IEHouse: A Non-Intrusive Household Appliance State Recognition System

Xingzhou Zhang*§, Yifan Wang*§, Lu Chao*§, Chundian Li*§, Lang Wu*§, Xiaohui Peng§ and Zhiwei Xu*§
*University of Chinese Academy of Sciences, Beijing, China
§Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{zhangxingzhou, wangyifan2014, chaolu, lichundian, wulang, pengxiaohui and zxu}@ict.ac.cn

***Abstract*—Recognizing the states of household appliance is helpful to monitor the power consumption and model user behaviors at home. Non-Intrusive Load Monitoring (NILM) receives widespread attention as it can identify a individual appliance state using a single sensor. However, presented approaches today can not be adopted in actual home scenarios because they either ignore the energy limitation of sensors or require a complex user configuration. To solve this problem, this paper proposes IEHouse which is a Non-Intrusive Household Appliance State Recognition System. It leverages a supervised learning process over the labeled appliance data sets which can be constructed dynamically based on a small number of appliance profiles. It uses Deep Neural Network (DNN), Convolutional Neural Network (CNN) and Gated Recurrent Unit (GRU) models, to identify appliance states and improves the accuracy through online learning gradually. By simulating a common household scenario, the energy consumption of sampling sensor is 5.12kJ per week and the average accuracy of recognizing 10 mixed typical appliance states is 92.9%, which achieves better accuracy with low energy.**

***Keywords*-Smart Home, Non-Intrusive Load Monitoring (NILM), Appliance State Recognition, Deep Neural Network (DNN)**

I. INTRODUCTION

With the development of artificial intelligence and smart home, recognizing appliance states has become an important research topic [1], [2]. The motivations for such a process are twofold. First, based on the researches, users can monitor the operating states of appliances to optimize energy consumption [3]. Second, since appliance states can provide information about users' lives, it is helpful to identify and predict users' behaviors. For instance, if the states of kitchen appliances are ON, it can be presumed that the user is cooking; if the TV and lights in the living room are ON, then the probability that users are resting will be high (i.e., *Household Appliances Are Sensors (HAAS)*).

However, if too many sensors are installed in a house, it will bring issues of privacy, user experience, cost and etc. Therefore, it's important for us to recognize as many appliance states as possible with fewer sensors. Non-Intrusive Load Monitoring (NILM) [4] aims to identify every individual appliance from the aggregate data collected via a single sensor. At present, researches in this field can be divided into two aspects: supervised learning and unsupervised learning. Supervised learning is used when the model is trained using aggregate data which is labeled to identify individual appliance states, which achieves high accuracy in some experiment setups. Unsupervised learning trains with aggregate data only, and no prior training with labeled data is required. However, supervised learning is impractical because it requires a long manual labeling process and unsupervised learning has problems in obtaining high accuracy.

Combining the strengths of both methods, we present IEHouse, a Non-Intrusive Household Appliance State Recognition System, which can achieve high accuracy with little user involvement. In IEHouse, a current sampling sensor is installed over the household main power entrance line. Users add new appliances by scanning the barcode on appliances. The training dataset can be composed dynamically by appliance profiles provided by manufacturers, rather than from data labeled by users. Inspired by the similarity between voice waves and current waves, we adopt the state-of-the-art DNN model, Gated Recurrent Unit (GRU) [5], to disaggregate each appliance's current signal from household aggregate signals. Moreover, online learning mechanism is designed to improve accuracy and increase training speed. We implement the prototype system and evaluate the performance in a lab environment. The contributions are summarized as follows:

1) A novel system, IEHouse, is proposed to recognize the states of appliances without intruding occupants in the house, which is an implementation of NILM for actual home scenario. IEHouse achieves simple configuration, low energy consumption, and low cost at the same time. Three indicators including energy-awareness, scalability and effectiveness are used to evaluate the performance of the proposed system.
2) A novel algorithm based on supervised learning model is used to recognize the states of appliances. As a supplement, online learning based on user feedbacks is leveraged to improve the accuracy. The average accuracy of 92.9% is achieved in our experiments.
3) More detailed Behavior-awared Sampling Interval (BSI) is put forward to achieve energy monitoring for sampling sensor with lower energy consumption, and the energy consumption of the sampling sensor is estimated as $5.12kJ$ per week.

The remainder of this paper is organized as follows. In Section II we introduce the related work. Section III describes the IEHouse in detail and the core algorithm of IEHouse is illustrated in IV. In section V, we evaluate our system and we conclude in Section VI.

II. Related Work

Appliance state recognition systems can be divided into four categories according to the way of collecting features: smart outlet system, indirectly sensing system, noise detection system and non-intrusive load monitoring system [6].

