

MSDC: Exploiting Multi-State Power Consumption in Non-intrusive Load Monitoring Based on a Dual-CNN Model

Jialing He^{1,3}, Jiamou Liu², Zijian Zhang^{3,4*}, Yang Chen⁵, Yiwei Liu⁶, Bakh Khossainov⁷, Liehuang Zhu³

¹ College of Computer Science, Chongqing University, Chongqing, China, 400044

² School of Computer Science, The University of Auckland, Auckland 1142, New Zealand

³ School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China, 100081

⁴ Southeast Institute of Information Technology, Beijing Institute of Technology, Fujian China, 351100

⁵ Strong AI Lab, The University of Auckland, Auckland 1142, New Zealand

⁶ Defence Industry Secrecy Examination and Certification Center, Beijing, China, 100089

⁷ School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China, 611731

hejialing@cqu.edu.cn, {jiamou.liu, yang.chen}@auckland.ac.nz, {zhangzijian, liehuangz}@bit.edu.cn, yiweiliu_disecc@163.com, bmk@uestc.edu.cn

Abstract

Non-intrusive load monitoring (NILM) aims to decompose aggregated electrical usage signal into appliance-specific power consumption and it amounts to a classical example of blind source separation tasks. Leveraging recent progress on deep learning techniques, we design a new neural NILM model *Multi-State Dual CNN* (MSDC). Different from previous models, MSDC explicitly extracts information about the appliance's multiple states and state transitions, which in turn regulates the prediction of signals for appliances. More specifically, we employ a dual-CNN architecture: one CNN for outputting state distributions and the other for predicting the power of each state. A new technique is invented that utilizes conditional random fields (CRF) to capture state transitions. Experiments on two real-world datasets REDD and UK-DALE demonstrate that our model significantly outperforms state-of-the-art models while having good generalization capacity, achieving 6%-10% MAE gain and 33%-51% SAE gain to unseen appliances.

1 Introduction

Energy efficiency amounts to one of the major challenges facing today's families (Alahakoon and Yu 2016). Smart grids, through their ability to meter and monitor energy consumption to the level of individual households, provide fine-grained data to discover patterns in a user's behaviors. With these patterns, appropriate strategies can be implemented to improve power efficiency (Wilson, Hargreaves, and Hauxwell-Baldwin 2015). A critical issue of this technology is *non-intrusive load monitoring* (NILM): dissecting aggregated energy consumption signals of a household – the usual input to smart meters – into per-appliance signals (Hart 1992). NILM is important not only to accurately unravel the household's power consumption patterns (Çimen et al. 2020; Hassan et al. 2022), but also to tasks such as load forecasting

(Dinesh, Makonin, and Bajić 2019; Wang, Zhang, and Ren 2018; Kong et al. 2017) and malfunction detection (Shao et al. 2017; Green et al. 2019; Rashid et al. 2019).

NILM is a classical example of a single-channel *blind source separation* (BSS) task such as the infamous *cocktail party problem*, i.e., extracting multiple data sources from a single mixed observation. BSS is known to be highly underdetermined (Naik, Wang et al. 2014). To date, the mainstream solutions for BSS have relied on statistical modeling. In the NILM literature, hidden Markov models (HMM) (Zia, Bruckner, and Zaidi 2011) and conditional random fields (CRF) (Azaza and Wallin 2017) are two popular modeling paradigms to capture the operations of household appliances. As an appliance's power consumption signal is very much dictated by (1) the appliances' states which have (stable) power consumption levels, i.e., *power states*, and (2) the patterns of the power state transitions, these models embed appliances' power states and state transitions. This information is in turn used for statistical inference of power signals. A severe limitation of these mainstream solutions is computational complexity. For large households that contain many appliances with potentially many power states, these methods tend to incur unreasonable computational costs. Moreover, they perform poorly for unseen appliances as the inferred model only fits appliances used for training.

Recent advances in deep learning techniques have presented new pathways to address these limitations. More specifically, one can model the NILM problem as a *sequence-to-sequence* (seq2seq) learning task: Training a deep learning model that maps the aggregated power signal to the target per-appliance power signals. Based on this idea, several NILM solutions have been introduced utilizing various neural network architectures such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), long-short-term memory (LSTM), denoising autoencoders (Kelly and Knottenbelt 2015a; Mauch and Yang 2015; Hsu et al. 2019), and transformers (Yue et al. 2020).

In this paper, we argue that the existing neural-based mod-

*Corresponding author.

els are insufficient for NILM. In particular, these models were mostly designed to directly translate input aggregated signals into per-appliance signals, without the necessary step of exploiting power states and state transitions of the appliances. As shown in classical statistical inference models, power state and transitions of appliances can provide key knowledge on the power consumption of appliances and could greatly strengthen the predictive ability of a model. Further, uncovering these power states and patterns of state shifts may endow the prediction results with better interpretability. Our goal is thus to develop a neural-based NILM model through mining power states and state transitions.

Our main contributions are three fold: (1) We define a formal model of the *multi-state NILM problem* and provide a theoretical justification (see Theorem 1, Sec. 3) for the advantage of the multi-state setting on variance reduction in sampling power data. (2) Accordingly, we propose a novel multi-state NILM model (called MSDC, see Sec. 4) that features a dual-CNN architecture: one CNN (*state-CNN*) for capturing the appliance’s multiple power states, the other (*value-CNN*) for predicting the power consumption for each state. A cross-entropy is added to *state-CNN* as a regularization item, which degrades the error in predicting the appliance’s power consumption. We further replace the cross-entropy loss with a CRF (Lafferty, McCallum, and Pereira 2001) regularization that enables capturing state transitions. (3) Experimental results on two real-world datasets REDD (Kolter and Johnson 2011) and UK-DALE (Kelly and Knottenbelt 2015b) (see Sec. 5) show that our model generalizes well to unseen appliances where it achieves 6%-10% MAE gain and 33%-51% SAE gain over the state-of-art on the accuracy of energy decomposing. Moreover, visualization results demonstrate that the power state transitions predicted by our model align with the ground truth at a higher level, compared with baselines.

2 Related Work

NILM was first proposed in (Hart 1992). Mainstream solutions of the problem employ statistical inference models such as HMM and CRF. (Zia, Bruckner, and Zaidi 2011) modelled an aggregated signal as a combination of HMMs, each corresponding to an appliance, and found that power consumption patterns of appliances can be differentiated from the aggregated profile. (Kim et al. 2011) investigated several variants of HMM and demonstrated that a conditional factorial HMM integrating additional features about the time usage of appliances outperforms other HMM models. (Kolter and Jaakkola 2012) incorporated additive factorial HMM into the NILM task and developed a convex formulation of approximate inference that is free of local optimal. Subsequent HMM variants further demonstrated non-trivial performance gain (Kolter and Jaakkola 2012; Kong et al. 2016; Mauch and Yang 2016a). Nevertheless, the assumption that any observation is independent of the other may violate real-life situations and lead to label bias issue.

