate power sequences in UK-DALE dataset.

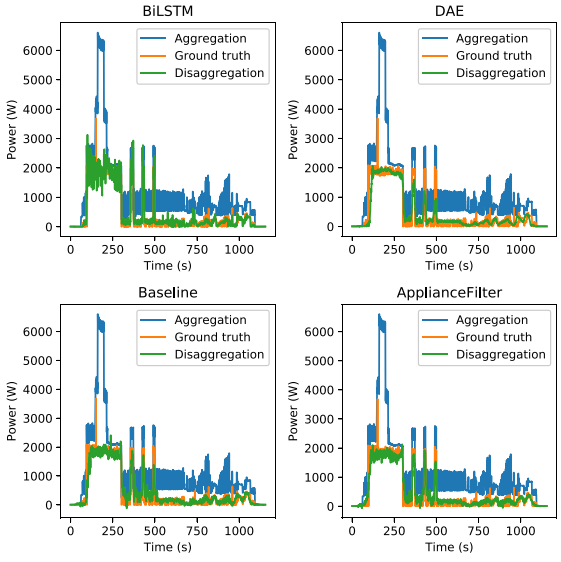


Fig. 11. WM's disaggregate power sequences in UK-DALE dataset.

illustrate that the model could reduce the generation of high bias values in the disaggregation results.

In Table 9, the MAE and RMSE values of WM disaggregation result on ApplianceFilter are significantly lower than the baseline model. This is because, as shown in Fig. 11, the state switching of WM is also procedural, including three states (i.e., washing, rinsing, and drying). However, the power variation of WM in different states is significant and not easily learned by the model. EFE could easily obtain a prior knowledge from the total power sequence and statistical features. The load decomposition method of a prior knowledge fusion is more advantageous for WMs than DWs.

As shown in Fig. 12, ApplianceFilter for FG is better than its baseline model. On MAE, the ApplianceFilter and baseline model are similar. Whereas, on RMSE, ApplianceFilter and baseline model differ significantly due to the more high bias values in baseline's disaggregation results. This is because FG is typically with lower power, easy to be masked by other high-power appliances, and thus it is difficult

Table 9
Comparison between Baseline, DAE, BiLTSM, MTT and Unet based-VAE models with UK-DALE dataset.

	Method	DW	FG	MW	WM	Overall
MAE	DAE [50]	138.22	20.88	386.09	240.06	785.25
	BiLSTM [51]	152.01	30.67	**122.34**	166.24	471.26
	Unet based-VAE [30]	322.63	14.68	194.58	332.35	864.24
	MTT [35]	318.23	**12.61**	314.51	250.92	896.27
	Baseline model	**61.99**	19.98	270.11	145.54	497.62
	ApplianceFilter	68.54	18.66	191.72	**135.98**	**414.9**
RMSE	DAE [50]	338.33	52.37	578.33	493.69	1462.72
	BiLSTM [51]	294.22	41.38	**251.47**	369.28	956.35
	Unet based-VAE [30]	543.21	**20.73**	308.74	580.95	1453.27
	MTT [35]	491.34	38.17	451.21	380.68	1361.40
	Baseline model	184.29	36.20	399.84	260.80	881.13
	ApplianceFilter	**171.39**	26.78	294.14	**245.40**	**737.71**

Table 10
Comparison between ApplianceFilter and EFE-attached DAE, EFE-attached BiLTSM, EFE-attached MTT and EFE-attached Unet-based VAE models on the REDD dataset.

	Method	DW	FG	MW
MAE	DEA-EFE	27.89	28.54	145.01
	BiLSTM-EFE	18.40	26.71	**54.89**
	Unet based-VAE-EFE	83.41	25.20	199.51
	MTT-EFE	99.99	78.46	380.44
	ApplianceFilter	**12.99**	**22.59**	142.36
RMSE	DEA-EFE	76.06	59.68	259.32
	BiLSTM-EFE	71.05	50.33	**174.07**
	Unet based-VAE-EFE	221.98	50.60	368.75
	MTT-EFE	282.66	107.82	745.00
	ApplianceFilter	**38.16**	**46.64**	291.06

Table 11
Comparison between ApplianceFilter and EFE-attached DAE, EFE-attached BiLTSM, EFE-attached MTT and EFE-attached Unet-based VAE models on the UK-DALE dataset.