Smart outlet system consists of several smart outlets and a computing unit. Smart outlet which transformed from the traditional outlet can detect the state of appliance and send it to the computing unit. Computing unit summarizes and analyzes the information from smart outlets. Study in [7] show that wireless sensor network technology is widely used in appliance state recognizing and they focus on evaluating the reliability and stability of the smart outlet network by deploying 49 nodes in a building. Such techniques are effective but expensive to deploy and may also raise privacy concerns in a home.

Indirectly sensing system, which leverages a variety of heterogeneous sensors, can identify appliance state by analyzing context information, such as electromagnetic field, sound and vibration in the vicinity of appliance. In [8], the authors presented ViridiScope which used magnetic field sensors to estimate the power consumption of a device. TinyEARS [9] monitored the power consumption of individual appliance primarily based on the acoustic signatures and reported the power consumption within a 10% error margin. HeatProbe [10], a thermal-based power meter system, used thermal imaging to track disaggregated appliance usage. However, this approach also adopts a large number of sensors, which brings serious limitations to the system.

Noise detection system can recognize the state of appliance based on voltage or current noise because each appliance state is accompanied by a unique circuit noise. These noises are mainly categorized into three types: on-off transient noise, steady-state line voltage noise and steady-state continuous noise. Gupta *et al.* [11] proposed ElectriSense which could sense the electromagnetic interference (EMI) created by Switched Mode Power Supplies (SMPS) oscillators. Appliances which equipped with SMPS can be characterized by analyzing steady-state voltage noise generated upon their operation. A significant improvement over ElectriSense was presented in [12]. The new technique did not required precise placement on the breaker panel anymore and it adopt a self-calibration technique using a neural network that dynamically learned the transfer function. These techniques can estimate true power consumption with an average accuracy of 95%. However, this method usually requires high sampling rate in order to capture the noise which is expensive and is sensitive to the wiring architecture of the monitored environment.

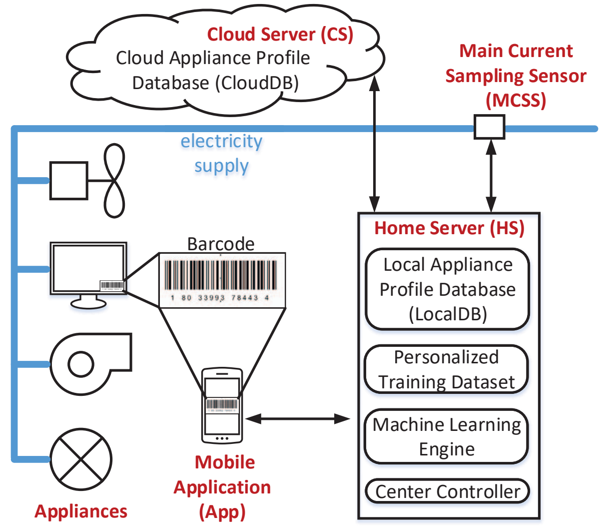

Figure 1. The framework overview

Non-intrusive load monitoring (NILM) system [4] aims to discern devices by identifying a single measurement from the aggregate data. The sensor is usually used to collect current or power signal in the home entrance line. The research of NILM focuses on algorithm, which falls into two categories: supervised learning and unsupervised learning. Supervised learning is used when the model is trained using the aggregate data and individual appliance states. Many researches used the techniques such as Hidden Markov Models (HMM) [13], Viterbi's algorithm and sparse coding algorithms [14] to infer the disaggregated state. Recently, [15] adapted three deep neural network architectures and drew a conclusion that all three neural nets achieved better performance than either combinatorial optimization or factorial hidden Markov models. A DNN-HMM-based approach was proposed in [16] which outperforms Factorial Hidden Markov model (FHMM) because there is no need of knowledge about the remaining loads in the aggregate signal except for the target load. Unsupervised learning had been presented in NILM trying to reduce cost as much as possible [17], [18]. Different variations of HMM are used to address this issue [19]–[21]. However, supervised learning requires a time-consuming manual labeling process and unsupervised learning falls short of achieving high accuracy.

Overall, an appliance state recognition system that can achieve low cost, low energy consumption, easy deployment and high accuracy is required. IEHouse which falls within the category of NILM can meet these requirements.

III. System Design

IEHouse consists of five components: Household Appliances, Mobile Application (App), Cloud Server (CS), Home Server (HS) and Main Current Sampling Sensor (MCSS).

The five components are described in detail in the first part of this section. Then the following part analyzes the system workflows. The overall framework of the system is illustrated in Figure 1.