CRF-based methods relax the independent assumption and utilize the contextual information from all observations to mitigate the label bias problem. (Azaza and Wallin 2017) exploited CRF and the clustering algorithm to capture the

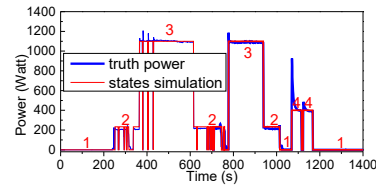


Figure 1: Multi-state power consumption of a dishwasher.

component appliance’s power signal and corresponding on-off states. This method was extended by (He et al. 2019a) to capture appliances’ multiple states. However, high computational complexity and poor scalability prevent these statistical modeling methods from practical usage. Our proposed model avoids these issues because it does not explicitly maintain a state transition network, but rather capture transitions implicitly using CNNs.

Neural networks are suitable for NILM thanks to their high expressibility. (Kelly and Knottenbelt 2015a) first tackled NILM using deep neural networks such as CNN, LSTM, and denoising autoencoders, with superior performance over statistical modeling methods. A series of subsequent CNN-based models (Zhang et al. 2018; Shin et al. 2019; Chen et al. 2019) revealed that CNN can provide a versatile framework for extracting latent features such as power thresholds, change point and duration for the power consumption of appliances. These features are useful for identifying appliances, thereby outperforming models based on other neural networks. By adding more convolutional layers, enlarging the number of filters and the kernel size, (Zhang et al. 2018) proposed a CNN model that gains considerable improvements. As CNN-based models are shown to exhibit consistently reliable performance, we also adopt CNN in this work.

Other models leveraged the correlations between whether an appliance is on or off and its power (Shin et al. 2019). A sub-network is added to capture appliances’ on-off states, which expedite identifying power signals. Several models have been proposed using dilated convolutions (Chen et al. 2019), generative adversarial network (Pan et al. 2020), and attention mechanisms (Sudoso and Piccialli 2019). However, these models are limited to predicting whether the appliance is on or off, without making an effort to reveal the full spectrum of power states. In some sense, one can view capturing the on-off states of an appliance as an extremely coarse approximation of the power states. Yet, doing this means that all different power levels when the appliance is turned on are regarded as being in one “on” state, hence does not provide enough information to deduce the power signal.

3 Problem Formulation

Multi-state non-intrusive load monitoring (MS-NILM) problem seeks to recover the power consumption of individual multi-state appliances from the aggregated power signal. Let $\mathbf{X} = (x_1, x_2, \dots, x_T) \in \mathbb{R}_+^T$ denote the *aggregated power*, where $T \in \mathbb{N}_+$ represents the *measured time*. Suppose the scope of investigation includes N appliances. Appliance i ’s power consumption is denoted by

$\mathbf{Y}^i = (y_1^i, y_2^i, \dots, y_T^i) \in \mathbb{R}_+^T$. Set $[N]$ as $\{1, 2, \dots, N\}$. Following standard assumption in NILM literature, we assume $\forall i \in [N], t \in [T]$, the signal $x_t = \sum_{i=1}^N y_t^i + z_t + \epsilon_t$, where z_t denotes the total power consumption of appliances included in the power reading but fall *outside* of our scope of investigation, and ϵ_t is a noise signal (Zhang et al. 2018).

A multi-state appliance can be modeled as a *finite state machine* with a fixed set of *operational states*. As we are interested only in power consumption, we view two *operational states* as in the same *power state* if they have the same power consumption level, and transitions between operational states are grouped into transitions between power states. For appliance i , let $M^i \in \mathbb{N}_+$ denote the number of its power states. In principle, each power state is characterized by a unique power level, and at each time step the appliance will be in one power state s_t^i . In reality, this assumption may be over-simplified: (1) The actual power reading for the appliance at a state is prone to perturbation so there could be small fluctuation at the fixed power level. Therefore it makes sense to individually express the power consumption level $c_{t,j}^i$ for appliance i at state j and at given time t . (2) Due to uncertainty within the learning process, it is difficult to pinpoint a single state for the appliance to be operating in at a given time, but rather a probability $p_{t,j}^i$ for appliance i to be in state j at time t . More formally, the power consumption signal for states at time step t is determined by power consumption at all M^i states $\mathcal{C}_t^i = (c_{t,1}^i, c_{t,2}^i, \dots, c_{t,M^i}^i) \in \mathbb{R}_+^{M^i}$ and probability distribution $\mathcal{P}_t^i = (p_{t,1}^i, p_{t,2}^i, \dots, p_{t,M^i}^i) \in [0, 1]^{M^i}$, where $c_{t,j}^i \neq c_{t,k}^i$ for $j \neq k \in [M^i]$ and $p_{t,s}^i$ is the probability $\Pr(s_t^i = s)$. Notably, the power state s_t^i generally depends on previous states s_1^i, \dots, s_{t-1}^i . Indeed, the state transitions would follow certain specific patterns. E.g., in Fig. 1 State 3 is a successor of State 2. Following standard convention, we assume that the state transition is Markovian, i.e., $\Pr(s_t^i | s_{t-1}^i, \dots, s_1^i) = \Pr(s_t^i | s_{t-1}^i)$. Given above settings, an appliance's power consumption can be viewed as the expectation on that of each state:

$$y_t^i = \mathcal{C}_t^i \cdot \mathcal{P}_t^i, \quad \forall i \in [N], t \in [T]. \quad (1)$$

MS-NILM can thus be addressed by inferring each appliance's power state sequence and the corresponding power consumption from the aggregated power signals; at each time step, each appliance's power consumption is subsequently estimated according to Eq. (1). To summarize, we state the problem of MS-NILM as follows:

Problem Statement: *Given a set of instances of energy disaggregation of the form $\{(\mathbf{X}, \mathbf{Y}^1, \dots, \mathbf{Y}^N)\}$, infer \mathcal{C}_t^i and \mathcal{P}_t^i for all $i \in [N]$ and $t \in [T]$, and further generalize to unseen aggregated power signal \mathbf{X}' .*

3.1 Justifications for the Multi-state Setting

We next discuss the need for a multi-state setting for NILM. The idea is to show that sampling power data from a fine-grained state structure can result in a smaller sample variance, making the average power of each appliance easier to predict. We next make some useful assumptions, based

on which we present our main theoretical result. Following conventional practice in NILM literature, we assume that an appliance's power consumption follows a normal distribution (Zhang et al. 2018; Mauch and Yang 2016b).