	Method	DW	FG	MW	WM
MAE	DEA-EFE	89.58	19.23	150.91	**120.18**
	BiLSTM-EFE	180.15	15.52	**103.21**	134.79
	Unet based-VAE-EFE	449.47	14.67	206.29	136.52
	MTT-EFE	205.57	**11.95**	525.41	419.84
	ApplianceFilter	**68.54**	18.66	191.72	135.98
RMSE	DEA-EFE	199.51	51.86	**231.54**	**214.10**
	BiLSTM-EFE	348.29	22.24	232.76	262.45
	Unet based-VAE-EFE	778.88	**20.72**	316.17	229.26
	MTT-EFE	533.37	32.12	792.22	790.36
	ApplianceFilter	**171.39**	26.78	294.14	245.40

to extract prior knowledge in FG's power sequence. What is more, the extracted prior knowledge is interfered by other appliances, thus incurring larger high bias values than baseline model.

As shown in Fig. 13, ApplianceFilter outperforms baseline model in MW. Since different foods are heated for different durations in the MW, the running is non-procedural in one activation, thus asking for more data for training. Fortunately, due to prior knowledge being more targeted for each appliance, it could compensate for the training data of MW [24]. Although MW is non-procedural, it only has a general heating state of different sequence lengths, thus the statistical characteristics can also be coded in the prior knowledge to improve ApplianceFilter.

The computational complexity in terms of model size, time to train a batch, and inference time for one sample of all the models for the appliance "dishwasher" is calculated and reported in Table 12. It is found that the proposed model has the most parameters and the least inference time compared to other models, which is because the last two Dense layers of the proposed model contribute over 10M of parameters, while the total number of parameters in the remaining layers is less than 1.4M. Since the ApplianceFilter has a convolutional structure in all layers except the last two, it has an excellent parallelism capability. Meanwhile, given that the total number of ApplianceFilter's layers is much smaller than MTT and Unet based-VAE, the proposed

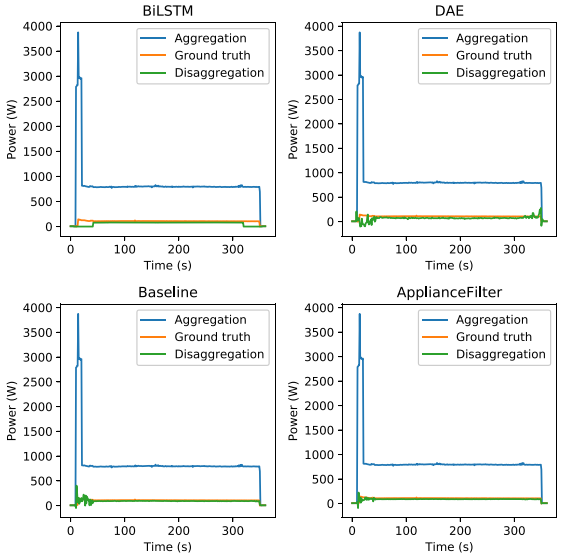


Fig. 12. FG's disaggregate power sequences in UK-DALE dataset.

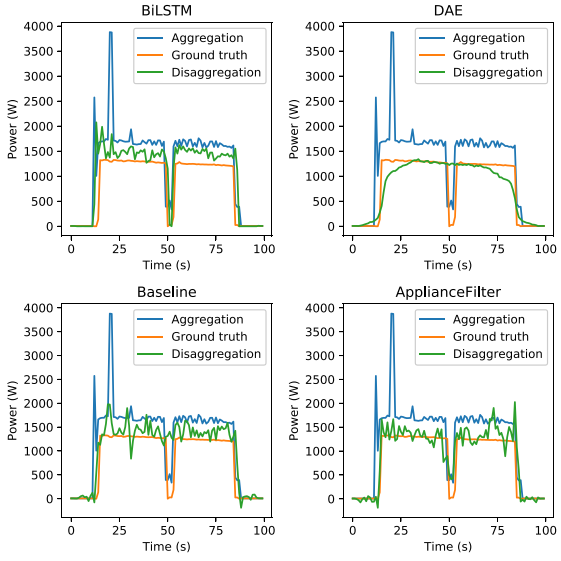


Fig. 13. MW's disaggregate power sequences in UK-DALE dataset.

model could exhibit faster processing time than MTT and Unet based-VAE during training and validation. Despite the most parameters, the proposed model is still the most suitable solution for the following reasons. First, the inference time is critical, which is the foundation

Table 12
Comparison of computational complexity of various models.