A. Components

Household Appliances: Every appliance owns a barcode affixed by the manufacturers, we regard the barcode as ID of one appliance. As the operating states of appliance vary from each other, we categorize appliances based on their operation states as follows [4]:

1) Two State (TS) appliances. They only have two opposite states: ON and OFF. Such as table lamp.
2) Finite State Machine (FSM) appliances. They own finite and more than one running state. Such as cooker.
3) Continuously Variable (CV) appliances. They have no fixed number of states. Such as computer.

We regard current signal as the feature of an appliance state (The reason that we focus on current will be explained in Section IV-A).

It is easy to obtain the states of TS and FSM appliances, but not for the CV appliances. IEHouse puts forward an approach to get the profile of the CV appliances. It samples $M(M >> m)$ cycles of current waveform and clusters the M cycles into x groups according to the K-means clustering algorithm. It choose m cycles of current waveform randomly from each group to indicate states of the appliance. In general, x is set to no more than four to prevent the exploding of the number of states.

Mobile Application(App): It is a mobile application installed on personal smart devices, such as smartphone, tablet and so on. It mainly supports three functions:

1) User login;
2) Adding a new appliance by scanning the barcode;
3) Monitoring appliance states in real time.

Cloud Server(CS): It contains a **Cloud Appliance Profile Database (CloudDB)** which stores all profiles of appliances in the world. CloudDB is opened up to appliance manufacturers and registered users. Manufactures can import profiles of their products in a batched way. Meanwhile, users can import or export profiles that they owned.

Home Server(HS): It is a core component of IEHouse containing four parts:

1) **Local Appliance Profiles Database (LocalDB)**: It stores the profiles, states and trained models of appliances that only the user used. Profiles can be downloaded from CloudDB by identifying the barcode. Of course, users can add the profiles that can not be retrieved from the CloudDB.
2) **Personalized Training Dataset**: It can be generated from the LocalDB and extended dynamically according to the user behavior.
3) **Machine Learning Engine**: It consists of two computing processes. When a new appliance is added, the Neural Network models will be trained according to the personalized training dataset. Another process is inferring the states of all the appliances in the house. Appliance states will be stored to LocalDB.
4) **Center Controller**: It controls the executing logic of HS, and communicates with CS, App and MCSS.

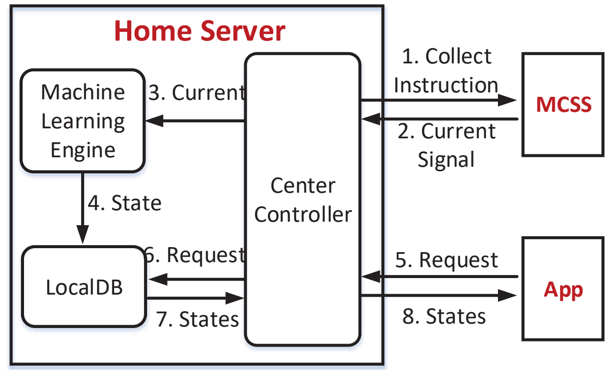

Figure 2. The data flow diagram of normal working

Home Server can be deployed on a PC or an intelligent router, or just be a single device like Google Home[1] and Amazon Echo[2]. Users own the right to manage the data and decide whether to open the data to others or not.

Main Current Sampling Sensor(MCSS): It adheres over the home entrance line, samples and sends current signal to HS.

B. System Workflow

In general, HS sends the instruction of sampling to the MCSS and acquires the current signal. Then the machine learning engine within HS makes inferences based on the trained Neural Network models (As is shown in Figure 2). The inference results, including all states of appliances, will be stored in the LocalDB. Finally users can monitor all states via App.

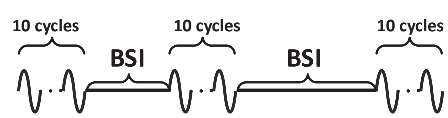

Figure 3. Behavior-awared Sampling Interval (BSI)

The interval between the adjacent instructions of sampling is variable. A shorter interval provides more states and increases the accuracy of predicating while a longer one reduces the energy consumption. To achieve both lower energy

[1]https://madeby.google.com/home/

[2]https://www.amazon.com/Amazon-Echo-Bluetooth-Speaker-with-WiFi-Alexa/dp/B00X4WHP5E