Assumption 1. *For each appliance i , the power consumption of each state s at each time step t is drawn from a normal distribution, i.e., $c_{t,s}^i \sim \mathcal{N}(\mu_s^i, \sigma_s^i)$.*

Recall that the observed power is the expectation on the power of all states, i.e., $y_t^i = \sum_{s=1}^{M^i} p_{t,s}^i c_{t,s}^i$. Due to the independence of power measures of all states and the additivity of normal distributions, we have the following fact.

Fact 1. *Under Assumption 1, the input power y_t^i of an appliance also follows a normal distribution $\mathcal{N}(\mu^i, \sigma^i)$ such that $\mu^i = \sum_{s=1}^{M^i} p_{t,s}^i \mu_s^i$ and $(\sigma^i)^2 = \sum_{s=1}^{M^i} (p_{t,s}^i \sigma_s^i)^2$.*

Since a multi-state model decomposes the total power into a fine-grained state structure, it is natural to propose the following assumption that enforces the variance of the power of each state to not exceed that of the observed total power. This can be seen from the truth depicted in Fig. 1: a dishwasher has a steady power level at each state; while, the variance would be increased if we merge any two states into an abstract state.

Assumption 2. *For all $s \in [M^i]$, $\sigma_s^i \leq \sigma^i$.*

We are now ready to present our main result that uses the following notations: \tilde{y}_t^i denotes the sampled power under the single-state assumption; $\bar{y}_t^i = \mathbb{E}_{s \in [M^i]} [c_{t,s}^i]$ is the sampled power under the multi-state setting which is obtain by averaging the sampled power $c_{t,s}^i$ of each state $s \in [M^i]$ (suppose the state information is known a priori). We show in the next theorem that the sampled power data can enjoy a reduced sample variance from the multi-state setting.

Theorem 1. *Suppose we have a sufficiently large number of independent samples. Under Assumptions 1 and 2, the expectations and variances of \tilde{y}_t^i and \bar{y}_t^i satisfy $\mathbb{E}[\tilde{y}_t^i] = \mathbb{E}[\bar{y}_t^i]$ and $\mathbb{D}[\tilde{y}_t^i] \leq \mathbb{D}[\bar{y}_t^i]$ for all $t \in [T]$, where the inequality is strict if $M^i \geq 2$.*

Proof. With a sufficiently large number of independent samples, both \tilde{y}_t^i and \bar{y}_t^i would approach normal distributions, i.e., $\tilde{y}_t^i \sim \mathcal{N}(\mu^i, \sigma^i)$ and $\bar{y}_t^i = \sum_{s=1}^{M^i} p_{t,s}^i c_{t,s}^i$. Then, according to Fact. 1, $\mathbb{E}[\tilde{y}_t^i] = \mathbb{E}[\bar{y}_t^i]$ follows immediately from the additivity of normal distributions. The reduced variance can be derived from Assumption 2 as follows:

$$\begin{aligned} \mathbb{D}[\tilde{y}_t^i] &= \mathbb{D}\left[\sum_{s=1}^{M^i} p_{t,s}^i c_{t,s}^i\right] = \sum_{s=1}^{M^i} p_{t,s}^2 \mathbb{D}[c_{t,s}^i] \\ &= \sum_{s=1}^{M^i} p_{t,s}^2 \sigma_s^i \leq \sum_{s=1}^{M^i} p_{t,s}^2 \sigma^i \leq \left(\sum_{s=1}^{M^i} p_{t,s}\right)^2 \sigma^i \\ &= \sigma^i = \mathbb{D}[\bar{y}_t^i]. \end{aligned}$$

If $M^i \geq 2$, we have $\sum_{s=1}^{M^i} p_{t,s}^2 < \left(\sum_{s=1}^{M^i} p_{t,s}\right)^2$ and hence the inequalities above is strict. \square

Corollary 1. *When the appliance has $M^i (M^i \geq 2)$ power states in total, then the power estimation regarding the ap-*

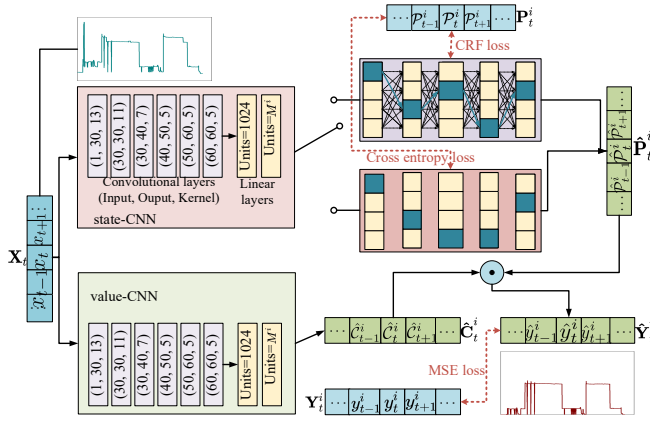


Figure 2: Our model features a dual-CNN architecture. Two versions, MSDC and MSDC-CRF, can be obtained by switching between the cross-entropy loss and the CRF loss.

pliance in a multi-state setting with M^i states would approach the mean value of truth power data with a higher probability than using the single-state setting.

Proof. Assume $\tilde{y}_t^i = \sum_{s=1}^{M^i} p_{t,s}^i c_{t,s}^i$ and \tilde{y}_t^i represent the power estimation utilizing M^i states and one state respectively at each time step. According to Theorem 1, \tilde{y}_t^i and \tilde{y}_t^i satisfy normal distribution, which denote as $\tilde{y}_t^i \sim \mathcal{N}(\tilde{\mu}^i, \tilde{\sigma}^i)$ and $\tilde{y}_t^i \sim \mathcal{N}(\tilde{\mu}^i, \tilde{\sigma}^i)$ respectively, then we have $\tilde{\mu}^i = \tilde{\mu}^i$, $\tilde{\sigma}^i < \tilde{\sigma}^i$. Furthermore for $\forall \xi > 0$:

$$\Pr(|\tilde{y}_t^i - \tilde{\mu}^i| < \xi) = \Pr\left(\left|\frac{\tilde{y}_t^i - \tilde{\mu}^i}{\tilde{\sigma}^i}\right| < \frac{\xi}{\tilde{\sigma}^i}\right) = 2\Phi\left(\frac{\xi}{\tilde{\sigma}^i}\right) - 1$$

