Non-Intrusive Load Monitoring based on Deep Neural Network and Differential Current

Shaokang Ding\textsuperscript{1,3}, Hongwei Zhu\textsuperscript{2,4}

\textsuperscript{1}Institute of VLSI design, Zhejiang University, Hangzhou, Zhejiang, 310027, China
\textsuperscript{2}Institute of VLSI design, Zhejiang University, Hangzhou, Zhejiang, 310027, China
\textsuperscript{3}21731083@zju.edu.cn; \textsuperscript{4}zhuhw@vlsi.zju.edu.cn

Abstract. In daily life, it is easy to obtain the power consumption information of a house by installing an energy meter at the entrance. The non-intrusive load monitoring method can decompose the power consumption and the working state of a specific load according to the aggregated information of the power inlet, and then help people to save electricity. Here, we present a novel and complete non-intrusive load identification algorithm based on deep neural network and differential current. The algorithm includes several modules for data preparation, event detection, feature extraction and load identification, and uses the BLUED dataset to test the feasibility of the proposed algorithm.

1. Introduction

In recent years, the global energy consumption level has been continued increasing, and the non-renewable fossil energy still dominates the world’s energy consumption. Energy, as an important lifeline in the process of social and economic development, is now attracting attention from all over the world and all mankind.

Numerous studies have shown that feedback to consumers on the details of power consumption can significantly reduce power consumption\cite{1}. The more timely and detailed feedback on power consumption, the more obvious effect on energy saving we can see. In the early days, there were two ways to get a feedback: the intrusive and the nonintrusive. Intrusive load identification requires separate monitoring of each load. This means installing a sensor for each load, which is economically expensive. In contrast, non-intrusive load identification only requires one sensor to be installed at the single point source such as smart meter, then the total energy is decomposed into the energy of every load according to the algorithm. Today, intrusive load identification has been almost completely replaced by non-intrusive research.

In this paper, we propose a new event based non-intrusive load identification method. The proposed method can be simply divided into four parts: data preparation, event detection, feature extraction, and load identification. In the event detection part, we use power envelope and neural network to treat event detection as a simple binary classification task. In load identification and feature extraction part, we propose differential current method and fusion feature method for identification. At the same time, some optimization details such as data augmentation and optimization loss function were used to improve the model during the experiment. And the BLUED dataset is used to test the practicability of our proposed method.

This paper is organized as follows, Section 2 reviews on the basic theory of NILM and some of the latest algorithms. Section 3 introduces the event detection algorithm we use and the experimental
result. In section 4, we introduce the algorithm of the load identification part and the analysis of the results. Finally, section 5 makes some conclusions and prospects for this paper.

2. Related Work
A simple load energy disaggregation problem can be described using Equation 1, where N is the total number of loads in the circuit, X represents the total current or energy signal of the circuit, \( \delta(t) \) means the error between the total signal and the sum of individual load signal. The purpose of non-intrusive load monitoring is to obtain respective signals \( y_i(t) \) of each load from the total signal \( X(t) \).

\[
X(t) = \sum_{i=1}^{N} y_i(t) + \delta(t)
\]  

Research on NILM was first started by George Hart in the mid-1990s[2], professor Hart and his team used switch state template matching to decompose some high-power loads. Unfortunately, NILM technology wasn’t received much attention for a long time afterwards. Until the energy has become tense in recent years, NILM technology has been reemphasized by some laboratories and power grid companies. Henning et.al.[3] proposed a deep learning approach, this approach uses high-frequency measurements of current, assume that there are only two working modes in every load, and map the current signal to the binary feature space and finally perform load identification. Barsim et.al.[4] proposed an algorithm named Neural Network Ensembles to Real-time Identification of Plug-level Appliance Measurements to solve NILM problem, in this algorithm every network takes a high-frequency current and voltage signals as inputs. The work by Kelly et.al.[5] used three different deep neural network structures to process low frequency current signals, then the results are compared of these three neural networks. Paulo et.al.[6] presented a comparison of variety of CNN and RNN across a number of appliances. Zhang et.al.[7] used a sequence-to-point network based on CNN for energy disaggregation, the input of the network is a mains window and the output is the midpoint element of the corresponding window of the target appliance. From the perspective of algorithms, the above studies can be roughly divided into two categories. The first one is the event-based load identification method. It first detects whether the working state of the load has changed in the circuit, then identifies which load the event occurred in. The other is a direct steady-state signal decomposition method, which directly decompose the steady-state working signal to get the status of each load.

