RESEARCH   **Open Access**

PUMPNET: a deep learning approach to pump operation detection

Luyao Ma, Qingyu Meng, Shirui Pan[*] and Ariel Liebman

*Correspondence:
shirui.pan@monash.edu
Faculty of IT, Monash University,
3800 Clayton, VIC, Australia

Abstract

Non-urgent high energy-consuming residential appliances, such as pool pumps, may significantly affect the peak to average ratio (PAR) of energy demand in smart grids. Effective load monitoring is an important step to provide efficient demand response (DR) to PAR. In this paper, we focus on pool pump analytics and present a deep learning framework, PUMPNET, to identify the pool pump operation patterns from power consumption data. Different from conventional time-series based Non-intrusive Load Monitoring (NILM) methods, our approach transfers the time-series data into image-like (date-time matrix) data. Then a U-shaped fully convolutional neural network is developed to detect and segment the image-like data in pixel level for operation detection. Our approach identify whether pool pumps operate given thirty-minute interval aggregated active power consumption data in kilowatt-hours only. Furthermore, the PUMPNET algorithm could identify pool pump operation status with high accuracy in the low-frequency sampling scenario for thousands of household, compared to traditional NILM algorithms which process high sampling rate data and can only apply to limited number of households. Experiments on real-world data validate the promising results of the proposed PUMPNET model.

Keywords: Electricity consumption monitoring, Demand side management, Pool pump, NILM, Deep learning, Convolutional neural network, Power segmentation

Introduction

Smart grids are modern electrical grids supplying electricity with monitoring and reacting to local demands, which result in an intelligent, high-efficient, and sustainable method in electricity delivery (Siano 2014). Also, smart grids face challenges from supply cost, sustainability, efficiency, availability, and reliability problems. Despite digitalizing the power generation, transmission and distribution of the current grid, active customer participation becomes a significant bi-directional communication in dynamic balanced smart grids (Aboulian et al. 2018).

Typically, there is a high peak to average power ratio (PAR) of energy demand in grids. Energy providers and operators usually adopt demand response (DR) programs to feedback the real-time load of customers, including direct load control programs, load curtailment programs, time of use pricing and real-time pricing (Jordehi 2019). These programs require centralized load monitoring performed to acquiring load sequence

(Malik et al. 2019). Hence, it is essential to identify load patterns for reasonable demand response programs' design.

The ceiling value of peak load would result in remarkable investments on electricity supply equipment, advanced capacity and quality requirements on grids and serious emission problems. Also, peak load time only occupies a few hours per day, and it is less efficient for partially loaded transformers at off-peak time (Zhu et al. 2012). Consequently, it is important to perform demand-side management (DSM) along with provider side demand response for electricity balancing. DSM target to encourage users to shift peak load consumption to off-peak time flattering the demand curve, then time pattern and magnitude of load could be shaped by customer engagement. Moreover, fuel cost by power generation fluctuation could be reduced, which contribute to exhaust gas emission reduction of generators (Ye et al. 2015). DR programs would also benefit from motivating customer engagement with rewards and price-based methods. Consequently, the PAR, total capacity and supply cost would be decreased; the grid efficiency would be increased also.

The pattern of customer-side electricity usage should be identified as prior information of DR and DSM design and implementation according to their benefits, especially non-urgent and shiftable appliances consumption pattern. Furthermore, customers would also be aware of their practical consuming model and restrict their usage behaviours by revealing facts on electricity usage.

The number of residential swimming pools has been raised to 10.4 million in the United States. To maintain the water clean and bacteria-free, pool pumps should be operated for three to eight hours a day. This long-time working duration leads to over 2,000 kilowatt-hours electricity consumption annually (Lopez et al. 2018). Despite the huge energy consumption, pool pumps usually start and end based on an electronic or mechanic timer with customized working intervals. This operation period commonly does not locate in the grid off-peak time. Hence, pool pumps would bring relatively heavy load at peak time to the grid. Consequently, it is possible to identify the pool pumps' working pattern for accurate residential load monitoring and smart grid energy delivery efficiency optimization with DR and DSM.

Non-intrusive load monitoring (NILM) is an effective method for appliance energy consumption disaggregation. NILM could utilize low-cost sensors integrated into the smart meter to provide real-time energy management and devices diagnostics for residential load monitoring (Aboulian et al. 2018). A series of machine learning algorithms would learn the appliances' working pattern automatically. NILM can be performed with neural network (Chang et al. 2013), fuzzy system (Lin and Tsai 2014), support vector machine (Saitoh et al. 2010), and other new algorithms (Guillén-García et al. 2019). Deep learning becomes popular in NILM research because of its strong ability in feature extraction from energy consumption sequential data. Deep learning NILM was realized via Convolutional Neural Networks (CNN) (Zhang et al. 2018), unsupervised learning (Gonçalves et al. 2011), denoising auto-encoder (DAE) and long-short-term memory network (LSTM) (Kelly and Knottenbelt 2015a).

Conventionally, NILM algorithms analyze high-frequency sampling energy consumption time-series data in terms of active power (P), reactive power (Q), current (I) and voltage (U). According to popular public NILM datasets summarized in (Faustine et al. 2017), the main scope of NILM methods are multiple appliances energy

disaggregation for several households (no more than 10) within (P, Q, I, V) features sampled at second-level. In such a context, appliances electricity consuming pattern would keep relatively consistent in fewer households. However, pool pumps' operation pattern varies depending on power rating, heating function, mechanic timer shifting, daylight saving time, temperature, water amount, and usage habits. Pumps among thousand households may not share solid demand time patterns.