Similarly, we have: $\Pr(|\tilde{y}_t^i - \tilde{\mu}^i| < \xi) = 2\Phi\left(\frac{\xi}{\tilde{\sigma}^i}\right) - 1$, where $\Phi(\cdot)$ represents the probability meets the standard norm distribution. Since $\tilde{\sigma}^i < \tilde{\sigma}^i$, then $\Phi\left(\frac{\xi}{\tilde{\sigma}^i}\right) > \Phi\left(\frac{\xi}{\tilde{\sigma}^i}\right)$, further we can get that $\Pr(|\tilde{y}_t^i - \tilde{\mu}^i| < \xi) > \Pr(|\tilde{y}_t^i - \tilde{\mu}^i| < \xi)$. \square

Remark 1. The corollary ensures the power estimation of our scheme using multi-state setting can attain a smaller MAE (see section 5. Perform metrics) with a higher probability on average than single state-based schemes.

4 The Multi-State Dual CNN Model

The discussion above justifies the benefit of the multi-state setting in MS-NILM over the single-state setting. In this section, we propose a novel model for MS-NILM called *Multi-State Dual CNN* (MSDC). We begin with an overview of the architecture, followed by the elaboration of its mechanisms. Code and data used for MSDC can be found from our link¹.

4.1 Overview

As Fig. 2 shows, we train a model for each appliance $i \in [N]$. The dataset consists of the aggregated power signal \mathbf{X} as input and \mathbf{Y}^i of appliance $i \in [N]$ as output. From \mathbf{Y}^i , an exogenous procedure pre-extracts a fixed set of M^i power

states as well as the sequence of power states $s_1^i, s_2^i, \dots, s_T^i$ of appliance i . Obtaining this information is possible as (1) the power consumption of common household appliances exhibit clear power-state patterns as shown in Appendix A: a power state appears as a consecutive sequence of relatively stable time steps and a transition takes place when a drastic shift occurs at a time step followed by another power state, whose signal is visibly separated from the previous state; and (2) the states can be conveniently extracted using statistical methods, e.g., a clustering algorithm that groups time steps together if they have similar power levels.

To address the MS-NILM problem, our model produces the probability distribution \mathcal{P}_t^i over the (pre-extracted) set of power states and the power consumption vector \mathbf{C}_t^i as its output. This is done using a *dual-CNN* architecture consisting of a *state-CNN* (for \mathcal{P}_t^i) and a *value-CNN* (for \mathbf{C}_t^i). Training the model requires a two-part loss function: First a prediction loss is defined over the disparity between the predicted power consumption signal $\mathcal{P}_t^i \cdot \mathbf{C}_t^i$ and label \mathbf{Y}^i , and then a state-based loss measuring inconsistency between the predicted power state sequence and the pre-extracted sequence.

A simple way to define the state-based loss is simply compare the probability distribution \mathcal{P}_t^i against the pre-extracted power state q_t^i . This, however, omits any pattern of state transitions of the appliance. As each appliance is seen as a finite state machine, the shifts between consecutive states in the power state sequence must follow certain patterns that correlate power states. To capture these patterns, we proposed a novel technique that employs *conditional random fields* (CRF) (Lafferty, McCallum, and Pereira 2001), a discriminative model for exploiting the correlations between consecutive elements of sequential data. Specifically, we use CRF as an regularization for the state-CNN to explicitly compute the contextual correlations between states. The model is illustrated in Fig. 2. Details of both pre-extracting power states and CRF follow shortly in this section.

4.2 Model Description

Pre-extracting states. The procedure that pre-extracts power state sequence from \mathbf{Y}^i naturally plays a significant role in determining the performance of the model, yet evidence has shown that this step can be easily accomplished for common household appliances whose power states are clearly distinguishable. For example, the *mean shift clustering* algorithm (Fukunaga and Hostetler 1975; Cheng 1995) that features assigning clusters to data without manually pre-defining the number of clusters, could be a useful tool. The cluster centers are determined in an iterative process through computing the mean of the samples in a certain region. It allows us to extract state labels for each appliance by inputting the appliance's power readings: $\mathbf{S}^i = \text{MeanShift}(\mathbf{Y}^i)$, where $s_t^i \in [M^i]$ and M^i is automatically determined according to the appliance's power readings (\mathbf{Y}^i).

Model structure. Aiming at the original power sequence, previous studies employ the sliding window method to overcome the long-sequence issue: separating the original data into a series of short-sequences/windows of equal length. We use a variant of the *sliding window* method to overcome the long-sequence issue, which predicts a subsequence

¹<https://github.com/sub-paper/MSDC-NILM>

centered at the midpoint of the input window. An input window is obtained by dividing the aggregated power signal into several w -length segments (can be overlapping). The two CNNs share the same input window, denoted by $\mathbf{X}_{t,w} = (x_{t-\lfloor \frac{w}{2} \rfloor}, \dots, x_{t+\lfloor \frac{w}{2} \rfloor-1})$. We use f_{state}^i and f_{power}^i to represent two CNNs, respectively. The output of our model is a shorter window centered at t , which represents a sequence of predicted power signals of an individual appliance. We formally write the output window as $\hat{\mathbf{Y}}_{t,q}^i = (\hat{y}_{t-\lfloor \frac{q}{2} \rfloor}^i, \dots, \hat{y}_{t+\lfloor \frac{q}{2} \rfloor-1}^i)$, where $q < w$.

We denote the *state-CNN* by $f_{\text{state}}^i: \mathbb{R}_+^w \rightarrow [0, 1]^{M^i \times q}$, i.e., it outputs a window of predicted state distributions:

$$\hat{\mathbf{P}}_{t,q}^i = f_{\text{state}}^i(\mathbf{X}_{t,w}), \quad (2)$$

where $\hat{\mathbf{P}}_{t,q}^i = (\hat{p}_{t-\lfloor \frac{q}{2} \rfloor}^i, \dots, \hat{p}_{t+\lfloor \frac{q}{2} \rfloor-1}^i)$ and $\hat{p}_{\tau}^i = (\hat{p}_{\tau,1}, \dots, \hat{p}_{\tau,M^i})$. Analogously, we represent the *value-CNN* by $f_{\text{power}}^i: \mathbb{R}_+^w \rightarrow \mathbb{R}_+^{M^i \times q}$ that predicts power consumption of each state at each time step:

