PowerAnalyzer: An Energy-Aware Power Monitor System Aiming at Energy-Saving

Yifan Wang*‡, Xingzhou Zhang*‡, Lu Chao*‡, Lang Wu*‡ and Xiaohui Peng†
*SKL Computer Architecture, ICT, CAS, Beijing, China
†Research Center for Ubiquitous Computing Systems, ICT, CAS, Beijing, China
‡University of Chinese Academy of Sciences, Beijing, China
{wangyifan2014, zhangxingzhou, chaolu, wulang and pengxiaohui}@ict.ac.cn

***Abstract*—To save the electrical energy in a household, it is essential to monitor where and how the power is consumed. To maximize the efficiency of energy conservation, it is necessary to make the running power low in the power monitor system, which the tradition systems pay less attention to. This paper presents PowerAnalyzer, an energy-aware system for monitoring running states and power of each household appliance plugged into power line from a single point detection. PowerAnalyzer takes steady-state current waveforms as the appliances signature, and uses the deep neural network (DNN) models to infer the running states and running power of household appliances. We focus on the energy consumption of PowerAnalyzer itself. The energy efficiency of PowerAnalyzer is optimized from these aspects: Using dynamic time intervals to collect electric data, replacing a cloud server with an edge node to process data, and transmitting differential data over a low power wireless protocol. The evaluation results show that PowerAnalyzer offers 3.45% average power metering error and 98.38% average accuracy of inferring running states of appliances. PowerAnalyzer draws less than $247mW$ static power and $304mW$ peak power.**

***Index Terms*—Power Monitor System, Energy-Aware, Running State Inference, Energy Efficiency Optimization, Edge Node**

I. INTRODUCTION

Electricity consumption in China has increased from $2170TWh$ to $5920TWh$ with annual growth rate of 10.55% between 2006 and 2016 [1]. The household consumed about 13% of China electricity energy, which equals to the total consumption of primary and tertiary industry [2]. Thus increasingly importance has been attached to household energy conservation. Compared with the coarse-grained knowledge about total energy consumption, a fine-grained information about where and how the energy is consumed is more helpful in guiding household energy saving [3]. Therefore, it is significant to deploy monitor devices in a home to track every household appliance states and energy consumption. With the rapid development of internet of things (IoT) [4], a large number of manufacturers design Smart Plug to meter household appliances power with high-fidelity. We will need N smart plugs if we want to monitor N appliances [5], which leads the high complexity of deployment and the high cost of maintenance. To solve this problem, Non-Intrusive Load Monitoring (NILM) [6], [7] method proposes some algorithms and frameworks to track the running states of N appliances by deploying only 1 or m smart meters, where $m \ll N$. The state-of-the-art research on NILM focuses the methodologies of detecting the specific load signatures of an individual appliance by single point monitor [3], [8], [9]. However, the methods of NILM usually ignore the energy consumption of themselves which may be greater than the saved power consumption, resulting in the meaningless of the power meter network deployment in a household.

In order to maximize energy saving, we propose an energy-aware power monitor system named PowerAnalyzer, which uses single point detection to infer the running states and calculates the running power of each appliance plugged into the power line. The edge computing [10] conception is incorporated into the NILM method, which concerns about the interrelationship of the data processing position in the network and the running power, helping to reduce the energy consumption of PowerAnalyzer. PowerAnalyzer consists of two main components: Data Collector (DC) and Data Processor (DP). DC uses low sampling frequency and dynamic time intervals to collect current waveforms of the power line to ensure the low energy consumption in collecting data. DP is implemented in the household gateway, where is closer to data producer (DC) than cloud server, which leads to low energy consumption in data processing. Furthermore, DC and DP transfer data using 802.15.4/6LowPAN/CoAP [11] protocol stack, which consumes less power than 802.11/HTTP stack. Without losing the power metering accuracy, PowerAnalyzer offers lower running power. According to the evaluation results, PowerAnalyzer has an average power metering error of 3.45% over the range of $1W$ to $1400W$ household appliances, matching current power meter products. Meanwhile, PowerAnalyzer consumes only $304mW$ when performing the inference operation, keeping 98.38% average accuracy over 7 appliances.

Targeting at presenting a complete design and implementation of PowerAnalyzer, we make the following contributions in the paper.

- A non-intrusive power monitor system named PowerAnalyzer has been presented. PowerAnalyzer takes the steady-state current waveforms as the signature of appliances and uses DNN models to infer the household appliances running states and power. Compared with the traditional power meter, this design reduces the cost and complexity of deploying the power monitor system.
- An energy-aware implementation of PowerAnalyzer has been proposed. The energy efficiency of PowerAnalyzer

is optimized from three aspects: data collecting, data processing and communicating. PowerAnalyzer provides high metering and inference accuracy under low running power, which will be more practical than other power monitor systems.

The rest of this paper is organized as follows. In section II, a survey of state-of-the-art methods on power monitor system has been provided. The basic function design and energy-aware implementation details of PowerAnalyzer are presented in section III and section IV respectively. The evaluation results of metering and inferring accuracy and power consumption has illustrated in section V and we conclude the paper in section VI.

II. Related Work

Numerous power metering systems have been designed, which aim at meeting the demand of energy monitoring in a residential or commercial building. These systems can be divided into two categories: IoT oriented and algorithm oriented. The former focuses on the implementation of meter hardware, and the latter focuses on reducing the meter number by algorithms on the condition of assuring accuracy to achieve non-intrusive appliances monitoring.