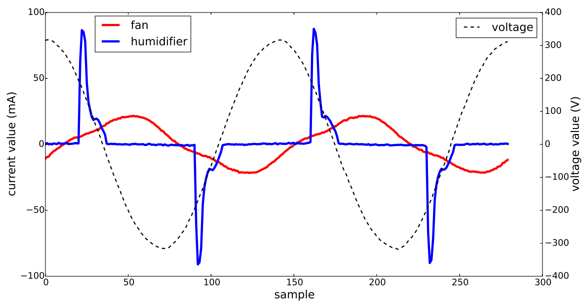

Figure 4. Current waveforms of fan and humidifier

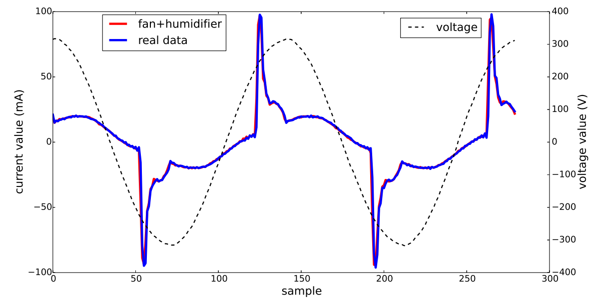

Figure 5. Actual and constructed current waveforms of fan add humidifier

consumption and higher accuracy, we put forward a novel dynamic sampling interval mechanism namely **Behavior-awared Sampling Interval (BSI)** (As is shown in Figure 3). It predetermines the interval based on the house profile and adjusts dynamically based on the daily routine. Firstly, we define the peak time interval and the off-peak one to describe the operating states of appliances. The peak one is defined when more than three appliance state changes are detected in an hour. On the contrary, the off-peak one is defined when no more than three state changes are detected in an hour. The BSI is set to one minute during the peak interval, while ten minutes during the off-peak interval. When IEHouse is used in the first week, the peak and off-peak intervals can be preset by the users or by default. Sequentially, they will be adjusted every week according to the collected appliance states information in the LocalDB.

When a new appliance is registered, Home Server retrieves the appliance's barcode from the App and then queries the LocalDB and CloudDB. If the barcode exists in LocalDB, its profile can be obtained directly. Otherwise its profile will be downloaded to LocalDB from CloudDB if exists. There is a corresponding Neural Network model for every appliance. A new training dataset will be built from the new profile and the existing appliances models will be trained online and updated based on the new training dataset. Meanwhile, the new appliance will lead to a new model training based on the new training dataset and the original training dataset.

IV. ALGORITHM IMPLEMENTATION

A. Feature selection

Application feature is essential to the accuracy of the proposed algorithm. It can be broadly categorized into steady-state and transient features. In order to capture the transient features, the sampling frequency should be extremely high, which results in high hardware costs and energy consumption. So transient features are not suitable for the IEHouse. For steady-state, V-I features of frequency domain also requires a high sampling rate. Therefore, we choose the current waveform signal in time domain as the feature of appliance state.

Considering sampling frequency, on the one hand, if the number of samples per cycle is insufficient, the current waveform may be plotted less accurately. On the other hand, higher frequencies can certainly lead to a higher accuracy, but it incurs sophisticated hardware and over fitting problems. Experiments in [22] indicate sampling frequency above 8kHz does not provide significant benefit for disaggregation since the signal will be buried in the noise. We adopt 7KHz as the sampling frequency in this paper with the consideration of hardware environments.

B. Pre-process

To reduce the complexity of MCSS and decrease the energy consumption, we only collect the total current signal and the voltage is not collected. However, the baseline is lost when only collecting the current, which leads to the diversity of current phases. So it is important to remove the uncertain phase. In the paper, an algorithm based on Fast Fourier transform (FFT) is proposed to remove the uncertain phase of current. First, FFT is used to derive the frequency spectrum of the current waveform:

$$F(\xi) = \int_{-\infty}^{+\infty} f(t)e^{-2\pi it\xi}dt \tag{1}$$

Where $f(t)$ is the time-domain current function, ξ is the frequency, i represents the imaginary unit. $f(\xi)$ is an array, each item is a imaginary number including the real and imaginary parts. The ith item represents the ith harmonic. The phase of fundamental wave can be extracted:

$$ph = \frac{F(\xi)_{(image,1)}}{F(\xi)_{(real,1)}} \tag{2}$$

Then the uncertain phase of every harmonic can be removed:

$$F(\xi)' = F(\xi)e^{-inph} \tag{3}$$

Where i represents the imaginary unit, n is the nth harmonic.