To achieve our goal, we propose a pool pump operation detection network (PUMPNET) to overcome inconsistent patterns of thousands of device. We perform a data transformation from time-series to matrix representation to expose pool pump working features spatially including pre-order and post-order information in the time dimension and the date dimension. After the realignment, the fixed working period and time intervals of a pump could be revealed. For example, if a pump works from 10 am to 2 pm on every day in June as shown in Fig. 1a, there would be a higher electricity consumption on the data points representing these time and date interval, and a 10-pixel height and 30-pixel width high electricity consumption rectangle is bounded by the pool pump working pattern. Consequently, the higher energy consumption of pool pumps can be represented with a fixed rectangle bound, as shown in Fig. 1b. It's obvious that in the given case, the pump was set to work from 10 am to 2 pm until the 120th date. After that date, it's still running, but the configured time had been changed. Thus, the pump operation could be recognized as a rectangle segmentation problem in the field of computer vision. Then, we introduce a U-shaped end-to-end pixel classification model accepting power consumption matrix of each household. The model is trained with 2,988 residential records and evaluated on 996 residential records, which are ensured with swimming pool possession for all records. Finally, we can detect pool pump operation periods using our framework and the trained PUMPNET model with high performance.
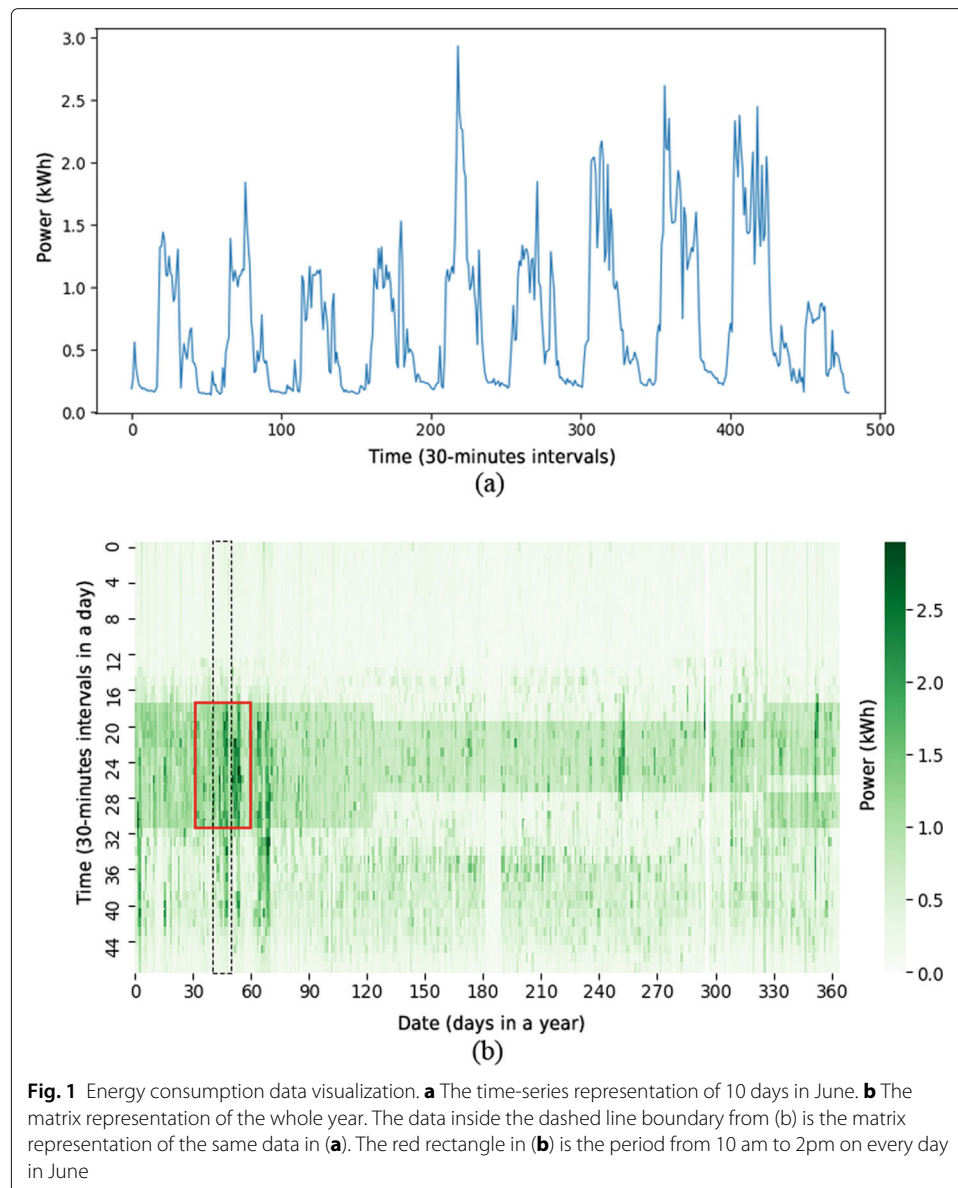
The main contributions of this paper are as follows:

- We propose a pool pump operation detection framework PUMPNET in 30-minute sampling context for thousands of household load monitoring purpose.
- We introduce matrix representation to highlight pool pumps operation feature: working in the same period for a long time, which could reduce impacts from inconsistent features among time-series sequences, such as power rating and usage habit varied in thousands of household.
- We introduce a semantic segmentation model to the energy analysis field, to classify each power consumption data point with spatial information in the time dimension and the date dimension, resulting in high performance on the pool pump operation detection problem.

The paper is structured as follows. "Related works" section discussed the related works. "Methodology" section describes our model PUMPNET and listed implementation details. In "Evaluation" section, we evaluated the PUMPNET and discussed case studies. Finally, "Conclusion" section concludes the whole paper.

Related works

In this section, we mainly focus on previous on-off detection deep learning models to demonstrate the inappropriateness of previous works to our scenario. Also, we would review matrix-based approaches in appliance analysis and semantic segmentation. Com-

Fig. 1 Energy consumption data visualization. **a** The time-series representation of 10 days in June. **b** The matrix representation of the whole year. The data inside the dashed line boundary from (b) is the matrix representation of the same data in (**a**). The red rectangle in (**b**) is the period from 10 am to 2pm on every day in June

bining matrix-based NILM context with semantic segmentation, we named this problem as power segmentation, which means to extract target operation pattern from background appliances consumption with matrix representation.

Time-series based operation detection models

The Neural NILM Rectangle model (Kelly and Knottenbelt 2015a) and Single Load Extraction model (Barsim and Yang 2018) two deep learning NILM models dealing with the time sequence of energy consumption for multiple appliances active power disaggregation. They are both build with UK-DALE dataset (Kelly and Knottenbelt 2015b) 1/6 Hz sampling active power time pattern.