	DAE	LSTM	Unet based-VAE	MTT	Proposed baseline model	ApplianceFileter
Model size (Mb)	1.2	1.2	3.1	1.85	12	12
Per-batch training time (ms)	6	169	176	246	18	21
Per-sample inference time (ms)	0.005	0.03	0.017	67.04	0.01	0.081

Table 13
Ablation research results of the ApplianceFilter on UK-DALE dataset.

Metrics	ApplianceFilter				ApplianceFilter without NF				ApplianceFilter without expert feature			
	DW	FG	MW	WM	DW	FG	MW	WM	DW	FG	MW	WM
MAE	68.54	18.66	191.72	135.98	76.51	17.75	215.13	141.49	61.99	19.98	270.11	145.54
RMSE	171.39	26.78	294.14	245.40	211.99	23.66	331.63	252.14	184.29	36.20	399.48	260.88

Table 14
Comparison between ApplianceFilter and Multiscale CNN.

	Method	DW	FG	MW	WM
MAE	Multi-scale model [48]	118.10	39.33	243.71	**55.90**
	Our method	**68.54**	**18.66**	**191.72**	135.98

for ensuring real-time performance. Second, the model provides better performance in the overall load disaggregation.

An ablation study is performed to evaluate the contribution of the NF and expert features to the model's overall performance. One experiment involves directly removing the NF operation after P and Z connections. Another experiment is to remove part two from the ApplianceFilter. Those experiments would help identify the contribution of NF and expert features. The experiment results of ablation study are illustrated in Table 13. From the experiments, both NF and expert features impact ApplianceFilter. This is because expert features could provide a new power feature representation for the model obtained through pre-trained load disaggregation tasks and NF could help the model reduce distribution differences in the feature fusion process.

Furthermore, in Table 14, when comparing our model with the multiscale model in [48] using the MAE, our disaggregation results in DW, FG and MW are better than the multiscale model. However, the disaggregation results of the WM are not satisfactory, since the receptive field size of the multiscale convolution kernel is not unique, which can better extract different scale features in power sequences. In general, the multiscale model has a better disaggregation effect in multi-state and low-power appliances. Therefore, multiscale methods can be further integrated into the prior knowledge fusion model to provide more insightful information for more features.

6. Conclusion

This study proposed a load disaggregation model based on both prior knowledge fusion and GASDF image. First, we converted aggregate power sequence to GASDF, and then built a baseline model. In the baseline model, GASDF could provide all GAF features (including GASF and GADF) and temporal correlation of power sequence sample points at different time intervals, facilitating the extraction of correlation features by the convolution layer from GASDF. Although results revealed that GASDF-based baseline model has better disaggregation results, the error for MW is yet too large, some important statistical characteristics (e.g., mean, standard deviation, maximum and minimum elements) cannot be extracted from GASDF. Moreover, MW's heating state duration is not fixed in one activation, thus asking for more power data for training. Therefore, to compensate for the less training data, we used prior knowledge fusion to provide more expert features for baseline model, and thus proposed a new model named ApplianceFilter. Through experiments, we found that the prior knowledge fusion model could improve the effect of load disaggregation. As compared to exising multiscale model, ApplianceFilter performs better in DW, FG and MW, but not in WM, because the multiscale model can provide convolution kernels of different scaled receptive field, thereby improving the feature extraction ability in the convolution-based model. Thus, the multiscale model has a better disaggregation result in multi-state appliances. In future work, we would try to integrate the multiscale model into ApplianceFilter to provide more multiscale features.

CRediT authorship contribution statement

Dong Ding: Writing – original draft, Methodology, Data curation, Conceptualization. **Junhuai Li:** Writing – review & editing, Supervision, Methodology, Funding acquisition. **Huaijun Wang:** Supervision, Funding acquisition. **Kan Wang:** Writing – review & editing, Writing – original draft, Funding acquisition. **Jie Feng:** Writing – review & editing, Validation. **Ming Xiao:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