However, it is impossible to use the current signal which with zero-phase to construct the total current signal because

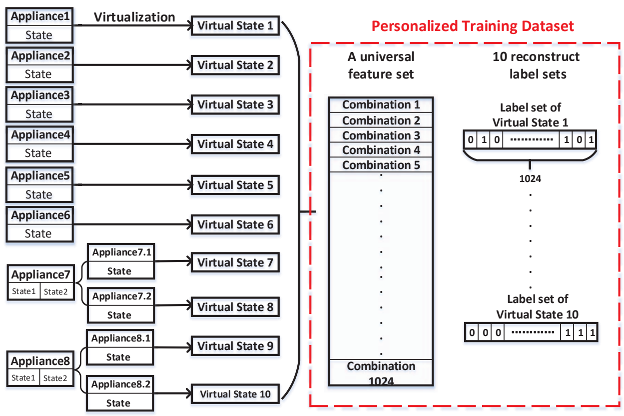

Figure 6. Construct Personalized Training Dataset

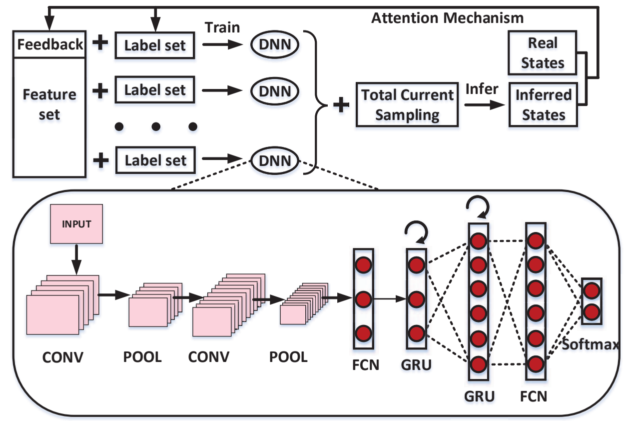

Figure 7. The training procedure and the structure of DNN

each appliance has a initial phase. So we should add the initial phase back to each appliance. The initial phase can be obtained from appliance profile. We take appliance A and B as an example to explain the procedure: The original current signal of A and B are: $f_A(t + pi + ph)$, $f_B(t + pi + ph)$. pi represents initial phase and ph represents the uncertain phase that produced because of the missing voltage baseline. As shown in Figure 4, the black line is voltage waveform which is the baseline, the red and blue lines are the current waveforms of fan and humidifier.

Then the phases are removed by using the method below, the current waveforms are:

$$f_A(t) = P(f_A(t)) \tag{4}$$

$$f_B(t) = P(f_B(t)) \tag{5}$$

We use function $P(x)$ to represent the procedure of removing phase. The initial phase should be added:

$$f_A(t + pi_A) = Q(f_A(t), pi_A) \tag{6}$$

$$f_B(t + pi_B) = Q(f_B(t), pi_B) \tag{7}$$

The function $Q(a, b)$ represents the procedure of adding phase. Parameter a is current signal and b is initial phase. The method of adding phase is the same as removing. Then the two current signal can added directly to construct a total current:

$$f_{A+B}(t + px) = f_A(t + pi_A) + f_B(t + pi_B) \tag{8}$$

Formula (8) may produce some phases px, so we should remove it again:

$$f_{A+B}(t) = P(f_{A+B}(t + px)) \tag{9}$$

So we extract the total current signal from the Appliance profile of A and B. MCSS collects the actual current signal when A and B are working: $G_{A,B}(t + px)$. It also need to remove phase:

$$G_{A,B}(t) = P(G_{A+B}(t + px)) \tag{10}$$

As shown in Figure 5, the black line is voltage waveform which is the baseline, the red line is current waveform of fan add humidifier ($G_{A,B}(t)$) and blue line is actual data.

C. Construct Personalized Training DataSet

Figure 6 illustrates the process of constructing the personalized training dataset. The personalized training dataset can be built up from LocalDB which stores all appliance profiles. For each appliance which has m ($m >= 1$, the state of OFF is not included) states, we regard it as m virtual states and each virtual state is an array representing of the current signal. For example, if the LocalDB includes 6 appliances each with 1 states and 2 appliances each with 2 states, we can get 10 virtual states. ($10 = 6 \times 1 + 2 \times 2$).

Then a universal feature set (i.e., a two dimensional matrix) can be generated from these virtual states. There are 2^{10} rows in the matrix and each row is a combination of virtual states. To get a training model for each virtual appliance, a corresponding and unique label set of 2^{10} Boolean values is constructed except for the feature set. Each value of label set indicates the virtual states, which is contained in the combination or not. So the personalized training dataset for a virtual state is composed of a universal feature set and a reconstruct label set. In this way, a very small amount of data can constitute a personalized training dataset, which can only depends on the appliance profiles owned by user. In addition, the feedback which is based on user behavior will also be added in the personalized training dataset.