$$\hat{\mathbf{C}}_{t,q}^i = f_{\text{power}}^i(\mathbf{X}_{t,w}), \quad (3)$$

where $\hat{\mathbf{C}}_{t,q}^i = (\hat{c}_{t-\lfloor \frac{q}{2} \rfloor}^i, \dots, \hat{c}_{t+\lfloor \frac{q}{2} \rfloor-1}^i)$, and $\hat{c}_{\tau}^i = (\hat{c}_{\tau,1}, \dots, \hat{c}_{\tau,M^i})$. To simplify the exposition, we will omit w and s in some notations. By Eq. (1), $\hat{\mathbf{Y}}_{t,q}^i$ is derived from the element-wise product of the outputs of two CNNs. More formally, we denote the combination of two CNNs by $f_{\text{comb}}^i: \mathbb{R}_+^w \rightarrow \mathbb{R}_+^q$ such that

$$\hat{\mathbf{Y}}_{t,q}^i = f_{\text{comb}}^i(\mathbf{X}_{t,w}) = f_{\text{state}}^i(\mathbf{X}_{t,w}) \odot f_{\text{power}}^i(\mathbf{X}_{t,w}). \quad (4)$$

Finally, the state can subsequently be determined as the one with the maximum probability: $\hat{\mathbf{S}}_{t,q}^i = \arg \max(\hat{\mathbf{P}}_{t,q}^i)$.

Loss functions. The cost function of the model consists of two parts. The first part considers the error in power consumption prediction, which we measure by the mean squared error averaged over all output windows:

$$J_{\text{power}}^i = \mathbb{E}_{t \in [T]} \left[\left(y_t^i - \hat{p}_t^i \hat{c}_t^i \right)^2 \right], \quad (5)$$

where y_t^i represents the true power signal of the i th appliance, and \hat{p}_t^i, \hat{c}_t^i are outputs of two CNNs as defined above. The second part penalizes the difference between predicted and true states. We propose two options for realizing it. The first option is to simply ignore the dependence in state shifts and thereby we can use the averaged cross-entropy to measure the difference between the predicted and true states:

$$J_{\text{state}}^i = -\mathbb{E}_{t \in [T]} \left[\sum_{s=1}^{M^i} p_{t,s}^i \log \hat{p}_{t,s}^i \right], \quad (6)$$

where the true state distribution $\mathcal{P}_t^i = (p_{t,1}^i, \dots, p_{t,M^i}^i)$ is estimated from pre-extracted states. The final cost function sums over two parts: $J_{\text{MSDC}}^i = J_{\text{state}}^i + J_{\text{power}}^i$. In the paper, we use the name **MSDC** to represent the model with the loss function defined above. We next give the second option to realize the loss in terms of state prediction.

4.3 The CRF Regularization

In order to capture the state transition, we replace the cross entropy loss J_{state}^i in Eq. (6) with the CRF regularization that explicitly computes correlations between states. Intuitively, Eq. (6) implies that the output of the *state-CNN* is a sequence of *independent* state distributions. The state at each time step can subsequently be determined as the one with the maximum probability. Training the state-CNN can thus be viewed as solving q independent M^i -classification problems. While, the state distributions in the output window are not independent in the face of state transitions. The CRF regularization allows us to capture state transition by train the *state-CNN* in a state sequence-centered way, i.e., view a state sequence as the minimum element in calculating the difference between the predicted and true states, rather than independently consider the state for each time step. Since there are $(M^i)^q$ possible state sequences, training the state-CNN thus turns to solving one $(M^i)^q$ -classification problem. The CRF regularization amounts to a maximum likelihood estimation (MLE) in terms of state sequences. The MLE objective is a log-likelihood consisting of two parts: one is called the *emission score* that captures the likelihood of each true state; the other is the *transition score* that captures the transition probability between each two neighboring states. It can be formally written as follows:

$$J_{\text{CRF}}^i = -\mathbb{E} \left[\sum_{t=1}^T \hat{p}_{t,s_t^i}^i + \sum_{t=1}^{T-1} \Psi(s_t^i, s_{t+1}^i) \right] + \log Z,$$

where the expectation is taken on all input state sequences. The sum $\sum_{t=1}^T \hat{p}_{t,s_t^i}^i$ is namely the emission score, $\Psi(s_t^i, s_{t+1}^i)$ is the transition score computed from all predicted state distributions (see Appendix B for details), and Z denotes the partition function for two scores, i.e., the total score for all input state sequences. We adopt **MSDC-CRF** for the model with CRF regularization, whose loss function is namely $J_{\text{MSDC-CRF}}^i = J_{\text{CRF}}^i + J_{\text{power}}^i$.

5 Experiments

We evaluate our model on the two most commonly used NILM benchmark datasets, REDD and UK-DALE, which are collected from appliances in different households in the US and UK, respectively. We seek to answer: (1) (*generalization capability*) Using a set of training data sampled from one household, can our model generalize to other (unseen) households? (2) (*predictive ability*) Can our model accurately predict the power consumption of individual appliances in the household where the training data is sampled?

5.1 Setup

Datasets. REDD consists of power readings of 6 households, where the aggregated power and individual appliances' power are recorded every 1 and 3 seconds, resp. We choose four appliances (microwave, washing machine, dishwasher, and fridge) in houses 1, 2, and 3 for evaluation. UK-DALE accommodates power readings of 6 UK households. Both aggregated and per-appliance power readings are measured every 6 seconds. We select 5 appliances (kettle, microwave, washing machine, dishwasher, and fridge) in

| Scheme | Fridge | Dishwasher | Microwave | Washing machine | Average | Improvement(%) |
|-----------|---------------------------|-----------------------------------|-----------------------------------|----------------------------------|--|--|
| FHMM | 95.96/29.03/49.17 | 180.35/713.80/168.34 | 41.56/79.92/27.15 | 219.22/498.89/213.46 | 134.27/330.41/114.53 | -/- |
| LSTM | 39.59/17.72/22.57 | 25.43/54.89/24.41 | 30.99/19.48/24.43 | 13.23/4.52/11.03 | 26.31/24.15/20.61 | -/- |
| S2P | 37.66/15.47/17.66 | 19.86/15.81/15.24 | 27.19/21.37/19.33 | 13.27/4.58/10.84 | 24.50/14.31/15.78 | -/- |
| BERT4NILM | 28.98 /31.49/17.66 | 23.31/88.76/23.05 | 17.53 /76.80/15.49 | 16.68/8.75/12.03 | 21.63/51.90/17.06 | -/- |
| SGN | 33.61/19.23/18.45 | 17.85/38.01/10.33 | 22.13/49.84/16.35 | 12.97 /2.30/ 10.05 | 21.64/27.35/13.80 | 0.00/0.00/0.00 |
| MSDC | 31.78/7.78/ 15.59 | 13.03/ 18.44 / 6.57 | 20.36/26.74/13.51 | 13.63/4.34/11.79 | 19.70/14.33/11.87 | 8.96/47.61/13.99 |
| MSDC-CRF | 30.64/ 5.15 /16.08 | 12.81 /21.87/6.75 | 20.35 /23.81/ 12.94 | 13.47/ 2.28 /10.65 | 19.32 / 13.28 / 11.61 | 10.72 / 51.44 / 15.87 |