Neural NILM rectangle model

The Neural NILM Rectangle model (Kelly and Knottenbelt 2015a) consists of two convolutional layers for feature extraction and five fully connected layers for regression. The

model could output three values, the start/end time offset and average power consumption, of each operation period for a single appliance. Target appliances of the algorithm are microwave, kettle, dishwasher, fridge and washing machine. These appliances all appear in at least three out of five households in the UK-DALE dataset. The training scope of the model is two houses and the test scope is one house. The ground truth on-off status was derived by NILMTK package (Batra et al. 2014) using a consecutive threshold power demand value. A threshold was set to filter short activations for noise reduction. Random windows (time periods) would be selected as an input sequence for model training. The target appliance of each window is determined by the first activation inside the interval. The outputs of the model are three regressed values indicating start time, end time and average power consumption of the target appliance. The start/end time points are represented by a proportion of the input window. Hence, the time values are always located inside $[0, 1]$. The target activation should be complete in each window as well.

Single load extraction model

The Single Load Extraction model (Barsim and Yang 2018) is a deep learning model with fully convolutional neural network predicting the on-off state of a single appliance for a 3-hour window sequential data. This model upsampled the aggregated 1/6 Hz real power in UK-DALE dataset to the 1 Hz time series using fill-forward methods. The preprocessing stage is similar to the Neural NILM Rectangle model: acquiring activations via NILMTK package with threshold filtering. After that, spikes would be reduced and generated activation labels are recognized as ground truth values. The model accepts a one-dimensional 10,800 length (3 hours 1 Hz data) energy consumption sequence as an input. The model has an asymmetric encoder-decoder structure extracting features by using dilated convolution (Yu and Koltun 2015). Dilated convolution could extend the receptive field comparing to original convolution process. Especially in the context of appliance monitoring, results of the dilated convolution contain the information carried by the distant data indicating on-off state transitions, because the home appliances on-off states tend to last greatly longer than the sample rate 1 second. As a result, the model output a one dimensional 10,800 length sequence containing "1" and "0" referring to on and off operation status.

Limitations

Above models address the electricity consumption disaggregation problem for 5 appliances in three households with sequential data and deep learning models. However, the model structure and feature extraction methods cannot be designed for a specific appliance considering the generalization ability of model structure on multiple devices. Consequently, only a relatively fixed sequential pattern can be effectively identified by previous models.

In our pump operation detection scenario, features of pumps vary among households, but the scope is one appliance only. As a result, we can utilize a novel feature representation method to highlight important common characters of thousands of pumps.

Matrix based appliance analysis and segmentation methods

Most of NILM analyses are with a high sampling rate, no greater than 1Hz. However, in this case, we have low-resolution data in 30-minute intervals only.

A 15-minute sampling rate pool pump analysis was stated in Burkhart et al. (2018). However, this research only predicts which household has a pool pump instead of detect operation time accurately. The data was converted into lower resolution (10 days and half-hour level) for noise reduction using morphological opening (Gonzalez et al. 2018). The opening processing is appropriate for pool pump ownership detection, but 90% information in the date dimension would be lost. Thus, in our 30-minute interval pool pump analysis, we extend this research with a convolutional neural network (CNN) which can perform accurate pixel-wise classification on a matrix representation.

This kind of pixel-wise classification problem is known as semantic segmentation task, which is very common in areas like medical image diagnosed. Semantic segmentation usually tries to recognise images in pixel level based on their semantic meaning, which is depending on surrounded pixels. In our case, the semantic is whether a pixel representing a time that a pool pump is operating. Deep convolutional neural networks (CNN) have contributed greatly to semantic segmentation tasks in recent years, due to strong spatial information extraction ability of convolutional computation and the complex architecture of deep neural networks models.

The very first end-to-end Fully Convolutional Network (FCN) was proposed in 2015 (Long et al. 2015). The model contains convolutional layers only. The model consists of eight convolutional layers as forward and inference part to compress the spatial features. Correspondingly, eight convolutional layers learn features extracted from inference part with upsampling to reform segmentation from dense feature maps. Also, skip connections are added in the model to combine original spatial features with reformed features. Finally, the model has been trained using a pixel-wise loss and measured with mean intersection over union (IOU) criterion.

Then, FCN was extended by U-Net (Ronneberger et al. 2015) for medical images segmentation task. The model could compute features in the contracting side and localize pattern pixel wisely in the expanding side. Similar to FCN, the contracting part extract features in different size using convolutional layers and max-pooling layers. The expanding part adopted deconvolutional layers to resize the feature map with trainable variables. Skip connections are kept from FCN to fuse high-level features with reconstructed features. Finally, the processed pixels was binary classified.

The U-Net architecture had been adopted and expanded widely then, such as Feature Pyramid Networks (FPN) (Lin et al. 2017), Pyramid Scene Parsing Network (PSPNet) (Zhao et al. 2017) and "DeepLabv3" (Chen et al. 2017). Although network structures and feature extraction methods are evolved for higher accuracy in complex image challenges, the U-shaped deep CNN architecture is still a classical base model for matrix-based segmentation problems.

Hence, we adopted the concept of a U-shaped model structure for a power segmentation problem in this paper. The model would performance binary classifications on each recorded electricity consumption data point. Finally, the pool pump operation pattern could be identified for further DR and DSM tasks in smart grids.

Methodology

In this section, the proposed PUMPNET is described in the 30-minute granularity electricity consumption segmentation context. Also, the classifier would be defined with its architecture and training hyper-parameters in detail. Beyond classical model structures,

we would also discuss the performance improvement methods on deep learning CNN models. Finally, these modifications would be compared in the "Evaluation" section, and the final PUMPNET would be formulated then.

An overview of the framework

The framework includes the whole processing of electricity consumption data from obtaining stage to results prediction on unseen data. The proposed framework is composed of the following four steps:

- **Data acquisition**: loading data and generating sequential data; completing missing values.
- **Feature representation**: converting time-series data into matrix representations, including raw data and ground truth of operation status of pool pumps.
- **Classifier modelling**: training the U-shaped deep CNN model with matrix representations and corresponding pool pump operation ground truth.
- **Pool pump operation detection**: applying the model on unseen data and generating prediction results; evaluating the identification results.

Data acquisition

The original data were collected by residential smart meters automatically. By recording the activate power consumption amount in sequential 30-minute intervals, time-series data can be generated and stored by the energy provider. The unit of electricity consumption data is kilowatt-hour (kWh).