D. The Design of Deep Neural Network

Recurrent Neural Networks (RNNs) are suitable for processing the input with time series like the current waveform.

Table I
TYPES OF APPLIANCE STATES

Types	Appliances	Virtual States
TS	electric fan	electric fan
	table lamp	table lamp
	air purifier	air purifier
	computer display1	computer display1
	computer display2	computer display2
FSM	humidifier	humidifier state1
		humidifier state2
CV	laptop computer	laptop computer state1
		laptop computer state2
		laptop computer state3

Table II
THE RESULT OF EVALUATION

Virtual State	P	R	Acc	F1
electric fan	0.995	0.987	0.991	0.991
table lamp	0.991	0.995	0.993	0.993
air purifier	0.984	0.917	0.951	0.949
computer display1	0.984	0.981	0.983	0.983
computer display2	0.952	0.942	0.947	0.947
humidifier state1	0.979	0.987	0.983	0.983
humidifier state2	0.839	0.787	0.817	0.812
laptop computer state1	0.977	0.980	0.979	0.979
laptop computer state2	0.997	0.974	0.969	0.985
laptop computer state3	0.965	0.978	0.969	0.969
All appliance states	0.938	0.918	0.929	0.928

Specifically, due to the requirement of online learning, it's reasonable to use Long Short-Term Memory (LSTM) to avoid the long-term dependency problem. LSTM has been widely used on a wide variety of sequence tasks including automatic speech recognition and machine translation. Since Gated Recurrent Unit (GRU) processes iterations faster than LSTM, so we finally choose GRU as our model. The training procedure and the structure of DNN is shown in Figure 7.

The original input of the model is a one dimensional array which contains current waveform of 10 cycles. We reshape it into a two-dimensional matrix. There are 10 rows in the matrix and each row is current waveform in a cycle. In this way, the matrix can present the relevance and locality not only in a cycle but also between cycles. Then we use CNN to capture features of the matrix. The model consists of two convolutional layers (CONV), two pooling layers (POOL), two GRU layers, two full connected layers (FCN) and a softmax classifier. The output is a boolean value. 1 and 0 represents appliance state ON and OFF, respectively.

E. Online Learning

The design of the online learning algorithm of IEHouse is based on Attention Mechanism. When users provide feedback, a comparison between the actual state and the inferred one will be made. If the state between actual and inferred are different, then the state will be regarded as a surprise. Therefore, the weight of input, including the state and current signal at that moment, will be set to higher than others and saved into the personalized training dataset. Then the model will be trained again using the new dataset. The number of training times is proportional to the frequency of feedback. With the increasing of usage, the training model will become stable and accurate.

V. EVALUATION

An lab experiment is setup to evaluate IEHouse. A personal computer is used as a Home Server and MCSS can be adhered in the power strip line which represents the main line of a house. We choose seven mixed typical appliances in the experiment. They can represent the three types of appliances: humidifier belongs to FSM, laptop computer belongs to CV and others are TS appliances. The types and their virtual states are shown in Table I.

A. Energy-awareness

Considering the application environment of a house, it is important to design a low energy system. MCSS, the battery-powered only component, is most sensitive to energy in IEHouse. The following three aspects address the low energy design of MCSS: 1) It adopts low-cost, low-power 802.15.4 chipset; 2) It applies BSI to decrease the number of samples and communications ; 3) It employs intermediate frequency sampling to capture the feature of appliance state. In our environment, the power of sampling sensor is about $0.3W$ and the energy consumption of per sampling is $1.27J$. If we apply the default BSI (8 hours are labeled as peak time and 16 hours are labeled as off-peak time in a day), then the number of sampling data is 576 ($8 \times 60 + 16 \times 6$) per day, 4032 per week. So the total energy consumption in a week is estimated as $5.12kJ$. Then a charge of $300kJ$ can be used for more than a year, which meets the need of low energy.

PowerBlade [23] is a state-of-the-art, low cost and low power AC plug-load meter that can measure real, reactive and apparent power. The power of PowerBlade is less than $0.18W$ and the energy consumption in a week can be estimated as $108.9kJ$. So the energy consumption of PowerBlade is 20x higher than that of MCSS.

B. Effectiveness

A typical classification metrics, Precision (P), Recall (R), Accuracy (Acc) and F-measure (F1), is used to evaluate the effectiveness of the algorithm [24] :

In this stage, the personalized training dataset is split so that 60% of the data is used for training, 20% of the data is used for validation and the remainder is used for test. A validation test is performed on the entire training data set. Every model is trained on the training dataset.