The experimental data is from BLUED[8], which is a high-frequency and an event-based dataset published by Kyle et.al. In this dataset, the events are defined as any change in power consumption greater than 30 watts and lasting at least 5 seconds. The sampling frequency used by this dataset is 12KHz, and the US grid frequency is 60hz, which means that one cycle of the signal has 200 sample points. The figure 1 and figure 2 below shows the total current signal collected from the meter at the moment when a load is turned on and off in the data.

![Figure 1. Total current change when refrigerator is turned on.](image1)

![Figure 2. Total current change when refrigerator is closed.](image2)
3. Event Detection

Event detection is to detect whether the working state of the load has changed from the total current signal, which is similar to abnormal point detection. In this section, the algorithm we use is a deep neural network. The input of the network is the power envelope vector of a circuit signal and the output is the probability of the event contained in this signal.

3.1. Power envelope extraction

In the BLUED dataset, the data label mainly includes two parts, one is the time stamp accurate to millisecond, and the other is the load mark, which indicates that the working status of a certain load has changed at a certain moment. According to this label, we can lock the location of the event, then intercept 60 cycles around the event location and calculate the power of each cycle. The power value constitutes a 60-dimensional vector. This vector is the power envelope of the event signal. It intuitively reflects the change of the power in the circuit when the event occurs. In addition, the paper also tried the peak current envelope, effective current envelope and so on for comparative experiments to find the most suitable characteristics of the characterization event.

3.2. Deep neural network

Because the event detection algorithm is different from the load identification part algorithm and the event detection algorithm needs to run continuously in the device, so it’s speed is required to be high. Therefore, this paper only uses the simplest fully connected neural network to detect events (Perhaps recurrent neural network is more suitable for processing time-series signals, but the amount of calculation is relatively large). The output layer of the neural network has only one node, and the output value of this node indicates the probability that this signal sample is an event.

3.3. Batch normalization

Deep neural network involves the superposition of many layers. During the training process of the deep network, the parameter update of each layer will cause the change in distribution of the input data of the next layer. With the number of layers deepens, the network input distribution of the top layer changes, which will be very violent. However, the top-level network needs to constantly adapt to changes in the distribution of input data caused by the underlying parameter updates, making the network difficult to train. Ioffe et.al.[9] proposed a batch normalization (BN) method. Its basic idea is to translate and scale the network input $x$ before it is sent to neurons, and normalize the distribution of $x$ into a standard distribution of fixed interval. The following formula is the implementation of BN.

$$
\xi_i = y_i \cdot \frac{x_i - \mu_i}{\sigma_i} + \beta_i
$$

(2)

3.4. Focal loss

For event detection in load monitoring, the data with the event has very few parts compared to the data without the event, which is called the imbalance of sample distribution. In the previous deep learning, hard example mining was often used to solve this problem, but this method was very troublesome to operate. Later, Lin, T. Y. proposed the focal loss which can be very elegant to solve the sample imbalance problem[10]. The formula for focal loss is as follows, where $\alpha_t$ is a balance factor and $\gamma$ is a adjustment factor which can solve sample unbalanced problem.

$$
P_t = \begin{cases} 
  p, & \text{if } y = 1 \\
  1 - p, & \text{otherwise}
\end{cases}
$$

$$
Focal\_Loss(p_t) = -\alpha_t(1 - p_t)^\gamma \log p_t
$$

(3)

(4)

3.5. Experimental details and Result analysis

In the event detection experiment, We use a fully connected neural network, where the hidden layer activation function uses RELU[11] and the output layers uses SIGMOID[12].
can normalize the output of a neural network to $0 \sim 1$, which only indicates the probability of the event. Here, we use gradually reduced learning rate, adagrad’s learning method, and 128 batch size to iterate for 60 epochs. Note that training should use a small learning rate when not using the BN layer.