Considering to the practical household energy usage habits, we complete missing values depending on the possible scenario: continuous missing values lasting for days would be filled with 0 indicating "away from home" and no energy consumed status; discrete missing values would be interpolated with forward-filling method indicating data transmission loss or recording loss.

For each household, the average aggregated power consumed when a pool pump is working is slightly over 0.9 kWh in all residential records. On the contrary, quite low kWh of electricity was consumed when the pool pump is off, 0.2 approximately. In the traditional methods, a fixed threshold would be applied based on the large difference of consumption amount to distinguish the on-off states. However, the accuracy of the threshold classification method is around 0.5.

The main factors degrading prediction performance are spikes from "background noise". In the appliance load monitoring scenario, an aggregated consumption data of devices $1, 2, \ldots, N$ at time t can be represented by:

$$P(t) = P_{pool\ pump}(t) + \sum_{i=1}^{N-1} P_i(t) + P_{noise}(t) \qquad (1)$$

, where $P_{pool\ pump}$ is the pool pump power consumption and $\sum_{i=1}^{N-1} P_i + P_{noise}$ presents power consumption of other appliances and random noise in the grid. As a result, the noise from other appliances and the grid would become a strong interference on the aggregated consumption value. For example, microwaves, clothes dryers, dishwashers, heaters and air conditioners may have different consumption pattern in high sampling scenario, but their electricity consumption amounts are similar to pool pumps' in coarse

sampling. As a result, noise data are indistinguishable from pool pump operation in the time-series data in terms of consumption amount of a single data point. Hence, we convert the representation of time-series data to a matrix in the next step revealing distinct features of pool pump operation rather than noise information.

Feature representation

The feature representation method is motivated by the nature of pool pump features, fixed working time interval and periodical operating in days. Pumps work as the core circulation system of the pool ensuring the water is clean and clear. Thus, this essential component is mostly controlled by a timer for daily periodic operations during the same time interval.

To highlight these features, we transformed the time-series data into a matrix representation visualized as Fig. 1. Ten days (480 30-minute intervals) energy consumption data in the line graph (a) only contains information in the squared area data of matrix (b). The fixed working period could be represented as horizontal edges in the figure, indicating that the on-off state of a high energy-consuming appliance is switched at the same time in consecutive days. Hence, it could be inferred as a timer controlled appliance. Also, the pump operation in each day is represented as a vertical line. With consecutive vertical lines and the same horizontal edges, the timer-controlled pool pumps operation pattern could be considered as a rectangle in an energy-consuming matrix. Energy-consumption amount of other appliances are represented as spikes in the matrix without shared state-shift attributes as pool pumps. Hence, the matrix representation can enrich the feature rendering of pool pumps rather than noises.

The matrix representation has 48 rows in 30-minute intervals and 365 columns in days. Each value indicates the power consumption in kilowatt-hour for the corresponding duration. According to the above discussion on patterns of pool pumps operation, We can convert the energy disaggregation problem into a "rectangle" power segmentation task with energy consumption matrix and strong noise. In the next step, we would introduce the segmentation model to address the load monitoring challenge.

Classifier modelling

According to the matrix representation and pool pump feature, our classifier is built concerning semantic segmentation methods. Comparing to other segmentation tasks, we have a larger dataset and simpler target pattern. Hence, we only kept the basic U-shaped structure in this case, and improve the U-Net in terms of the parameter amount, local feature fusion and avoiding training hamper and degradation problems. The proposed U-shaped network for pool pump operation detection as shown in Fig. 2.

The model accepts realigned matrix from the feature representation step. Also, model outputs share the same size as inputs: height 48, width 365 and channel 1, containing value 1 and 0 as operating status indicators.

The U-shaped model is defined as Algorithm 1. Compare to original U-Net, we reduce the number of the smallest feature map's channels from 1024 to 512 and the kernel size from 3x3 to 2x2. Also, we replace the two consecutive layers by convolution blocks. We adopt 4 types of CNN backbone networks: consecutive convolutional layers, residual blocks (He et al. 2016), dense blocks (Huang et al. 2017), and residual-dense blocks. Through experiments, we do extend the performance further with block replacement processing. The basic block structures are displayed in Fig. 3.

Algorithm 1: Algorithm of the U-shaped model

Input : Matrix data in 48(H) * 365(W) * 1(C)

Output: Binary classification mask

input ← *matrix*;

conv1 ← *conv_block_1*(*input*, *channels* = 64);

mp1 ← *MaxPooling*(*conv1*, *kernel_size* =[2, 2]);

conv2 ← *conv_block_2*(*mp1*, *channels* = 128);

mp2 ← *MaxPooling*(*conv2*, *kernel_size* =[2, 2]);

conv3 ← *conv_block_3*(*mp2*, *channels* = 256);

mp3 ← *MaxPooling*(*conv3*, *kernel_size* =[2, 2]);

conv4 ← *conv_block_4*(*mp3*, *channels* = 512);

mp4 ← *MaxPooling*(*conv4*, *kernel_size* =[2, 2]);

conv5 ← *conv_block_5*(*mp4*, *channels* = 512);

up1 ← *deconvolution*(*conv5*, *kernel_size* =[2, 2]);

conv6 ← *conv_block_6*(*crop&concat*(*up1*, *conv4*), *channels* = 512);

up2 ← *deconvolution*(*conv6*, *kernel_size* =[2, 2]);

conv7 ← *conv_block_7*(*crop&concat*(*up2*, *conv3*), *channels* = 256);

up3 ← *deconvolution*(*conv7*, *kernel_size* =[2, 2]);

conv8 ← *conv_block_8*(*crop&concat*(*up3*, *conv2*), *channels* = 128);

up4 ← *deconvolution*(*conv8*, *kernel_size* =[2, 2]);

conv9 ← *conv_block_9*(*crop&concat*(*up4*, *conv1*), *channels* = 64);