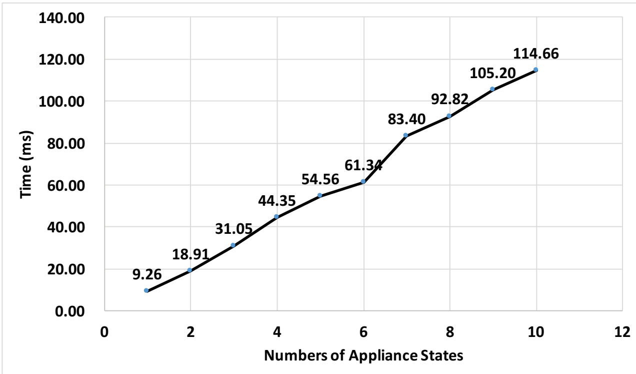

Figure 8. Inferring time of different numbers of appliances states

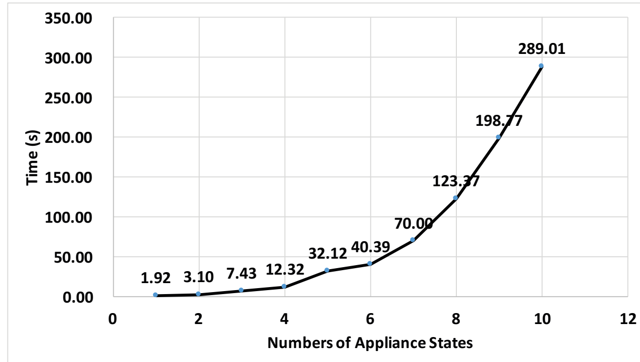

Figure 9. Average training time of different numbers of appliances states

Then, performance of the obtained model is estimated on the validation subset. This procedure will be repeated until a model with a best performance is trained. The training method may have a little different compared to the real scenario for it is used to evaluate the performance of algorithm only. Then we use the models to infer states on the test set. The result is shown in Table II.

The result shows that IEHouse performs good accuracy in recognizing not only different appliances, but also the same appliance with different operating states. The laptop computer has three states presents a high accuracy (e.g., 97.9% accuracy at most). It also shows the potential to identify different appliances with the same feature (e.g., computer display1 and computer display2, 98.3% accuracy at most). The accuracy of humidifier state2 is lower than others because the features of two states of humidifier are very similar.

The accuracy of different NILM algorithms is hard to compare because researches did not use uniform dataset. The accuracy of most algorithms is between 80% and 90%. Recently, [15] adapted three deep neural network architectures and drew a conclusion that all three models achieved better performance than traditional methods. The accuracy of three models are 91%,94% and 68%. But it is worth nothing that they can identify 6 appliance states while IEHouse can recognize 10 appliance states.

C. Scalability

In order to evaluate the scalability of the system, we conduct tests considering the training time on DNN models with different number of appliance states and the inference time. When a new appliance state is added, a new model will be built and trained. As shown in Figure 9, although the Personalized Training Dataset exhibits exponential growth, the average training time of a new state do not take exponential time in the experience. Meanwhile, with the users' feedback, the models of existing state will obtain more information of the users' behavior. The growth of training time will be slower. Thus, IEHouse can be expanded to more appliance states. When a new appliance state is added, the inference time will be longer because the number of models increases. As shown in Figure 8, the relationship between inference time and the number of appliance states is linear. It cost about 10 ms to infer an appliance state when there is only an appliance state. If there are 30 appliance states in a house, then the inference time increases to 300ms, which is acceptable for IEHouse and meet the real-time demands. According to the low training and inference time when adding a new appliance state, IEHouse illustrates its scalability.

VI. CONCLUSION

We present IEHouse, an efficient NILM system to recognize appliance states in this paper. The design of IEHouse is based on the observation that personalized behavior data can help for recognizing the states of appliances. By constructing the training dataset dynamically based on appliance profiles, it saves the time cost of tagging samples. By using DNN models, IEHouse can accurately discern different appliances or the different states of the same appliance. By evaluating the system, we show that IEHouse has significant advantages in the following aspects.

- Energy-awareness: The energy consumption of the sampling sensor is 1/20 of that of a similar product.
- Effectiveness: IEHouse achieves 92.9% accuracy averagely for recognizing 10 appliance states, which is higher than that of most algorithms.
- Scalability: The inference time increases proportionally to the number of appliance states and that of one appliance state is approximately 10 ms.

VII. ACKNOWLEDGEMENTS

We are greatly indebted to Chen Feng and Fan Liang for commenting and modifying on earlier versions of this paper, to Jingjie Liu and Linyang Wu for their previous work. This work was partially supported by the NSF of China (61532016) and the MOST (2016YFB1000200).