In order to evaluate the performance of the algorithm, we did a series of experiments on the previously processed data. The experimental results are shown in table 1. The GLR algorithm was proposed by Kyle Anderson[13], Kely Anderson is also one of the builders of the BLUED dataset. Therefore, the GLR detection algorithm is used as the benchmark detection algorithm for datasets in BLUED papers. And figure 3 shows the ROC curves of event detection under different algorithms. It can be seen from the results that the use of power envelope is indeed better than the use of effective current, and the ability of the model does improve in the case of using BN and focal loss. The effect of phase b is obviously worse than that of phase a. This is because the loads contained in phase b are mostly electronic loads and some low-power loads.

| Method             | Optimization | Result(Missing) |
|--------------------|--------------|-----------------|
|                    | Batch        | Focal loss      | Phase a | Phase b |
| GLR                | -            | 16              | 459     |
|                   | normalisation|                 |         |
| Effective current  | √            | 26              | 302     |
|                   | √            | 15              | 269     |
| Real-time power    | √            | 13              | 177     |
| ROC                |              | 17              | 222     |

4. Load Identification

In this section, we continue to explain the algorithm details and experimental results of the load identification part. In this paper, we propose a new differential current based load identification method. Unlike the most common ones mentioned earlier, which directly decompose the steady-state signal, the
method we use is to get a difference from the steady-state signals before and after the event to obtain a separate signal of the load that caused the event, and then identify this separate signal.

4.1. Differential current extraction
As can be seen in Equation 1, the current in the circuit at each moment is the sum of the divided currents of all loads. In this case, after detecting the event in the previous step, we intercept the steady-state current from the two ends of the event, and then get the difference, we can get the individual current of the load which caused this event. At the same time, in order to minimize internal noise, we used a steady-state differential currents of 5 cycles to average the final load current.

In addition, because there are too few events in the load, it is too difficult if the event is used directly to judge the load. But based on the differential current method proposed in this paper, a large amount of steady-state current can be taken from both ends of the event to get a difference. Thus a large amount of load data can be generated, which is more conducive to algorithm optimization and evaluation. Figure 4 and figure 5 show the current curves of different loads obtained by our differential current method.

4.2. Data augmentation
In the BLUED dataset, a total of more than 30 kinds of load switching states are included within a week. However, it is difficult to learn the characteristic of those loads that only change their working status once a week. Therefore, the data augmentation is necessary. As mentioned earlier, the differential current method we use is obtained from the steady-state signals, because we only need one cycle of signal to identify composite load, we can obtain a large number of differential currents from the current before and after the load state changes for model training to improve model recognition ability.

4.3. Feature extraction and identification algorithm
At present, the commonly used features include time domain features and frequency domain feature. However, because of the limitations of artificial feature extraction, more and more people use neural network to extract features with the extensive application of the deep learning. Here, we use the de-noise auto encoder [14] to extract features and compare the experimental results with traditional features.

Since the artificial features used in this experiment include 33 time and frequency domains, the number of output nodes of the coded layer of the de-noise auto encoder is also 33, which makes the experimental results more meaningful. At the end of the experiment, the effect of the fusion characteristics of the two on the load recognition was tested. Then we can simply analyze the relationship between the two features from the experimental results.

After the feature extraction is completed, it is time to identify the extracted features to determine which load this is. The input of the network is the extracted feature vector, and the output is the probability corresponding to each load. According to the different feature vectors, the input layer needs to be adjusted accordingly in the experiment.
4.4. Center loss
In order to increase the recognition ability of the model, this paper also uses center loss[15]. The equation of center loss is shown in (5). The parameter $c_{yi}$ in the formula represents the center point of the category $yi$. As can be seen from the publicity, the optimization goal of this loss function is to make the final result of each class as close to the center of this category as possible.

$$L_c(x) = \frac{1}{2} \sum_{i=1}^{m} \|x_i - c_{yi}\|_2^2$$  \hspace{1cm} (5)

4.5. Experimental details and Result analysis
As before, the hidden layer activation function uses RELU, and the last layer activation function uses SIGMOID. And we also use BN in every experiment. The basic loss function is CE loss, and the ratio of the training set to the test set is 7:3. Table 2 shows our final load identification results. It can be seen from the table that after using data augmentation and center loss, the effect has been significantly improved, especially the effect of data augmentation. In addition, it can be seen from the table that the abstract feature extracted by the de-noise auto-encoder is not as good as the artificially extracted feature effect, but the effect is also improved after combining the two, indicating that the features extracted by DAE can make up for the lack of artificial features to a certain extent.