logits ← *output_conv_layer*(*conv9*, *channels* = 1, *kernel_size* =[1, 1]);
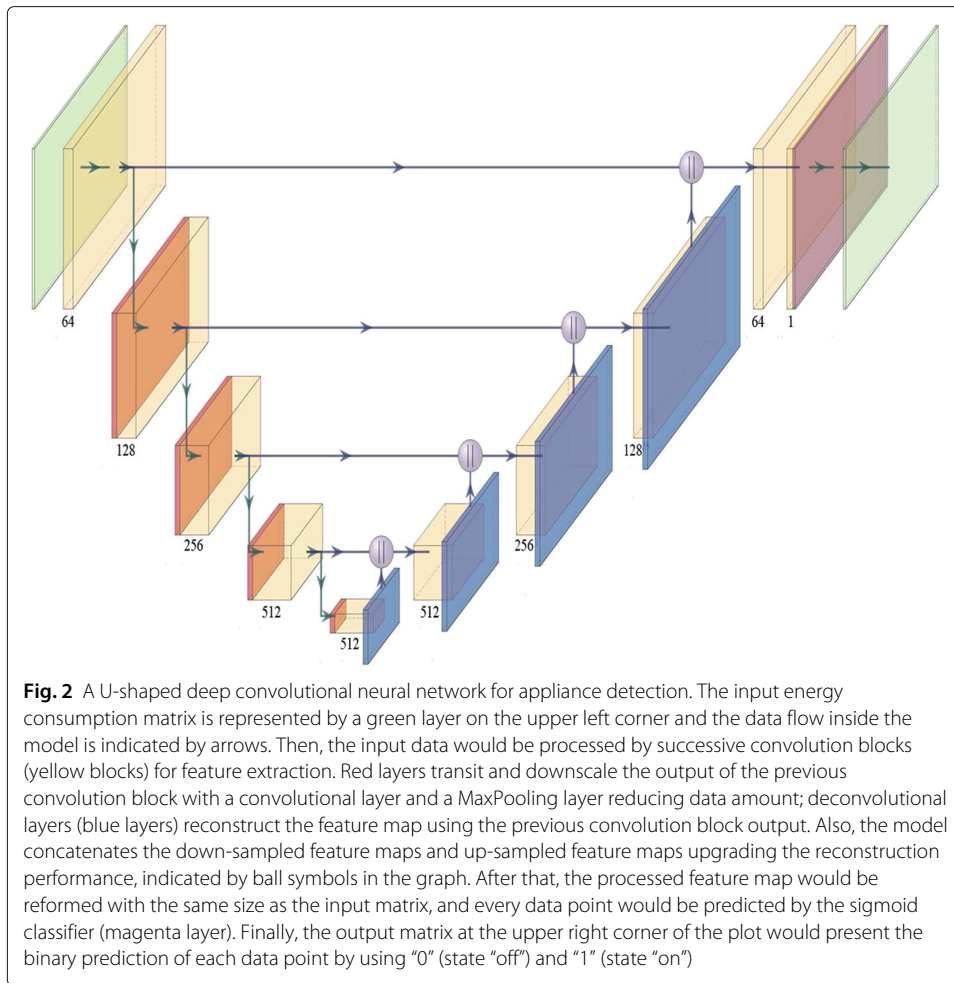
prediction ← *sigmoid*(*logits*);

The consecutive convolutional layers block is as same as architecture proposed in the U-net. This kind of block has a simpler structure and fewer parameters than other blocks. Based on the U-shaped model shown in Fig. 2, the model has 9 blocks, 18 convolutional layers in total for feature extraction and reform. However, the model may be simplified by reducing the number of layers for efficient convergence. Consequently, we introduce other three types of block structure with inside skip connections to keep high-level features without loss of low-level features.

The residual block structure could add the convolution results with original input of the block pixel-wise. Residual learning would lead to network structure simplification and avoid training hamper and degradation problem (He et al. 2016). A residual block can be represented as:

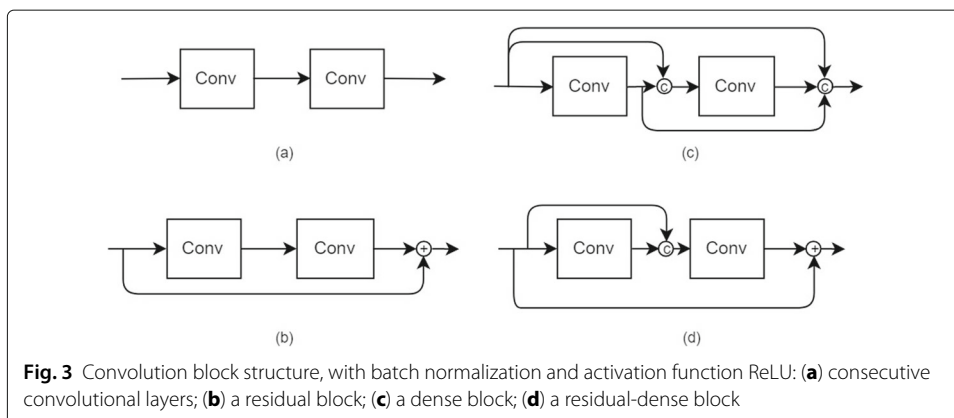$$y_i = \mathcal{H}(x_i) + \mathcal{F}(x_i) \tag{2}$$

where x_i is the input of the i-th residual block, $\mathcal{F}()$ is the residual function learned by consecutive convolutional layers, and $\mathcal{H}()$ is a mapping function. Here we choose $\mathcal{H}(x_i) = x_i$, the identity function. The pixel-wise addition calculation is labelled as a circle with symbol "+" in Fig. 3.

Based on the residual learning theory, densely connected convolutional layers block (Huang et al. 2017) has proposed with more complex structure leading to a trainable architecture. One layer inside the block would receive all information from its previous

Fig. 2 A U-shaped deep convolutional neural network for appliance detection. The input energy consumption matrix is represented by a green layer on the upper left corner and the data flow inside the model is indicated by arrows. Then, the input data would be processed by successive convolution blocks (yellow blocks) for feature extraction. Red layers transit and downscale the output of the previous convolution block with a convolutional layer and a MaxPooling layer reducing data amount; deconvolutional layers (blue layers) reconstruct the feature map using the previous convolution block output. Also, the model concatenates the down-sampled feature maps and up-sampled feature maps upgrading the reconstruction performance, indicated by ball symbols in the graph. After that, the processed feature map would be reformed with the same size as the input matrix, and every data point would be predicted by the sigmoid classifier (magenta layer). Finally, the output matrix at the upper right corner of the plot would present the binary prediction of each data point by using "0" (state "off") and "1" (state "on")

layers. Also, the output of the block consists of all levels of features from previous convolution computation. The output of the block would be passed directly to MaxPooling layers for downsizing and further inputs. The input of the dense block is:

$$x_i = \mathcal{H}([x_0, x_1, \ldots, x_{i-1}]) \tag{3}$$

Fig. 3 Convolution block structure, with batch normalization and activation function ReLU: (**a**) consecutive convolutional layers; (**b**) a residual block; (**c**) a dense block; (**d**) a residual-dense block

where $[x_0, x_1, \ldots, x_{i-1}]$ means the concatenation of all previous layers' input. The concatenation processing is labelled as a circle with character "c".