REFERENCES

[1] N. Batra, "Non intrusive load monitoring: Systems, metrics and use cases," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '15. New York, NY, USA: ACM, 2015, pp. 501–502. [Online]. Available: http://doi.acm.org/10.1145/2809695.2822523

[2] N. Batra, A. Singh, and K. Whitehouse, "Gemello: Creating a detailed energy breakdown from just the monthly electricity bill," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 431–440. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939735

[3] A. H. Kazmi, M. J. O'Grady, and G. M. O'Hare, "Energy management in the smart home," in *Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC)*. IEEE, 2013, pp. 480–486.

[4] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[5] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[6] J. Liu, "The function sensing framework for power consumption behaviors in households," Ph.D. dissertation, University of Chinese Academy of Sciences, 2015.

[7] X. Jiang, S. Dawson-Haggerty, P. Dutta, and D. Culler, "Design and implementation of a high-fidelity ac metering network," in *Information Processing in Sensor Networks, 2009. IPSN 2009. International Conference on*. IEEE, 2009, pp. 253–264.

[8] Y. Kim, T. Schmid, Z. M. Charbiwala, and M. B. Srivastava, "Viridiscope: design and implementation of a fine grained power monitoring system for homes," in *Proceedings of the 11th international conference on Ubiquitous computing*. ACM, 2009, pp. 245–254.

[9] Z. C. Taysi, M. A. Guvensan, and T. Melodia, "Tinyears: spying on house appliances with audio sensor nodes," in *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*. ACM, 2010, pp. 31–36.

[10] B.-J. Ho, H.-L. C. Kao, N.-C. Chen, C.-W. You, H.-H. Chu, and M.-S. Chen, "Heatprobe: a thermal-based power meter for accounting disaggregated electricity usage," in *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 2011, pp. 55–64.

[11] S. Gupta, M. S. Reynolds, and S. N. Patel, "Electrisense: single-point sensing using emi for electrical event detection and classification in the home," in *Proceedings of the 12th ACM international conference on Ubiquitous computing*. ACM, 2010, pp. 139–148.

[12] M. T. I. Aumi, S. Gupta, C. Pickett, M. Reynolds, and S. Patel, "A self-calibrating approach to whole-home contactless power consumption sensing," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2014, pp. 361–371.

[13] J. Z. Kolter and M. J. Johnson, "Redd: A public data set for energy disaggregation research," in *Workshop on Data Mining Applications in Sustainability (SIGKDD), San Diego, CA*, vol. 25. Citeseer, 2011, pp. 59–62.

[14] J. Z. Kolter, S. Batra, and A. Y. Ng, "Energy disaggregation via discriminative sparse coding," in *Advances in Neural Information Processing Systems*, 2010, pp. 1153–1161.

[15] J. Kelly and W. Knottenbelt, "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, 2015, pp. 55–64.

[16] L. Mauch and B. Yang, "A novel dnn-hmm-based approach for extracting single loads from aggregate power signals," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 2384–2388.

[17] H. Goncalves, A. Ocneanu, M. Berges, and R. H. Fan, "Unsupervised disaggregation of appliances using aggregated consumption data," in *The 1st KDD Workshop on Data Mining Applications in Sustainability (SustKDD)*, 2011.

[18] H. Kim, M. Marwah, M. F. Arlitt, G. Lyon, and J. Han, "Unsupervised disaggregation of low frequency power measurements." in *SDM*, vol. 11. SIAM, 2011, pp. 747–758.

[19] J. Z. Kolter and T. S. Jaakkola, "Approximate inference in additive factorial hmms with application to energy disaggregation." in *AISTATS*, vol. 22, 2012, pp. 1472–1482.

[20] M. J. Johnson and A. S. Willsky, "Bayesian nonparametric hidden semi-markov models," *Journal of Machine Learning Research*, vol. 14, no. Feb, pp. 673–701, 2013.

[21] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "An unsupervised training method for non-intrusive appliance load monitoring," *Artificial Intelligence*, vol. 217, pp. 1–19, 2014.

[22] A. Adabi, P. Manovi, and P. Mantey, "Seads: A modifiable platform for real time monitoring of residential appliance energy consumption," in *Green Computing Conference and Sustainable Computing Conference (IGSC), 2015 Sixth International*. IEEE, 2015, pp. 1–4.

[23] S. DeBruin, B. Ghena, Y.-S. Kuo, and P. Dutta, "Powerblade: A low-profile, true-power, plug-through energy meter," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2015, pp. 17–29.

[24] P. P. M. do Nascimento, "Applications of deep learning techniques on nilm," Ph.D. dissertation, Universidade Federal do Rio de Janeiro, 2016.