| Scheme | Kettle | Fridge | Dishwasher | Microwave | Washing machine | Average | Improvement(%) |
|-----------|--|--|-----------------------------------|----------------------------------|---------------------------------|---|---|
| S2P | 18.95/26.78/9.25 | 28.04/31.79/13.71 | 26.01/27.82/16.41 | 12.35/14.23/6.25 | 10.85/ 4.35 /9.54 | 19.24/20.99/11.03 | -/- |
| BERT4NILM | 5.02 / 10.94 / 3.46 | 31.22/35.04/19.91 | 34.25/81.11/33.53 | 6.57 /99.80/6.58 | 10.31/41.36/9.45 | 17.47/53.65/14.58 | -/- |
| SGN | 11.81/25.28/10.38 | 21.19/14.28/9.07 | 15.05 /17.47/ 11.12 | 7.06 /80.31/6.77 | 13.27/26.79/11.85 | 13.68/32.83/9.94 | 0.00/0.00/0.00 |
| MSDC | 11.14/17.88/7.63 | 16.34 / 6.49 / 4.42 | 19.21/ 15.75 /13.60 | 9.92/ 45.71 / 6.10 | 7.46 /22.50/ 4.93 | 12.81 / 21.67 / 7.34 | 6.36 / 33.99 / 22.13 |

Table 1: Results for MAE /SAE /SAE_δ in REDD(top) and UK-DALE(bottom). The bold numbers indicate the best results.

house 1 and 2 for use. We choose these five types of appliances for three reasons: (1) Generalization capability of our model is the most critical factor we aim to evaluate. For this, the training and testing data should be taken from different houses. That is the same type of appliances must be installed in more than one house. We thus eliminate appliances that are only installed in a single house. We further exclude appliances whose power data are almost all zero values caused by equipment failure and anomalies. (2) As stated in (Kelly and Knottenbelt 2015a; Zhang et al. 2018), these five types of appliances consume a significant portion of household energy. Furthermore, they represent a wide range of possible ‘power spectra’ from the simple on/off-states of a kettle to the complex multiple states of a dishwasher. (3) The datasets and the appliances are chosen by all previous baselines, providing the grounding for fair comparisons.

Baselines. We compare our model against five baselines: (1) **FHMM** (Kim et al. 2011). The (classical) HMM-based model in which each state corresponds to an abstract state that absorbs several states of an appliance. (2) **LSTM** (Kelly and Knottenbelt 2015a). The first RNN architecture for NILM, which utilizes the long short-term memory (LSTM) network. (3) **S2P** (Zhang et al. 2018). A single-state CNN-based model that uses the sequence-to-point strategy, i.e., output only the middle point instead of the entire window. (4) **BERT4NILM** (Yue et al. 2020). A bidirectional transformer model with 2 transformer layers and 2 attention heads within each layer. (5) **SGN** (Shin et al. 2019). A state-of-the-art model with a dual-CNN architecture that can capture the appliance’s on-off states.

Performance metrics. We adopt the following three metrics as performance indicators. (1) *Mean Absolute Error* (MAE). MAE(Watt) is a general metric to measure the estimation error at each time point. It is formally computed by: $MAE^i = \frac{1}{T} \sum_{t=1}^T |\hat{y}^i - y^i|$. (2) *Normalized Signal Aggregate Error* (SAE). SAE(%) measures the total estimation error in total test time: $SAE^i = |\hat{r}^i - r^i|/r^i$, where $\hat{r}^i = \sum_{i=1}^T \hat{y}^i$ and $r^i = \sum_{i=1}^T y^i$. This metric is also adopted in S2S (Zhang et al. 2018). (3) *A Variant of SAE* (SAE_δ). SAE_δ(Watt) is a variant of SAE that measures the average total error in a sub-

period of the total time: $SAE_{\delta}^i = (\sum_{k=1}^{T_{\delta}} \frac{1}{N_{\delta}} |\hat{r}_k^i - r_k^i|)/T_{\delta}$. Here, δ represents a physical time period. The total measured time are split to T_{δ} time periods, each of an equal length of N_{δ} time steps. Following (Shin et al. 2019), we set $\delta = 1$ hour and $N_{\delta} = 1200$. The average predicted value and true value in the k th period are captured by $\hat{r}_k^i = \sum_{t=1}^{N_{\delta}} \hat{y}_{k+t}^i$ and $r_k^i = \sum_{t=1}^{N_{\delta}} y_{k+t}^i$, respectively.

Parameter settings. We train one model per appliance. Each of two CNNs consists of 6 convolutional layers plus 2 fully connected layers. The CNNs are implemented by Python and Pytorch 1.4.0+cuda 10+cudnn7, and trained on machines with GTX 1070 Ti (8G) + Ryzen 7 1700 (16 cores). The CRF is implemented by invoking a Pytorch package.² The size of the input/output window is set as $w = 400/s = 64$ for REDD and $w = 200/s = 32$ for UK-DALE. Following (Zhang et al. 2018), for both datasets, we normalize the power readings beforehand through subtracting the mean values and dividing them by the standard deviations. We use the *mean shift clustering algorithm* to pre-extract appliances’ power states and publicize corresponding label data (see our code link¹). In each experiment, results for the report are averaged over 20 independent runs.

5.2 Generalization Capability

To investigate the generalization capability, we take training and test data from different houses: for REDD, house 2 and 3 are for training and house 1 for test; for UK-DALE, the setting turns to house 1 for training and house 2 for test.

The top of Tab. 1 shows the comparisons of our MSDC and MSDC-CRF against baselines on REDD. All the deep learning-based models perform better than FHMM and our models perform the best on almost all appliances. MSDC achieves an average improvement for MAE, SAE, and SAE_δ by 8.96%, 47.61%, and 13.99%, respectively, compared to the best baseline. Further, MSDC-CRF demonstrates even higher improvements on all metrics (10.72%, 51.44%, and 15.87% on average). This demonstrates the advantage of using CRF to capture state transition.