A. IoT Oriented System

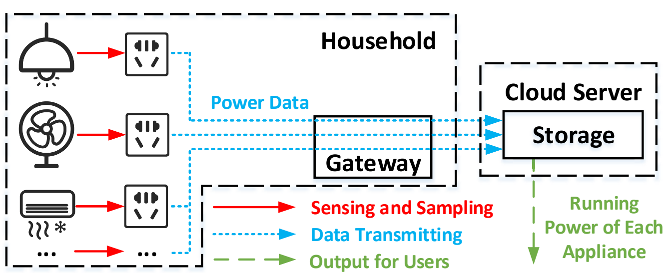


Fig. 1. IoT Oriented System of Energy Monitoring

IoT oriented system can also be called sensor oriented system, which focuses on power meter performance and usability. Several researches on the AC-Load power meter and commercial smart plug can be classified as the IoT oriented system, whose structure is presented in Fig. 1. These researches optimize metering system in different aspects. ACme [5] focuses on the high-fidelity of power metering, and designs a sensor network to monitor coarse-grained energy consumption in a building. Monjolo [12] uses an energy-harvesting method to power itself, and has the lowest static power ($4mW$), but it cannot report the true power of appliances. Gemini [13] is a non-invasive power meter using separated hardware to sense current and voltage waveforms. PowerBlade [14] concerns about the volume of the meter and presents the design of energy meter hardware occupying two-dimensional volume with high metering accuracy. There are several other commercial smart plugs [15], [16], [17], [18] paying more attention to the appearance and interactive behavior of the products. However, if N appliances should be monitored at a household, then N power meter should be used with IoT oriented system, which will result in high deployment complexity and cost. If we reduce the number of power meters, it will cause coarse and inconsistent grained monitoring problem and power consumption of each household appliance cannot be reported respectively.

B. Algorithm Oriented System

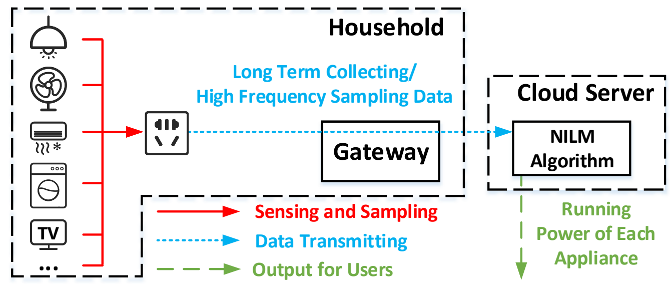
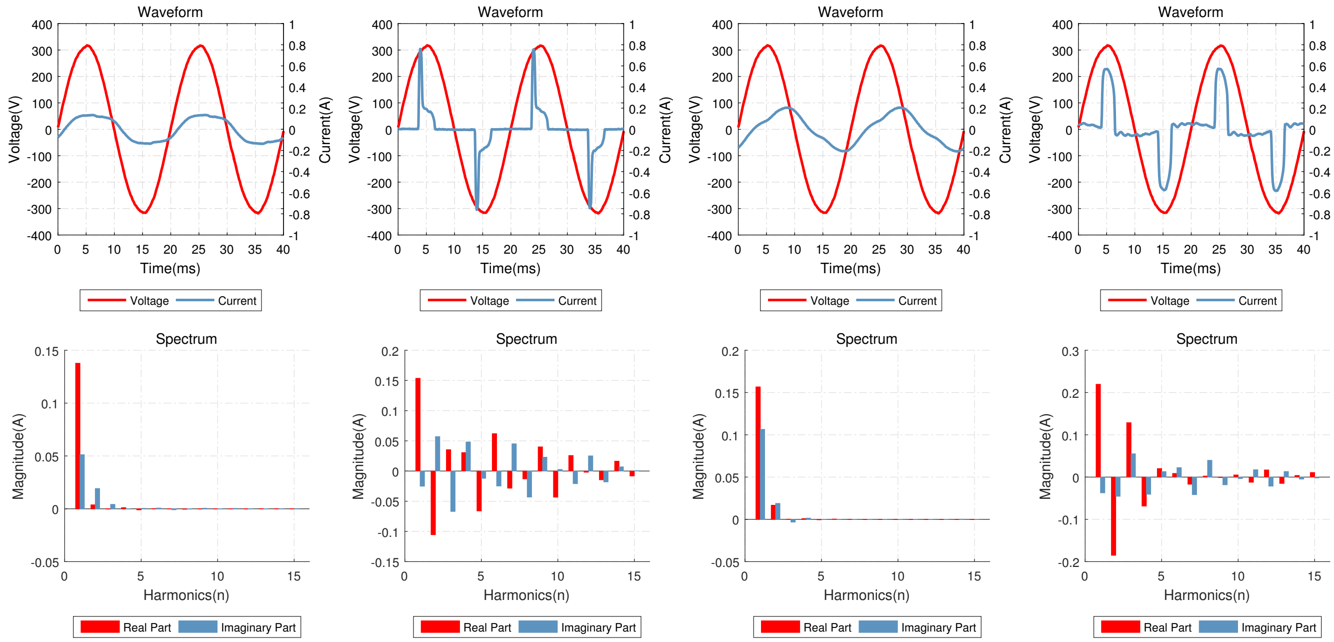