References

[1] Kelly J, Knottenbelt W. Does disaggregated electricity feedback reduce domestic electricity consumption? A systematic review of the literature. In: Proc. international workshop on non-intrusive load monitoring. 2016, p. 1–5.
[2] Gopinath R, Kumar M, Joshua C, Srinivas K. Energy management using non-intrusive load monitoring techniques—state-of-the-art and future research directions. Sustain Cities Soc 2020;62:102411.
[3] Zhang Y, Qian W, Ye Y, Li Y, Tang Y, Long Y, et al. A novel non-intrusive load monitoring method based on ResNet-seq2seq networks for energy disaggregation of distributed energy resources integrated with residential houses. Appl Energy 2023;349:121703.
[4] Dash S, Sahoo NC. Electric energy disaggregation via non-intrusive load monitoring: A state-of-the-art systematic review. Electr Power Syst Res 2022;213:108673.
[5] Christos LA, Theofilos AP, Dimitrios ID. Real-time non-intrusive load monitoring: A light-weight and scalable approach. Energy Build 2021;253:111523.
[6] Dong Y, Wang L, Wang J, Hu X, Zhang H, Yu F, et al. Accelerating wireless federated learning via Nesterov's momentum and distributed principle component analysis. IEEE Trans Wirel Commun 2023;1–15. http://dx.doi.org/10.1109/TWC.2023.3329375, early access.
[7] Li J, Wang R, Wang K. Service function chaining in industrial internet of things with edge intelligence: a natural actor-critic approach. IEEE Trans Ind Inf 2023;19(1):491–502.