²Pytorch-crf: <https://pytorch-crf.readthedocs.io/en/stable/>

| Model | Metric | house1 | house2 | house3 | Average | Improvement |
|-------|------------------|-------------|-------------|--------------|--------------|---------------|
| S2P | MAE | 8.18 | 4.18 | 12.83 | 8.40 | — |
| | SAE | 23.87 | 29.61 | 74.35 | 42.61 | — |
| | SAE _δ | 5.59 | 2.90 | 12.67 | 7.05 | — |
| SGN | MAE | 7.46 | 1.52 | 10.27 | 6.42 | 0.00% |
| | SAE | 15.46 | 5.18 | 89.90 | 36.85 | 0.00% |
| | SAE _δ | 5.36 | 0.70 | 10.38 | 5.48 | 0.00% |
| MSDC | MAE | 4.96 | 1.09 | 7.87 | 4.64 | 27.73% |
| | SAE | 8.49 | 7.87 | 59.27 | 25.21 | 31.59% |
| | SAE _δ | 2.81 | 0.56 | 7.31 | 3.56 | 35.04% |

| Model | Metric | house1 | house2 | Average | Improvement |
|-------|------------------|-------------|-------------|-------------|---------------|
| S2P | MAE | 12.50 | 4.99 | 8.75 | — |
| | SAE | 34.92 | 1.51 | 18.72 | — |
| | SAE _δ | 9.16 | 2.36 | 5.76 | — |
| SGN | MAE | 6.18 | 4.04 | 5.11 | 0.00% |
| | SAE | 12.52 | 3.29 | 7.91 | 0.00% |
| | SAE _δ | 5.30 | 2.86 | 4.08 | 0.00% |
| MSDC | MAE | 3.24 | 3.51 | 3.38 | 33.86% |
| | SAE | 3.49 | 3.51 | 3.38 | 33.86% |
| | SAE _δ | 2.45 | 2.33 | 2.39 | 41.42% |

Table 2: Results for the dishwasher in REDD (top) and UK-DALE (bottom). Best results are highlighted in bold.

We further compare these baselines that attain better average performance. From the results on UK-DALE (the bottom of Tab 1), we can see that our MSDC performs best on average, achieving an improvement of 6.36%-33.99% on the standard metrics. From Tab 1 we may find that BERT4NILM can achieve better results on some metrics, it, however, cannot ensure to produce robust and consistent performance. On the other hand, BERT4NILM is a transformer that is a very costly option, which requires more time for model training.

We then visualize the results to intuitively display the difference between the predicted results and the truth data. The top of Fig. 3 depicts the predicted power curves for one working cycle of the dishwasher. For a clear demonstration, we only show results of S2P, SGN, and our MSDC-CRF. Among all models involved, MSDC-CRF attains the smallest prediction error and predicts on/off states with the highest accuracy. Moreover, the power levels of all “on” states predicted by MSDC-CRF are much more consistent with the truth. Although the average power level for each state in training houses is not the same as in the testing house, MSDC-CRF achieves the highest accuracy, indicating superior generalization capability of our model. However, no model can well predict the power level in the last time period of the cycle (from 1000s to 1200s in the figure). The reason may be that the dishwashers in two houses belong to different models, and this time period corresponds to a working state which only exists in the dishwasher of test house, but not in the training houses. Considering this mismatching issue, we next move our focus to the predictive ability of our model assuming two appliances match in states.

5.3 Predictive Ability

To make appliances in training and testing matching in states, for both datasets we take the training and testing data from the same house. We set the ratio of the data for training, validation, and testing as 7:1:2. Several previous work

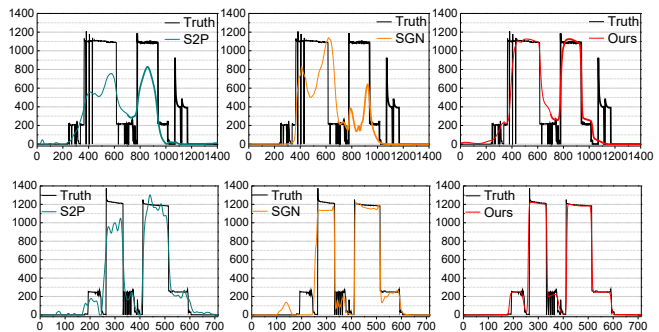


Figure 3: Predicted power curves of the dishwasher.

(Kolter, Batra, and Ng 2010; Elhamifar and Sastry 2015; He et al. 2019b) also adopted a similar setting.

Due to space limitation, we only report the results of S2P, SGN, and our MSDC on the dishwasher. Tab. 2 records the results of dishwasher for two datasets, which reveal that our model achieves significant performance enhancements in all houses. Specifically, the average improvements on MAE, SAE, and SAE_δ are up to 27.73%, 31.59%, and 35.04%, respectively in REDD dataset; 33.86%, 33.86%, and 41.42%, respectively in UK-DALE dataset. We owe the improved performances to the multi-state setting.

The bottom of Fig. 3 depicts the predicted power curves for one working cycle of the dishwasher in house 2 in REDD. The predicted power signal of our model demonstrates a closer resemblance to the true signal compared to baselines. The signal output by our model is also smoother whereas baselines output many nonzero values when the dishwasher is actually “off” or has many abnormal peaks.

6 Conclusions and Future Work

In this paper, we extend NILM task to the multi-state setting and formalize the *multi-state-NILM* (MS-NILM) problem. The problem seeks a model that predict per-appliance power signal through capturing the feature of multiple power states of appliances. We theoretically justify the advantage of the multi-state setting on reducing the sample variance of training data. We then proposed a dual-CNN-based model called MSDC to address MS-NILM, which uses two CNNs to predict the state distribution and acquire the power consumption of each state, respectively. To capture state transitions, we further design a novel technique that incorporates CRF into MSDC, leading to a variant model called MSDC-CRF. Experimental results show that our model has the excellent ability in recovering power consumption and good capability to predict for unseen appliances.

As future work, one could attempt to generalize MSDC to more general setups, say, Type-3 appliances which have continuous state space. Possible ideas include discretizing the state space or parametrizing the state functions. Another future work is to scale our model to other BSS problems such as speech separation and recognition where each speech signal corresponds to a specific identifiable hidden state.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grant No.62172040, National Key R&D Program of China under Grant No.2022YFB3103500, and National Natural Science Foundation of China under Grants No.U1836212, No.61872041 No.U20A20176 and No.62072062.