Fig. 2. Algorithm Oriented System of Energy Monitoring

NILM system is a typical algorithm oriented system. Several NILM approaches are based on appliance signature extraction. Suzuki [19] models the single-cycle current waveforms of appliances, and infers appliances running state via solving integer programming problems. ElectriSense [20] presents a new solution to detect and classify the usage of appliances in a house from a single point of sensing the EMI signal which is generated during power supplies operation. Liu [9] proposes a function sensing framework based real-time current waveforms sensing for every appliance. The framework disaggregates total current into the currents of all appliances in a household, and acquires their on-off states. Another kind of NILM approach focuses on state transition of appliances. Hidden Markov model (HMM) is the base model of each appliance state inference by energy disaggregation. Kim [21] uses multi-dimensional prior knowledge and Factorial HMM (FHMM) to reduce the dependence on the training set. Liu [9] proposes Event-driven HMM (EHMM) and efficient decode algorithm improving the accuracy of on-off state sensing for each appliance. Algorithm oriented can also be called cloud server oriented, which structure is illustrated in Fig. 2. High-frequency signal feature extraction and long-term prior knowledge analysis rely on the performance of cloud server, which increases power in their systems and reduces the practicability of these systems.

III. Design

Based on the survey of IoT oriented system and algorithm oriented system, reducing sensor numbers to get low deployment costs is needed in PowerAnalyzer system. Meanwhile, the plug and play feature of IoT oriented system is necessary. PowerAnalyzer needs steady-state electrical data to infer the running states and power of N appliances plugged into the power line. It is the basic feature and function that PowerAnalyzer has to provide. The design scheme of PowerAnalyzer which achieving this basic function is presented in this section.



(a) 20W Desk Lamp (b) Air Humidifier (c) AC Fan (d) Display

Fig. 3. The Current Waveforms and Spectrums of Four Household Appliances.

A. Design Overview

PowerAnalyzer consists of Data Collector (DC) and Data Processor (DP). DC focuses on collecting steady-state electrical data of power line. DP infers the running state (e.g., switch on or switch off, max level or standby) and power consumption of each appliance according to the data acquired from DC which contains the characteristics of appliances. PowerAnalyzer no longer needs to learn users behavior, provides the plug and play feature. We call the characteristics contained in electrical data the Appliances Signature. The inference of running state of appliances can be regarded as a classification problem.

B. Appliances Signature

As Fig. 3 shows, the voltage waveforms of different household appliances are consistently sinusoidal, but current waveforms vary wildly with a certain degree of identification. Resistive loads, like the lamp, have sinusoidal current waveforms, with zero phase-shifted to voltage waveform. There is usually a phase-shifted phenomenon between current and voltage waveform of inductive loads, as the AC fan shows. However, for most household appliances, the current waveforms are not sinusoidal, but triangular or other shapes. The difference of amplitude, phase, and pattern in time domain matches the distribution characteristics of the real and imaginary parts of the fundamental and nth harmonic in the frequency domain.

The running current waveforms in the time domain of appliances is identifiable, so we take it as the signature of appliances. Especially, the relative phase bias between appliances current and voltage waveforms is a significant feature. However, collecting current waveforms at the arbitrary time will bring a new phase bias. The current waveforms of an appliance is shown in Equation (1).

$$f(t) = \sum_{n=1}^{\infty} A_n cos(n\omega_0 t + \varphi_n) \tag{1}$$

The current waveforms collected at the arbitrary time is shown in Equation (2).

$$f(t+\Delta t) = \sum_{n=1}^{\infty} A_n cos(n\omega_0 t + \varphi_n + \Delta\varphi_n) \tag{2}$$

Here,

$$\Delta\varphi_n = n\omega_0 \Delta t \tag{3}$$

when $\Delta t = m \cdot 2\pi/\omega_0$, $f(t) \neq f(t+\Delta t)$, we cannot guarantee the phase consistency of the appliances signature. Thus, the current collection needs a unified start flag to make sure current waveforms data contains the consistent phase information. The voltage of each power branch is consistent [13]. Therefore, regarding the peak or zero point of the voltage as the start flag of the collection will guarantee all current waveforms have the unified phase reference information, which provides a complete data for DP.

C. Appliances Real-Time States Inference

To achieve the plug and play characteristic, we need to do real-time classification based on current waveforms of power line. We use the simplest case to describe the inferring process. If there are N household appliances with two states (switch on/off), the current waveforms of the power line will have 2^N kinds of patterns. Then we use a supervising method to train

N classifiers, with 2^N patterns for training data and the label corresponds switch on/off state. We can get the running states of each appliance by inputting the current waveforms acquired by DC into N classifiers in real time.

To ensure real-time inferring, we use DNN as a classifier. We design the structure of DNN according to the characteristics of the current waveforms, as shown in Fig. 4. The details of building training data sets and iterative training process were introduced in our prior work IEHouse [22]. Here, we give a brief description of DNN structure. We first use convolutional layer and pool layer to extract features of training data. We leverage the Gated Recurrent Unit (GRU) [23] to extract time correlation characteristics after CNN, and then output classification results through the Softmax classifier. DNN models of each appliance are stored in the DP. DNN models take the current waveforms data as the input, and output the appliances running states. The infer process is shown in Fig. 4.

IV. Implementation

Unlike the systems described in section II, PowerAnalyzer is an energy-efficiency oriented system. While ensuring basic function, we also focus on energy consumption of PowerAnalyzer. This section introduces the energy-aware implementation details of PowerAnalyzer.