Due to the sophisticated output structure of (c) dense block, which has three times of layer channel comparing to the input matrix, computation of the whole U-shaped network would be too complex. Hence, we combine the structure (b) and (c) by replacing the concatenation processing with pixel-wise addition, and reform a residual block with densely connected convolutional layers. The output of a residual-dense block could be represented as:

$$x_1 = x_0 \tag{4}$$

$$x_2 = [\mathcal{F}_1(x_1), x_0] \tag{5}$$

$$y = \mathcal{H}(x_0) + \mathcal{F}_2(x_2) \tag{6}$$

where x_0, x_1, x_2 refers to the input of block, and two convolutional layers correspondingly; $\mathcal{F}_1$ and $\mathcal{F}_2$ represent convolution computation, and these two nested functions form the residual learning function; $[\mathcal{F}_1(x_1), x_0]$ means concatenation of variables; function $\mathcal{H}()$ is the identity mapping function here.

Besides convolutional blocks, max-pooling layers and deconvolutional layers are added in the contracting part and expanding part for feature compression and size restore. Skip connections also pass high-level features to restore feature maps. Different size of feature maps are cropped to the smaller size between them and be concatenated together for richer information for training. Finally, the last convolution block output (logits) would be processed by the sigmoid function for classification. The sigmoid is defined as:

$$\hat{y} = \sigma(logits + b) = \frac{1}{1 + e^{-(logits+b)}} \tag{7}$$

where y is the output of sigmoid function; *logits* is the variable input; b refers to bias.

For the implementation, we used TensorFlow 1.15 to implement our model. In the training stage, we split the 75% dataset as training data (2,988 households) and the remaining 25% as validation data (996 households). Each record was realigned as a $48 \times 365 \times 1$ matrix, and labels are transformed to matrices with value 0 and 1 in the same size. For the optimization, we used Adam optimizer. The training batch is 20. The learning rate is 0.0001 with a decay rate of 0.96 for 1000 training steps. Since we used a smaller kernel and channel size, the training time spends around 30-40 minutes, and the inference can be done within 1 second by a GTX 1080 Ti graphic card. A train step vs loss plot is displayed in Fig. 4.
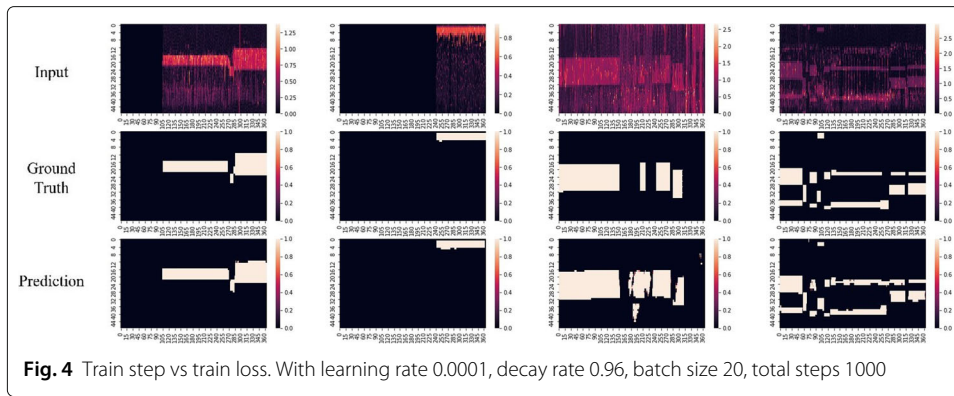
The loss function we utilize in the training is cross entropy with logits defined as:

$$CrossEntropyLoss = -\sum_{k=1}^{N} \left(y_k * \log \hat{y}_k \right) \tag{8}$$

where N refers to the number of classes in the prediction. In this case, we only have two classes, then N equals to 2.

Pool pump operation detection

In this step, the trained model would be finally applied to other data. The measurement of model performance is the mean intersection over union (mIOU). The mIOU indicates

Fig. 4 Train step vs train loss. With learning rate 0.0001, decay rate 0.96, batch size 20, total steps 1000

the proportion of overlap area prediction and the ground truth based on their union area. It can be defined as:

$$mIOU = \frac{1}{n} \sum_{i=1}^{n} \frac{|\hat{y} \bigcap y|}{|\hat{y} \bigcup y|} \tag{9}$$

The mIOU value locates between 0 and 1, and a larger value indicates a higher degree of coincidence of the predicted pattern and the ground truth pattern.

The final output matrix of pool pump operation on-off state could be analyzed further combined with energy consumption time pattern to investigate whether the pool pump can be shifted for demand-side management. Also, the load shape after shifting pool pump consumption could be estimated to decrease the peak-to-average ratio effectively.

Evaluation

In this section, we would focus on the prediction results and evaluation of our model. A comparison would be raised among baseline models (Neural NILM Rectangle model and Single Load Extraction Model) and our base PUMPNET. Baseline models are one-dimensional deep learning methods and are designed for a very small range of households. Comparing to baseline models, PUMPNET indeed has better performance in thousands of household context. And it is shows that our feature representation method is quite suitable for pool pump working patterns.

The second comparison is the convolution block backbone performance comparison. We implement 4 U-shaped networks with different blocks on the same training data and random seeds, but the performance of models varies slightly. We would compare the mIOU value of each model and point out which one is the best architecture for pool pump operation detection.