[8] Nalmpantis C, Vrakas D. Machine learning approaches for non-intrusive load monitoring: from qualitative to quantitative comparation. Artif Intell Rev 2019;52(2):217–43.
[9] Kelly J, Knottenbelt W. Neural nilm: Deep neural networks applied to energy disaggregation. In: Proc. ACM international conference on embedded systems for energy efficient built environments. 2015, p. 55–64.
[10] Hart G. Nonintrusive appliance load monitoring. Proc IEEE 1992;80(12):1870–91.
[11] Yan L, Tian W, Han J, Li Z. eFHMM: Event-based factorial hidden markov model for real-time load disaggregation. IEEE Trans Smart Grid 2022;13(5):3844–7.
[12] Wang A, Chen B, Wang C, Hua D. Non-intrusive load monitoring algorithm based on features of V-I trajectory. Electr Power Syst Res 2018;157:134–44.
[13] Hassan T, Javed F, Arshad N. An empirical investigation of V-I trajectory-based load signatures for non-intrusive load monitoring. IEEE Trans Smart Grid 2014;5(2):P870–878.
[14] Liu Y, Wang X, You W. General optimization technique for high-quality community detection in complex networks. IEEE Trans Smart Grid 2019;10(5):5609–19.
[15] Du L, He D, Harley RG, Habetler TG. Electric load classification by binary voltage-current trajectory mapping. IEEE Trans Smart Grid 2016;7(1):358–65.
[16] Faustine A, Pereira L, Klemenjak C. Adaptive weighted recurrence graphs for appliance recognition in non-intrusive load monitoring. IEEE Trans Smart Grid 2021;12(1):398–406.
[17] Estebsari A, Rajabi R. Single residential load forecasting using deep learning and image encoding techniques. Electronics 2020;9(1):68–85.
[18] Matindife L, Sun Y, Wang Z. Image-based mains signal disaggregation and load recognition. Complex Intell Syst 2022;7(2):901–27.
[19] Chen J, Wang X. Non-intrusive load monitoring using gramian angular field color encoding in edge computing. Chin J Electron 2022;32:1–9.
[20] Wang Z, Oates T. Imaging time-series to improve classification and imputation. In: Proc. international joint conference on artificial intelligence. 2015, p. 3939–45.
[21] Chen J, Wang X, Zhang X, Zhang W. Temporal and spectral feature learning with two-stream convolutional neural networks for appliance recognition in NILM. IEEE Trans Smart Grid 2022;13(1):762–72.
[22] Dias D, Dias U, Menini N, Lamparelli R, Maire GL, Torres RdS. Image-based time series representations for pixelwise eucalyptus region classification: a comparative study. IEEE Geosci Remote Sens Lett 2020;17(8):1450–4.
[23] Pan Y, Qin C. Identification method for distribution network topology based on two-stage feature selection and gramian angular field. Autom Electr Power Syst 2022;46(16):170–7.
[24] Zhang T, Chen J, He S, Zhou Z. Prior knowledge-akugmented self-supervised feature learning for few-shot intelligent fault diagnosis of machines. IEEE Trans Ind Electron 2022;69(10):10573–84.
[25] Liu G, Liu J, Zhao J, Qiu J, Wu Z, Mao Y, et al. Real-time corporate carbon footprint estimation methodology based on appliance identification. IEEE Trans Ind Inf 2023;19(2):1401–12.
[26] Figueiredo M, Almeida AD, Ribeiro B. Home electrical signal disaggregation for non-intrusive load monitoring (NILM) systems. Neurocomputing 2012;96:66–73.
[27] Le T-T-H, Kim H. Non-intrusive load monitoring based on novel transient signal in household appliances with low sampling rate. Energies 2018;11(12):1–35.
[28] Ciancetta F, Bucci G, Fiorucci E, Mari S, Fioravanti A. A new convolutional neural network-based system for NILM applications. IEEE Trans Instrum Meas 2021;79:1–12.
[29] Ding D, Li J, Zhang K, Wang H, Wang K, Cao T. Non-intrusive load monitoring method with inception structured CNN. Appl Intell 2022;52(6):1–18.
[30] Antoine L, Cheriet M, Ghyslain G. Efficient deep generative model for short-term household load forecasting using non-intrusive load monitoring. Sustain Energy Grids Netw 2023;34:101006.
[31] Antoine L, Marc-André C, Mohamed C, Ghyslain G. Energy disaggregation using variational autoencoders. Energy Build 2022;254:111623.
[32] Le T-T-H, 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(4):3252–64.
[33] Zhou Z, Xiang Y, Xu Yi Z, Shi D, Wang Z. A novel transfer learning-based intelligent nonintrusive load-monitoring with limited measurements. IEEE Trans Instrum Meas 2021;70:1–12.
[34] Kundu A, Juvekar GP, Davis K. Deep neural network based non-intrusive load status recognition. In: Proc. clemson university power systems. 2018, p. 1–6.
[35] Dash S, Sahoo NC. Attention-based multitask probabilistic network for nonintrusive appliance load monitoring. IEEE Trans Instrum Meas 2023;72:1–12.
[36] Shan Z, Si G, Qu K, Wang Q, Kong X, Tang Y, et al. Multiscale self-attention architecture in temporal neural network for nonintrusive load monitoring. IEEE Trans Instrum Meas 2023;72:1–12.
[37] Schirmer P, Mporas I. Double fourier integral analysis based convolutional neural network regression for high-frequency energy disaggregation. IEEE Trans Emerg Top Comput Intell 2022;6(3):439–49.
[38] Lam H, Fung G, Lee W. A novel method to construct taxonomy electrical appliances based on load signatures. IEEE Trans Consum Electron 2007;53(2):653–60.
[39] Zhang Z, Wang X, Jung C. DCSR: dilated convolutions for single image super-resolution. IEEE Trans Image Process 2019;28(4):1625–35.
[40] Wang P, Chen P, Yuan Y, Liu D, Huang Z, Hou X, et al. Understanding convolution for semantic segmentation. In: Proc. IEEE winter conference on applications of computer vision. 2018, p. 1451–60.
[41] Zhang Z, Wang X, Jung C. DCSR: Dilated convolutions for single image super-resolution. IEEE Trans Image Process 2019;28(4):1625–35.
[42] Al-Hmouz R, Pedrycz W, Balamash A, Morfeq A. Logic-oriented autoencoders and granular logic autoencoders: developing interpretable data representation. IEEE Trans Fuzzy Syst 2022;30(3):869–77.
[43] Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE 1998;86(11):2278–324.
[44] Ioffe S, Christian S. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proc. international conference on machine learning. 2015, p. 762–72.
[45] Kolter J Zico, Johnson Matthew J. REDD: A public data set for energy disaggregation research. In: Workshop on data mining applications in sustainability, vol. 25. Citeseer; 2011, p. 59–62.
[46] 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(1):1–14.
[47] Batra N, Kelly J, Parson O, Dutta H, Knottenbelt W, Rogers A, et al. NILMTK: an open-source toolkit for non-intrusive load monitoring. In: Proc. ACM international conference on future energy systems. 2014, p. 265–76.
[48] Zhou G, Li Z, Fu M, Feng Y, Wang X, Huang C. Sequence-to-sequence load disaggregation using multiscale residual neural network. IEEE Trans Instrum Meas 2021;70:1–10.
[49] Takuya A, Shotaro S, Toshihiko Y, Takeru O, Masanori K. Optuna: A next-generation hyperparameter optimization framework. In: Proc. ACM SIGKDD international conference on knowledge discovery and data mining. Association for Computing Machinery; 2019, p. 2623–31.
[50] García-Pérez D, Pérez-López D, Díaz-Blanco I, González-Muñiz A, Domínguez-González M, Vega A. Fully-convolutional denoising auto-encoders for NILM in large non-residential buildings. IEEE Trans Smart Grid 2021;12(3):2722–31.
[51] Tongta A, Chooruang K. Long short-term memory (LSTM) neural networks applied to energy disaggregation. In: Proc. IEEE international electrical engineering congress. 2020, p. 1–4.