A. System Overview

To ensure real-time data processing and to reduce the processing power consumption, an edge node is used to replace the a cloud server (algorithm oriented system) to process data, as shown in Fig. 5. In our implementation, the DC transmits real-time current data to the DP to process and DP pulls the trained DNN model from the cloud server at the initial stage of the system. DP is located at the edge of data transmission link between DC and cloud server [10], which means DP is the data process center at the edge of the network. Aiming at this edge computing system design, we will optimize energy-efficiency from the implementation of DC, DP and the communication method to achieve the energy-aware of PowerAnalyzer.

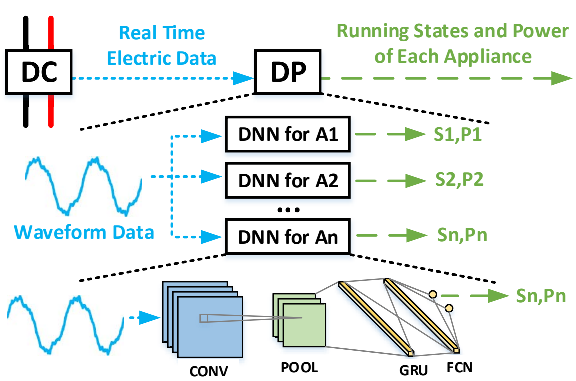


Fig. 4. Process of Appliances States Inference [22]

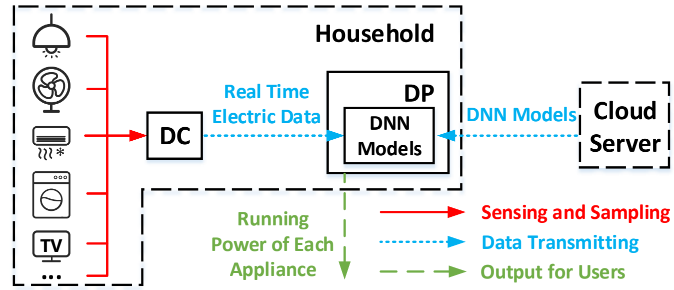


Fig. 5. Energy-Efficiency Oriented System of Energy Monitoring

B. Data Collector

Structure of DC is illustrated in Fig. 6. Voltage transformer and current transformer provides collectable voltage and current analog signal. DC uses an energy metering IC to collect electric data which is the most accurate approach under the same power to provide current waveforms and power data. In our implementation, we choose ADE7763 as energy metering IC, which can provide the zero-crossing interrupt vector of voltage channel. PowerAnalyzer regards the rising of zero-crossing interrupt as the start flag to sample current. Using this method, it can ensure that the sampling current data has the unified phase reference.

PowerAnalyzer infers the running states of the appliances using steady-state current waveforms, so we can reduce the energy consumption in data collection by reducing the sampling frequency and increasing the data collecting interval. According to our implementation, SoC in DC (will describe later) can provide $2kB$ RAM for caching the sampling data, $7ksps$ sampling rate to ensure sampling enough cycles of current data to extract the appliances signatures along with a low running power.

In the real scene, the states of household appliances usually are not change frequently at most of the time (e.g. weekday and midnight). Therefore, we use the dynamic time intervals to collect electrical data. We set the upper and lower limits of the collection interval. When the system is initialized, the collection interval is the lower limit. If the appliances are in a steady state for a long time, the data collection interval will increase gradually and reach the upper limit after a period of

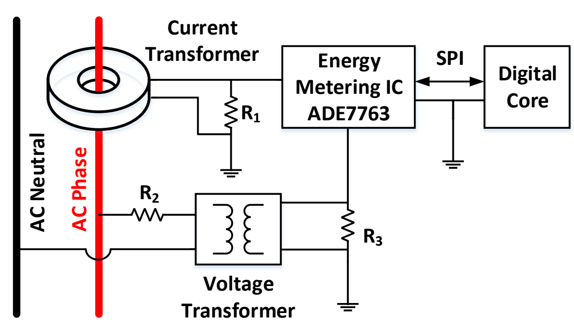
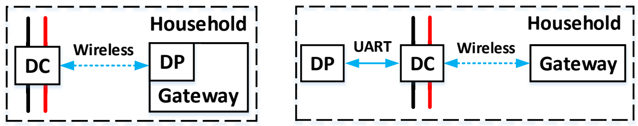


Fig. 6. Structure of Data Collector.



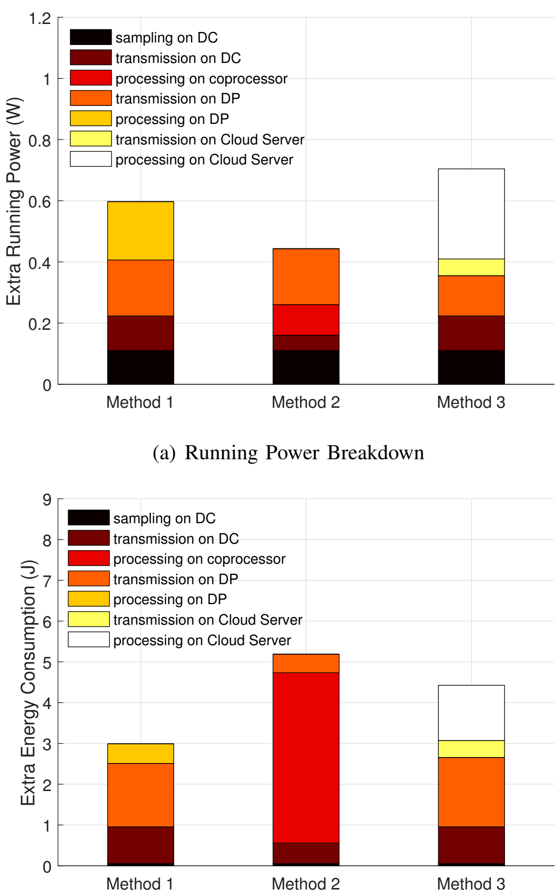
(a) Structure of Method 1 (b) Structure of Method 2

Fig. 7. Different Position of DP Implementation.

time. When the states of household appliances are unstable, data collection interval directly goes back to the lower limit, to ensure that the change of appliances running states could be captured by PowerAnalyzer. We measure power consumption for DC both with fixed and dynamic collecting interval. It reduces power consumption by 5.05% by using dynamic collecting interval under the same measurement accuracy. The detailed evaluation of dynamic collecting interval will be presented in section V.