The dataset we used for evaluation consists of 3984 households' energy consumption in 2013. For each household, the energy consumption has been recorded as positive numbers referring to the kilowatts of power consumed by all appliances in the house by every day in a year with 30-minutes time intervals. Thus, each record includes 17,520 data points in time sequence. It has been split into 75% as training data (2,988 households) and 25% as validation data (996 households).

Comparing with baseline models

In this section, we compare our base PUMPNET (with consecutive convolutional layers) with baseline model. The baseline models that we selected is Neural NILM Rectangle (Kelly and Knottenbelt 2015a) and Single Load Extraction (Barsim and Yang 2018) . Details about baseline models have been discussed in 2.1 and the result is as follows:

As the baseline models in the original papers have adopted precision, recall, accuracy and F1 score measurements, we calculated these values for the base PUMPNET also for a complete comparison. These measurements are defined as:

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$Accuracy = \frac{TP + TN}{P + N} \tag{12}$$

$$F1 = 2 * \frac{precision * recall}{precision + recall} = \frac{2 * TP}{2 * TP + FN + FP} \tag{13}$$

, the P, N indicates counts of positive results and negative results correspondingly; the TP, FP, FN, TN mean counts of true positive, false positive, false negative and true negative predictions.

According to the comparison in Table 1, the base PUMPNET has the highest values in all measurements. The baseline 1D models both have precision values around 0.4. As the definition, precision refers to TP over all classified positive values. A lower precision indicates there is a large amount of false positive predictions.

High FP may result in the wrong classification depending on energy consumption values. 1D methods only can deal with close data points in the time dimension, which could result that a spike would be classified as positive status because of its high power consumption as real pool pump operation pattern. If the pool working period is very short, the feature would not clear after pooling layers. Then the high-level feature would be dismissed.

In conclusion, the performance of base PUMPNET proves the effectiveness of feature selection and transformation processing of matrix representation for pool pumps. Also, matrix representation combined with deep learning segmentation model would achieve better performance than 1D NILM models.

Comparing among convolution blocks

We train the U-shaped model with four convolution blocks and our energy consumption dataset. Four models all can converge with the given pool pump identification scenario. Finally, a comparison of models performance in mIOU measurement is shown in Table 2.

Table 1 Performance comparison between baseline models and PUMPNET

Model	mIOU	Precision	Recall	Accuracy	F1 score
Neural NILM rectangle	0.550	0.378	0.828	0.780	0.519
Single load extraction	0.568	0.403	0.738	0.806	0.521
PUMPNET	0.926	0.979	0.963	0.992	0.971

Table 2 mIOU of the U-shape models with different convolution block structure

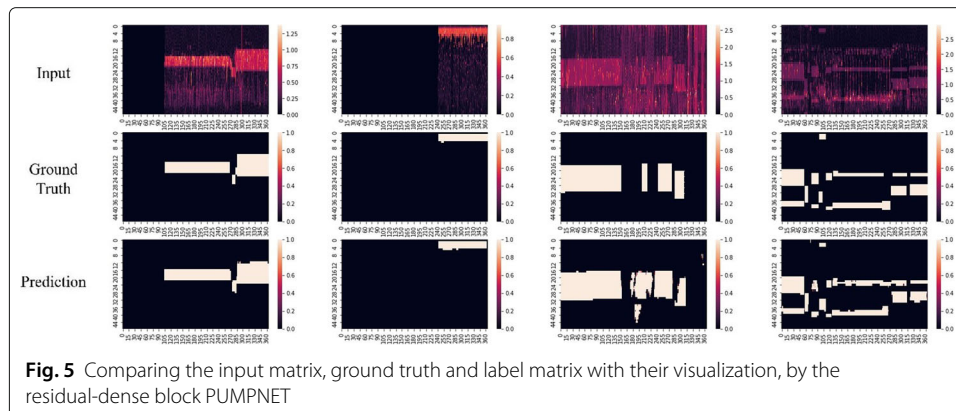| Convolution block backbone	mIOU
Consecutive convolutional layers	0.9259
Residual block	0.9412
Dense block	0.9390
Residual-dense block	0.9437

Models are all with mIOU value over 90%, and the U-shaped network with residual-dense blocks has the highest mIOU, 94.37%. This result indicates that the periodical appliance operation detection problem can be solved by image semantic segmentation methods. With a single end-to-end model, a group of universal parameters could be trained with well generalization ability. Hence, this method would lead to high improvement in load monitoring methodology for energy providers.

Case study

Sample of prediction results is compared in Fig. 5, with original input and corresponding ground truth. These results are all output by the model with residual-dense blocks. All images are visualizations of the numerical data matrix inputs, ground truth and predictions.

The model can directly extract rectangles which are clearly with higher power consumption value in the two results on the left side in Fig. 4. Areas we are interested in have an obvious boundary in the input matrix. However, as the prediction is pixel-wise, values on the state switch boundary are not exactly corresponding to the ground truth. This may be caused by that the output sigmoid classifier is relatively simple with one value input instead of the spatial information of neighbours'. Also, the start working time of a pool would not at the beginning of the 30-minute interval, resulting in the lower consumption amount comparing to normal on states. Thus, the aggregated data may not reflect the real working initialization, and a positive spot would be classified as negative. This is a type II error leading to false negative classification.

Similarly, the last two results indicate that pool pump working period classification are also influenced by the strong noise from other high power consumption devices, especially at the peak load time. This results in type I error, leading to false positive predictions. Although convolution has a strong ability in noise reduction, the input of our

Fig. 5 Comparing the input matrix, ground truth and label matrix with their visualization, by the residual-dense block PUMPNET

model contains less information with its single channel. Type I errors are caused by the high similarity between closed pixels. Such as the third results in Fig. 4, the bottom triangle is a type I error. It is far from the pattern of this household. It may result from parameters trained with other households. An improvement can be performed by using the Bayesian method, combining the prediction with a prior data distribution probability derived from training data.

Overall, the model could identify pool pumps with a high mIOU. According to the analysis on type I error and type II error, the model can be extended with sophisticated classifier adopting spatial information from neighbourhoods and be combined with a prior probability of pool pump operating time distribution to enlarge the prediction accuracy on boundaries.

Performance interpretation

The PUMPNET model introduces medical image classification architecture to energy analysis field. Surprisingly, the semantic segmentation mole could perform well in the "energy segmentation" task. Here, we try to deliver some interpretation of the model performance.