References

- Alahakoon, D.; and Yu, X. 2016. Smart Electricity Meter Data Intelligence for Future Energy Systems: A Survey. *IEEE Trans. Ind. Informatics*.
- Azaza, M.; and Wallin, F. 2017. Finite state machine household's appliances models for non-intrusive energy estimation. *Energy Procedia*.
- Chen, K.; Zhang, Y.; Wang, Q.; Hu, J.; Fan, H.; and He, J. 2019. Scale-and Context-Aware Convolutional Non-Intrusive Load Monitoring. *IEEE Transactions on Power Systems*.
- Cheng, Y. 1995. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence*, 17(8): 790–799.
- Çimen, H.; Çetinkaya, N.; Vasquez, J. C.; and Guerrero, J. M. 2020. A Microgrid Energy Management System based on Non-Intrusive Load Monitoring via Multitask Learning. *IEEE Transactions on Smart Grid*.
- Dinesh, C.; Makonin, S.; and Bajić, I. V. 2019. Residential power forecasting using load identification and graph spectral clustering. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 66(11): 1900–1904.
- Elhamifar, E.; and Sastry, S. 2015. Energy Disaggregation via Learning Powerlets and Sparse Coding. In *Proc. of AAAI*.
- Fukunaga, K.; and Hostetler, L. 1975. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1): 32–40.
- Green, D.; Lindahl, P.; Leeb, S.; Kane, T.; Kidwell, S.; and Donnal, J. 2019. Dashboard: Nonintrusive electromechanical fault detection and diagnostics. In *2019 IEEE AU-TOTESTCON*, 1–9. IEEE.
- Hart, G. W. 1992. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*.
- Hassan, M. U.; Rehmani, M. H.; Du, J. T.; and Chen, J. 2022. Differentially private demand side management for incentivized dynamic pricing in smart grid. *IEEE Transactions on Knowledge and Data Engineering*.
- He, H.; Liu, Z.; Jiao, R.; and Yan, G. 2019a. A novel nonintrusive load monitoring approach based on linear-chain conditional random fields. *Energies*.
- He, J.; Zhang, Z.; Zhu, L.; Zhu, Z.; Liu, J.; and Gai, K. 2019b. An Efficient and Accurate Nonintrusive Load Monitoring Scheme for Power Consumption. *IEEE Internet of Things Journal*.
- Hsu, C.-Y.; Zeitoun, A.; Lee, G.-H.; Katabi, D.; and Jaakkola, T. 2019. Self-supervised learning of appliance usage. In *Proc. of ICLR*.
- Kelly, J.; and 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*.
- Kelly, J.; and Knottenbelt, W. 2015b. The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes. *Scientific data*.
- Kim, H.; Marwah, M.; Arlitt, M.; Lyon, G.; and Han, J. 2011. Unsupervised disaggregation of low frequency power measurements. In *Proc. of SDM*.
- Kolter, J. Z.; Batra, S.; and Ng, A. Y. 2010. Energy disaggregation via discriminative sparse coding. In *Proc. of NeurIPS*.
- Kolter, J. Z.; and Jaakkola, T. 2012. Approximate inference in additive factorial hmms with application to energy disaggregation. In *Artificial intelligence and statistics*.
- Kolter, J. Z.; and Johnson, M. J. 2011. REDD: A public data set for energy disaggregation research. In *Workshop on data mining applications in sustainability (SIGKDD)*, San Diego, CA.
- Kong, W.; Dong, Z. Y.; Hill, D. J.; Luo, F.; and Xu, Y. 2017. Short-term residential load forecasting based on resident behaviour learning. *IEEE Transactions on Power Systems*, 33(1): 1087–1088.
- Kong, W.; Dong, Z. Y.; Hill, D. J.; Ma, J.; Zhao, J.; and Luo, F. 2016. A hierarchical hidden Markov model framework for home appliance modeling. *IEEE Transactions on Smart Grid*.
- Lafferty, J. D.; McCallum, A.; and Pereira, F. C. N. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. of ICML*.
- Mauch, L.; and Yang, B. 2015. A new approach for supervised power disaggregation by using a deep recurrent LSTM network. In *Proc. of GlobalSIP*.
- Mauch, L.; and Yang, B. 2016a. A novel DNN-HMM-based approach for extracting single loads from aggregate power signals. In *Proc. of ICASSP*.
- Mauch, L.; and Yang, B. 2016b. A novel DNN-HMM-based approach for extracting single loads from aggregate power signals. In *Proc. of ICASSP*.
- Naik, G. R.; Wang, W.; et al. 2014. Blind source separation. *Berlin: Springer*.
- Pan, Y.; Liu, K.; Shen, Z.; Cai, X.; and Jia, Z. 2020. Sequence-to-subsequence learning with conditional gan for power disaggregation. In *Proc. of ICASSP*.
- Rashid, H.; Singh, P.; Stankovic, V.; and Stankovic, L. 2019. Can non-intrusive load monitoring be used for identifying an appliance's anomalous behaviour? *Applied energy*, 238: 796–805.
- Shao, H.; Jiang, H.; Zhao, H.; and Wang, F. 2017. A novel deep autoencoder feature learning method for rotating machinery fault diagnosis. *Mechanical Systems and Signal Processing*.

- Shin, C.; Joo, S.; Yim, J.; Lee, H.; Moon, T.; and Rhee, W. 2019. Subtask gated networks for non-intrusive load monitoring. In *Proc. of AAAI*.
- Sudoso, A. M.; and Piccialli, V. 2019. Non-intrusive load monitoring with an attention-based deep neural network. *arXiv e-prints*, arXiv-1912.
- Wang, X.; Zhang, M.; and Ren, F. 2018. Learning customer behaviors for effective load forecasting. *IEEE Trans. Knowl. Data Eng.*, 31(5): 938–951.
- Wilson, C.; Hargreaves, T.; and Hauxwell-Baldwin, R. 2015. Smart homes and their users: a systematic analysis and key challenges. *Personal and Ubiquitous Computing*.
- Yue, Z.; Witzig, C. R.; Jorde, D.; and Jacobsen, H.-A. 2020. BERT4NILM: A Bidirectional Transformer Model for Non-Intrusive Load Monitoring. In *Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring*.
- Zhang, C.; Zhong, M.; Wang, Z.; Goddard, N.; and Sutton, C. 2018. Sequence-to-point learning with neural networks for nonintrusive load monitoring. In *Proc. of AAAI*.
- Zia, T.; Bruckner, D.; and Zaidi, A. 2011. A hidden Markov model based procedure for identifying household electric loads. In *Proc. of IECON*.