C. Data Processor

The Data Processor takes the electric data collected by DC as the input of the states classification model for each appliance, and produces the running states and power of



(a) Running Power Breakdown

(b) Energy Consumption Breakdown

Fig. 8. Running Power and Energy Consumption Breakdown of Three Methods.

TABLE I
DIGITAL CORE COMPARISON

Digital Core	Static Current	RX/Tx Current	Cost	Size
CC2650	$1\mu A$	$5.9/6.1mA$	\$3.3	$49mm^2$
MSP430+nRF51822	$8.9\mu A$	$9.7/6.3mA$	\$3.18	$52mm^2$
MSP430+CC2420	$26.3\mu A$	$18.8/17.4mA$	\$5.47	$65mm^2$

this appliance. In the algorithm oriented system, DP is implemented in the cloud server, as shown in Fig. 2. In our implementation, we move DP to the household range (i.e., the edge of the home network). DP can be implemented in two different positions in the system, as shown in Fig. 7. Different implementations of DP will bring different performance and running power.

We take Raspberry Pi 3 Model B (RPi3) as a household gateway. In method 1, DP is implemented on RPi3. RPi3 obtains current data collected by DC through the wireless protocol, and the trained DNN models are implemented by TensorFlow framework. In method 2, we adopt the low power consumption MCU stm32L152 [24] as the coprocessor to run DNN inference models. DC send current data to stm32L152 via UART. Gateway obtains running states and power of appliances from DP via the wireless protocol. The running power and energy consumption of the two methods is shown in Fig. 8. To prove the efficiency of the edge computing method, the algorithm oriented system is regarded as the baseline (method 3). This method has been conducted on an Apple® MacBook Pro. It features an Intel® Core™i5 3.1GHz processor and 8GB main memory.

Fig. 8 shows the running power and energy consumption breakdown of these three methods while performing inference operation. In method 3, as the data should be transmitted to the cloud server for calculating, the power consumption in data transmitting is $2.12J$, 1.36x more than the method 1. In method 2, DNN inference model is running in a coprocessor. Although data transmission energy consumption is decreased, due to the limitation of computing resources, coprocessor takes longer time to complete the inference operation, leading to the $4.18J$ energy consumption consumed by method 2 in the data processing, 4.56x more than the other two methods. In conclusion, the method 1 has enough computing performance to ensure less processing time and processing power. Meanwhile, method 1 will not bring additional transmission energy consumption compared with method 3. We choose method 1 as the implementation of Data Processor.

D. Communication

As shown in Fig. 8, the data transmission takes approximately half of the time in once inference operation, so the data transmission between DC and DP should be energy-efficient optimized. According to the works in [25], the 6LowPAN protocol based on 802.15.4 has lower transmission power compared to Wi-Fi and has a longer transmission distance than the BLE protocol. In our implementation, PowerAnalyzer takes the ultra-low power wireless SoC CC2650 [26] as the radio module. Compared with the radio module selected in

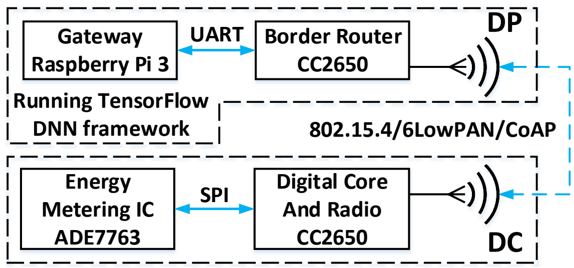


Fig. 9. Communication Mode of PowerAnalyzer.

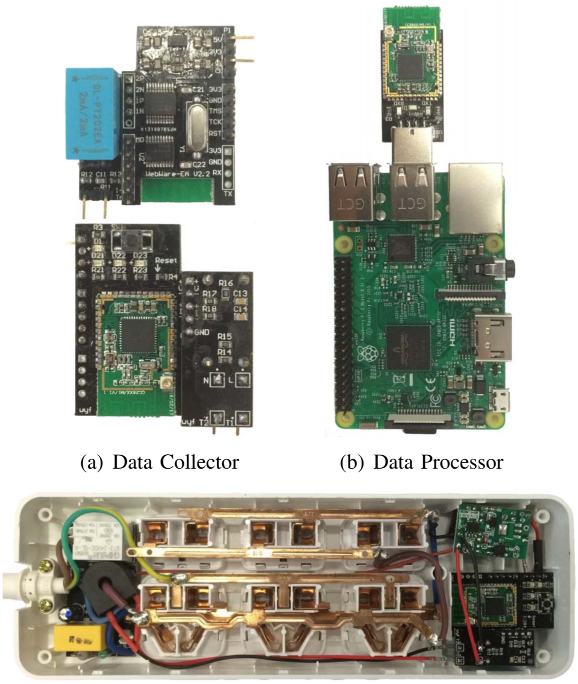
other work, the CC2650 has lower static and running power as shown in Table I.