| Method          | Optimization | Result (accuracy) |
|-----------------|--------------|------------------|
|                 | Data          | Center loss | Train Set | Test Set |
|                 | augmentatio   |             |           |         |
| Artificial      | a             | 0.865       | 0.853     |         |
| feature         | √             | 0.936       | 0.934     |         |
|                 | √             | 0.950       | 0.949     |         |
| DAE-feature     | √             | 0.868       | 0.865     |         |
|                 | √             | 0.876       | 0.872     |         |
|                 |               | 0.879       | 0.864     |         |
| Fusion feature  | √             | 0.958       | 0.955     |         |
|                 | √             | 0.967       | 0.965     |         |

Then, in order to visually observe the distribution of each type of load in space after passing through the neural network, we extracted four types from the original dataset (more than 30 types in the original dataset) to visualize in 3D space and visualize the results. As shown in figure 6, it can be seen that under the action of center loss, different loads are distributed in different clusters in space.
5. Conclusion
This paper first introduced the current mainstream load identification method, and then introduced our own load identification method. Firstly, the power envelope within one second and neural network algorithm are used to detect the event. Secondly, we use the differential current to identify the load and determine which load is causing the event. Through experimental analysis on the BLUED dataset, it shows that the method of load identification method proposed in this paper is feasible. However, the method of this paper still has limitations, and there is still a long way to go before it can be put into use. Firstly, the problem of load pattern is not considered in the dataset used in this experiment. For example, some loads may switch between multiple working modes. Secondly, the dataset rarely involves simultaneous effects of multiple loads. Although they rarely appear in actual environments, they still need to be analyzed in future experiments.

References
[1] Armel, K. C., Gupta, A., Shrimali, G., Albert, A. (2013). Is disaggregation the holy grail of energy efficiency? The case of electricity. Energy Policy, 52: 213-234.
[2] Hart, G. W. (1992). Nonintrusive appliance load monitoring. In: Proceedings of the IEEE, 80(12), 1870-1891.
[3] Henning, L., Bergés, M. (2016). The neural energy decoder: Energy disaggregation by combining binary subcomponents. In Proc. 3rd Int. NILM Workshop. Vancouver. pp. 1-5.
[4] Barsim, K. S., Mauch, L., Yang, B. (2018). Neural network ensembles to real-time identification of plug-level appliance measurements. https://arxiv.xilesou.top/abs/1802.06963.
[5] Kelly, D. (2016). Disaggregation of domestic smart meter energy data. https://spiral.imperial.ac.uk/handle/10044/1/49452
[6] do Nascimento, P. P. M. (2016). Applications of deep learning techniques on NILM. PhD, thesis.
[7] Zhang, C., Zhong, M., Wang, Z., Goddard, N., Sutton, C. (2018). Sequence-to-point learning with neural networks for non-intrusive load monitoring. https://arxiv.xilesou.top/abs/1612.09106.
[8] Anderson, K., Oconeau, A., Benitez, D., Carlson, D., Rowe, A., Berges, M. (2012). BLUED: A fully labeled public dataset for event-based non-intrusive load monitoring research. In: 2nd KDD workshop on data mining applications in sustainability (SustKDD). Beijing. pp. 1-5.
[9] Ioffe, S., Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. https://arxiv.org/abs/1502.03167.
[10] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. Venice. pp. 2980-2988.

[11] Nair, V., Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10). Haifa, Israel. pp. 807-814.

[12] Han, J., Moraga, C. (1995). The influence of the sigmoid function parameters on the speed of backpropagation learning. In: International Workshop on Artificial Neural Networks. Springer, Berlin, Heidelberg. pp. 195-201.

[13] Anderson, K. D., Bergés, M. E., Ocneanu, A., Benitez, D., Moura, J. M. (2012). Event detection for non intrusive load monitoring. In: IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society. Montreal. pp. 3312-3317.

[14] Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P. A. (2008). Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on Machine learning. Helsinki, Finland. pp. 1096-1103.

[15] Wen, Y., Zhang, K., Li, Z., Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. In: European conference on computer vision. Amsterdam. pp. 499-515.