Shallow information Shallow information is high-resolution information directly passed from the encoder to the same height decoder via a concatenate operation. High-level information can provide more detailed features such as gradients for segmentation.

Deep information Deep information is low-resolution information after multiple down-sampling. The ability to provide contextual semantic information for the segmentation target throughout the image can be understood as a feature of the relationship between the response target and its environment. This feature helps the category judgment of objects (so classification problems usually only require low resolution/deep information, not involving multi-scale fusion).

Characteristics of the task Because boundaries in energy consumption matrices are blurred as medical images and more high-resolution information is needed for precise segmentation. As well, patterns of pool pump are relatively fixed as the internal structure of the human body in medical images. The distribution of the segmentation target is very regular, the semantics are simple and clear, and the low-resolution information can provide this information for the recognition of the target object. PUMPNET could combine low-resolution information (providing object-based recognition) and high-resolution information (providing accurate segmentation and positioning), leading to high performance in energy analysis task.

Conclusion

In this paper, we discussed the potential to use CNN for operation detection and proposed a pool pump operation detection network (PUMPNET) with a U-shaped convolutional neural network with residual-dense blocks. By converting time-series data into a date-time two-dimension matrix, we highlight pool pumps periodical operation features with low-frequency sampling data, instead of analyzing load activations using time-series

features only. The PUMPNET model could outperform time-series based deep learning NILM models, and achieve 94.37% mIOU in predictions. Overall, this model could address timer-based pool pump hidden patterns across thousands of households with a strong noise from other appliances. This power segmentation method offers a high-performance solution for centralized load monitoring instead of time-series based NILM methods. Also, PUMPNET would contribute to demand-side management and offer prior information for demand response programs.

References
Aboulian A, Green DH, Switzer JF, Kane TJ, Bredariol GV, Lindahl P, Donnal JS, Leeb SB (2018) Nilm dashboard: A power system monitor for electromechanical equipment diagnostics. IEEE Trans Ind Inform 15(3):1405–1414

Barsim KS, Yang B (2018) On the feasibility of generic deep disaggregation for single-load extraction. arXiv preprint arXiv:1802.02139

Batra N, Kelly J, Parson O, Dutta H, Knottenbelt W, Rogers A, Singh A, Srivastava M (2014) Nilmtk: an open source toolkit for non-intrusive load monitoring. In: Proceedings of the 5th International Conference on Future Energy Systems. ACM, Cambridge. pp 265–276

Burkhart S, Unterweger A, Eibl G, Engel D (2018) Detecting swimming pools in 15-minute load data. In: 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). IEEE, New York. pp 1651–1655

Chang H-H, Lian K-L, Su Y-C, Lee W-J (2013) Power-spectrum-based wavelet transform for nonintrusive demand monitoring and load identification. IEEE Trans Ind Appl 50(3):2081–2089

Chen L-C, Papandreou G, Schroff F, Adam H (2017) Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587

Faustine A, Mvungi NH, Kaijage S, Michael K (2017) A survey on non-intrusive load monitoring methodies and techniques for energy disaggregation problem. arXiv preprint arXiv:1703.00785

Gonzalez RC, Woods RE, Eddins SL (2018) Digital Image Processing. 4th edn. Pearson, New York

Gonçalves H, Ocneanu A, Bergés M, Fan R (2011) Unsupervised disaggregation of appliances using aggregated consumption data. In: The 1st KDD Workshop on Data Mining Applications in Sustainability (SustKDD), San Diego

Guillén-García E, Morales-Velazquez L, Zorita-Lamadrid AL, Duque-Perez O, Osornio-Rios RA, de Jesús Romero-Troncoso R (2019) Identification of the electrical load by c-means from non-intrusive monitoring of electrical signals in non-residential buildings. Int J Electr Power Energ Syst 104:21–28

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas. pp 770–778

Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu. pp 4700–4708

Jordehi AR (2019) Optimisation of demand response in electric power systems, a review. Renew Sust Energ Rev 103:308–319

---

[1]https://www.ausgrid.com.au/Industry/Our-Research/Data-to-share

Kelly J, Knottenbelt W (2015a) 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. ACM, New York. pp 55–64

Kelly J, Knottenbelt W (2015b) The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. Sci Data 2:150007. https://doi.org/10.1038/sdata.2015.7

Lin T-Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu. pp 2117–2125

Lin Y-H, Tsai M-S (2014) Non-intrusive load monitoring by novel neuro-fuzzy classification considering uncertainties. IEEE Trans Smart Grid 5(5):2376–2384

Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston. pp 3431–3440

Lopez JMG, Pouresmaeil E, Canizares CA, Bhattacharya K, Mosaddegh A, Solanki BV (2018) Smart residential load simulator for energy management in smart grids. IEEE Trans Ind Electron 66(2):1443–1452

Malik SA, Gondal TM, Ahmad S, Adil M, Qureshi R (2019) Towards optimization approaches in smart grid a review. In: 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET). IEEE. pp 1–5. https://doi.org/10.1109/icomet.2019.8673392

Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-assisted Intervention. Springer, Cham. pp 234–241

Saitoh T, Osaki T, Konishi R, Sugahara K (2010) Current sensor based home appliance and state of appliance recognition. SICE J Control Meas Syst Integr 3(2):86–93

Siano P (2014) Demand response and smart grids–A survey. Renew Sust Energ Rev 30:461–478

Ye F, Qian Y, Hu RQ (2015) A real-time information based demand-side management system in smart grid. IEEE Trans Parallel Distrib Syst 27(2):329–339

Yu F, Koltun V (2015) Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122

Zhang C, Zhong M, Wang Z, Goddard N, Sutton C (2018) Sequence-to-point learning with neural networks for non-intrusive load monitoring. In: Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans

Zhao H, Shi J, Qi X, Wang X, Jia J (2017) Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu. pp 2881–2890

Zhu Z, Tang J, Lambotharan S, Chin WH, Fan Z (2012) An integer linear programming based optimization for home demand-side management in smart grid. In: 2012 IEEE PES Innovative Smart Grid Technologies (ISGT). IEEE. pp 1–5. https://doi.org/10.1109/isgt.2012.6175785

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.