A CC2650 chip is used as controller and radio in DC. CC2650 obtains electric data collected by ADE7763 through the SPI bus and transmit the data to DP. Another CC2650 chip is used as border router on DP to communicate with DC and forward the data to RPi3 via UART bus. To provide an easy way to view data, we embed the Contiki operating system in CC2650, which can satisfy the limited processing power and memory and support 6LowPAN/RPL/CoAP protocol. DC provides different RESTful URL for each electric data and requests data through GET/POST. This implementation allows users to access data directly over the URL without relying on specific hardware and software. The communication mode of the PowerAnalyzer is shown in Fig. 9.

Electric data is filled in the request data section of the CoAP layer for transmission. According to our measurement, $1kB$ of data requires $0.29J$ to transmit. For current waveforms data, we use the difference transfer methods to reduce the transmission bytes. The average transmission power of origin data transmission is $2.46J$. Due to the use of differential data transmission, the average transmission power consumption decreased to $2.05J$, which is 83.57% of previous one.

V. Evaluation

We evaluate the measurement accuracy of PowerAnalyzer in this section. The evaluation results contain the accuracy of reporting appliances running power and the accuracy of inferring N appliances running states. We also measure the energy consumption of PowerAnalyzer when reporting power and inferring the running states of appliances. In our evaluation scenario, we install DC in a power strip. The Fig. 10 shows the PowerAnalyzer hardware implementation and the installation.



(a) Data Collector (b) Data Processor

(c) Installation in a Power Strip

Fig. 10. PowerAnalyzer Hardware Implementation and Installation.

A. Accuracy

We divide evaluation the accuracy of reporting N appliances running power into two steps. First, we evaluate the metering accuracy of reporting appliances running power. Second, we evaluate the inference accuracy of N appliances running states.

- Metering Accuracy

To evaluate the power metering accuracy of PowerAnalyzer, we take some common household appliances as the benchmark. We utilize PowerAnalyzer and Xiaomi smart power strip [18] to measure the power of each appliance respectively. At the same time we use the measurements from professional AC and DC power analyzer AWE1611 [27] as our ground truth. The comparison results are shown in Table II.

According to the results, the power of household appliances range from $1W$ to $1400W$, the scope of power factor is vary from 0.39 to 0.95. The power measuring maximum error and average error of PowerAnalyzer are 5.21% and 3.45%, while the average error of Xiaomi smart power strip is 5.57%. It means that the power metering range and accuracy of PowerAnalyzer meet the requirement of commercial products.

- Inference Accuracy

To evaluate the inference accuracy of N appliances running states, we perform a full test for several appliances. The household appliances we used are marked An in Table II. We compare the inference accuracy of PowerAnalyzer with prior works: NN Method [28] and ElectriSense [20]. The comparison is shown in Fig. 11. The accuracy of PowerAnalyzer can

TABLE II
Metering Accuracy for Household Appliances

Appliance	PF	Power	XM Error	PA Error
Electric Cooker	1.00	355.43W	-0.43W (0.12%)	6.11W (1.72%)
Hair Drier (H)	0.99	1423.58W	3.58W (0.25%)	6.42W (0.45%)
Hair Drier (L)	0.99	733.27W	1.73W (0.24%)	4.29W (0.58%)
20W lamp (A5)	0.94	21.60W	-0.60W (2.77%)	0.89W (4.11%)
40W Display (A4)	0.64	35.24W	-1.24W (3.52%)	0.94W (2.67%)
20W Display (A6)	0.59	16.12W	-0.42W (2.61%)	-0.99W (6.14%)
Humidifier (A2)	0.57	22.86W	-0.86W (3.78%)	0.38W (1.65)%
AC Fan (A1)	0.56	16.84W	-0.84W (4.96%)	0.87W (5.21%)
Air Purifier (H) (A7)	0.56	30.54W	-0.54W (1.78%)	0.61W (1.31%)
Router (A3)	0.41	1.06W	0.34W (0.32%)	0.57W (4.69%)
Air Purifier (L)	0.39	4.59W	-0.28W (4.09%)	0.09W (2.00%)

achieve 100% at the number of appliances under 6. And the accuracy drops to 98.38%, when the number rises to 7. And the average accuracy is 85.82% and 96.80% for NN Method and ElectriSense respectively over $4-10$ household appliances. So we can conclude that PowerAnalyzer reaches same inference accuracy level with prior works.

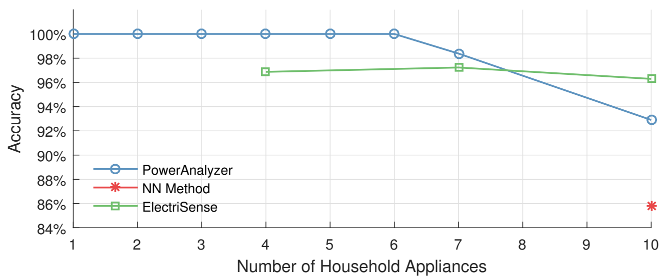


Fig. 11. Comparison of Inference Accuracy.

B. Power and Energy Consumption

- Runing Power

Table III presents the comparison of running power between PowerAnalyzer and other similar works. Sense [29] is a similar product which using single point detection to acquire all appliances running power. The running power of PowerAnalyzer is reduced by 95.06% compared with Sense. Comparing with other power meters, the static power of PowerAnalyzer is not the lowest. The reason is that a part of the static power is required in DP of PowerAnalyzer to execute inference operation. The traditional power meter is used to collect the total power on the socket, which is same as the DC function, while the static power of the DC is $110mW$, near to the lowest static power ($100mW$) of the power meters.

Deployment power in Fig. 12 means the running power of the metering devices when monitoring 7 household appliances. As we described above, we will need N power meters if we want to monitor N appliances. The total static power of the power meter products increases with the number of monitored appliances linearly. We do not need more PowerAnalyzer to monitor 7 or fewer household appliances, which makes the PowerAnalyzer has lower deployment power. Static power drops 93.06% comparing with other power meters when monitoring 7 household appliances, and the peak power drops 91.46%. We can conclude that PowerAnalyzer has low running power when deploying a power monitor system.

- Energy Consumption

Running power will reach the peak power $304mW$ when the PowerAnalyzer is executing inference operation. Due to the introduction of dynamic collection intervals, as the description in section IV, PowerAnalyzer is not always performing in the inference operation. When PowerAnalyzer is idle, running power will remain in static power $247mW$. The energy consumption of PowerAnalyzer is greatly related to the data collection interval, we cannot calculate it directly from static power. We measured the actual daily energy consumption

TABLE III
COMPARISON OF VARIOUS METERING DEVICE.

Metering Device	Data output	Standard Power	Deployment Power
XiaoMi Power Strip [18]	Wi-Fi	900mW	6.3W
UNI [17]	LCD	600mW	4.2W
ACme-A [5]	802.15.14	1000mW	7W
ACme-B [5]	802.15.14	100mW	700mW
PowerBlade [14]	BLE	128mW	896mW
Sense [29]	Wi-Fi&BLE	5W	5W
PowerAnalyzer (this work)	802.15.14	247mW	247mW
PowerAnalyzer (peak)		304mW	304mW
PowerAnalyzer (DC)		110mW	110mW
PowerAnalyzer (DP)		137mW	137mW

of PowerAnalyzer in a weekday and a weekend-day. The results are shown in Fig. 12. The daily energy consumption of PowerAnalyzer using fixed collecting interval is regarded as the baseline.

Based on the measurement results, the monthly energy consumption of PowerAnalyzer adopting dynamic collecting interval is $618.28kJ = 0.17kWh$, and in fixed collecting interval case is $654.03kJ = 0.18kWh$, which is 1.05x higher. And the monthly energy consumption of Xiaomi smart power strip will reach $3.89kWh$. According to the research in work [30], the monthly energy consumption of a household in China is $127.89kWh$. And if the monitoring results of energy consumption can be direct fed back to the users, approximately 10% energy will be saved [31]. E_s and E_c denote monthly energy saved and monthly energy consumed by one monitored device, n is the number of monitored devices to deploy monitor system, the energy-saving efficiency η can be calculated by Equation (4).

$$\eta = (E_s - E_c \times n)/E_s \tag{4}$$

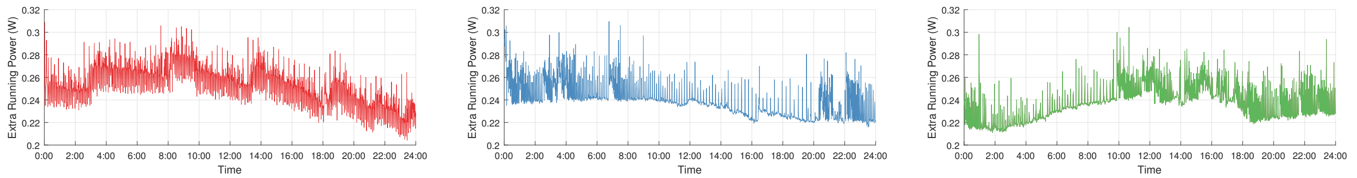
According to the previous evaluation results, PowerAnalyzer monitors the running states of 7 appliances with the accuracy of 98.38%. For a household, power monitor system is deployed using PowerAnalyzer, Xiaomi smart power strip, and Sense respectively, the energy-saving efficiency is shown in Table IV. We assume that the number of appliances in a household is 20. Due to the PowerAnalyzer's energy-aware feature, its own monthly energy consumption has been reduced by 95.39% comparing to the other two products. While the energy-saving efficiency rises to 95.95%.

TABLE IV
COMPARISON OF ENERGY CONSUMPTION AND ENERGY-SAVING EFFICIENCY

Metering Device	Power	n	E_c	$E_s - E_c \times n$	η
Sense	5W	1	3.6kWh	9.19kWh	71.85%
XiaoMi	0.9W	20	0.65kWh	-0.17kWh	-1.33%
PowerAnalyzer	247mW	3	0.17kWh	12.27kWh	95.95%

VI. CONCLUSION

The state-of-the-art works on power monitor systems have limited ability to keep their own low energy consumption, which reduces the efficiency of energy saving. To address



(a) Fixed Collecting Interval. (b) Dynamic Collecting Interval(A Weekay) (c) Dynamic Collecting Interval(A Weekend-day).

Fig. 12. Running Power of PowerAnalyzer in Different Scenes.

these challenges, we present an energy-aware power monitor system, PowerAnalyzer. PowerAnalyzer consists of Data Collector (DC) and Data Processor (DP). DC collects the real-time current data of power line and transmits the data to DP. DP takes the current waveforms data as the input of DNN classifiers, and output the appliances running stats and power. According to the evaluation results, PowerAnalyzer can provide 2.89% error rate of power metering and 98.38% average accuracy of appliances running states inference over 7 appliances. Comparing with similar works, the running power of PowerAnalyzer drops 93.06%, and monthly energy consumption is $0.17kWh$, reducing 95.39% than other metering devices, making the power saving efficiency increased to 95.95%. A new insight in designing power monitor systems is provided in PowerAnalyzer. In addition to ensuring the accuracy of reporting the running power consumption of each appliance, the energy-efficiency of Power monitor system is also addressed in our work.

References

[1] “CIAWorldFactbook,” https://www.cia.gov/library/publications/resources/the-world-factbook/fields/2233.html.

[2] “China Electricity Council,” http://english.cec.org.cn/No.119.index.htm.

[3] G. Tang, K. Wu, and J. Lei, “A Distributed and Scalable Approach to Semi-Intrusive Load Monitoring,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 6, pp. 1553–1565, June 2016.

[4] K. Ashton, “That Internet of Things Thing,” *RFID Journal*, vol. 22, no. 7, pp. 97–114, 2009.

[5] X. Jiang, S. Dawson-Haggerty, P. Dutta, and D. Culler, “Design and Implementation of a High-Fidelity AC Metering Network,” in *International Conference on Information Processing in Sensor Networks, 2009. IPSN 2009*, April 2009, pp. 253–264.

[6] G. W. Hart, “Prototype Nonintrusive Appliance Load Monitor,” 1985.

[7] ——, “Nonintrusive Appliance Load Monitoring,” *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[8] Y. Wang, X. Hao, L. Song, C. Wu, Y. Wang, C. Hu, and L. Yu, “Monitoring Massive Appliances by a Minimal Number of Smart Meters,” *ACM Trans. Embed. Comput. Syst.*, vol. 13, no. 2s, pp. 56:1–56:20, January 2014.

[9] J. Liu, “The function sensing framework for power consumption behaviors in households,” Ph.D. dissertation, Institude of Computing Technology, Beijing, 2015.

[10] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, “Edge Computing: Vision and Challenges,” *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[11] M. Kovatsch, S. Duquennoy, and A. Dunkels, “A Low-Power CoAP for Contiki,” in *IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems*, 2011, pp. 855–860.

[12] S. DeBruin, B. Campbell, and P. Dutta, “Monjolo: An Energy-Harvesting Energy Meter Architecture,” in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2013, pp. 18:1–18:14.

[13] B. Campbell and P. Dutta, “Gemini: A Non-invasive, Energy-Harvesting True Power Meter,” in *2014 IEEE Real-Time Systems Symposium*, December 2014, pp. 324–333.

[14] 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.

[15] “Belkin International. WeMo Insight Switch,” http://www.belkin.com/us/F7C029-Belkin/p/P-F7C029/.

[16] “D-Link Corporation,Inc. DSP-W215 Wireless Smart Plug,” http://shop.us.dlink.com/shop/shop-automation/smart-plugs/wi-fi-smart-plug-dsp-w215.html.

[17] “Uni-Trend Technology Limited. UNT230,” http://www.uni-trend.com/productsdetail.aspx?ProductsID=1304.

[18] “Xiaomi Technology Co.,Ltd. Xiaomi Smart Power Strip,” http://www.mi.com/chaxianban2/.

[19] K. Suzuki, S. Inagaki, T. Suzuki, H. Nakamura, and K. Ito, “Nonintrusive Appliance Load Monitoring Based on Integer Programming,” in *SICE Annual Conference, 2008*, August 2008, pp. 2742–2747.

[20] 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.

[21] H. S. Kim, “Unsupervised Disaggregation of Low Frequency Power Measurements,” in *Eleventh SIAM International Conference on Data Mining, SDM 2011*, April 2012, pp. 747–758.

[22] X. Zhang, Y. Wang, L. Chao, C. Li, L. Wu, X. Peng, and Z. Xu, “IEHouse: A Non-Intrusive Household Appliance State Recognition System,” in *International Conference on Ubiquitous Intelligence and Computing*, August 2017.

[23] J. Chung, C. Gulcehre, K. H. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” *Eprint Arxiv*, 2014.

[24] “STMicroelectronics. STM32L152 Ultra-Low-Power ARM Cortex-M3 MCU,” http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32l1-series/stm32l151-152/stm32l152cc.html.

[25] Z. Shelby and C. Bormann, *6LoWPAN: The Wireless Embedded Internet*. Wiley Publishing, 2010.

[26] “Texas Instruments. CC2650 Ultra-Low Power Wireless MCU,” http://www.ti.com.cn/product/CC2650.

[27] “AITE Precision Measurement Technology Co.,Ltd. AWE1611,” http://www.aitek.tw/CN/awe1611.html.

[28] D. Srinivasan, W. S. Ng, and A. C. Liew, “Neural-Network-Based Signature Recognition for Harmonic Source Identification,” *IEEE Transactions on Power Delivery*, vol. 21, no. 1, pp. 398–405, 2006.

[29] “Sense. Sense,” https://sense.com/product.html.

[30] X. He and D. Reiner, “Electricity Demand and Basic Needs: Empirical Evidence from China’s Households,” *Energy Policy*, vol. 90, pp. 212–221, 2016.

[31] S. Darby, “The Effectiveness of Feedback on Energy Consumption,” *A Review for DEFRA of the Literature on Metering, Billing and direct Displays*, vol. 486, 2006.