This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2776300, IEEE Transactions on Industrial Informatics

Towards Energy-Awareness Smart Building: Discover the Fingerprint of Your Electrical Appliances

Meng Ma, *Member*, *IEEE*, Weilan Lin, Jingbin Zhang, Ping Wang, *Senior Member*, *IEEE*, Yuchen Zhou, *Senior Member*, *IEEE* and Xiaoxing Liang, *Member*, *IEEE*

***Abstract*—Energy efficiency raises significant concerns as it is one of the most promising ways to mitigate climate change. Disaggregation and identification of individual electrical appliances activities is one of the essentials for energy preservation especially for smart buildings. This paper proposes a lightweight electrical appliance activity detection approach for smart building, which leverages a single smart metering device to establish a learning and detection processing for multiple appliances. In this system, data interpolation and transition detection algorithm are proposed to effectively reduce the cost of model training and optimize the detection accuracy. The concept of appliance fingerprint is proposed and a variety of fingerprints, including appliance-based and context-based, are defined to depict fine-grained appliance characteristics. Based on these fingerprints, the paper proposes a multi-source fingerprint-weighting KNN (FWKNN) classification algorithm and presents a boosting framework for continuous online learning and detection. A prototype system is implemented and demonstrated in IBM Bluemix PaaS cloud platform. Experimental result and analysis prove that FWKNN outperforms other benchmark methods in detection accuracy.**

***Index Terms*—appliance load monitoring; internet of things; activity learning; smart building; fingerprint extraction; classification algorithm**

I. INTRODUCTION

Energy resource preservation has recently become a major society concern, and motivated extensively research and development efforts [1]. Fortunately, emerging information technologies such as Internet of Things and Big Data provide us new approaches to tackle the energy issue. This work focuses on a key scenario of energy preservation - energy activity identification in smart buildings, also known as Appliance Load Monitoring (ALM) [2]. It is one of the essentials for energy preservation solutions, allowing them to obtain appliance-specific consumptions that can further be used to devise optimal energy utilization strategies [3]. Various approaches have been proposed taking advantage of signal decomposition algorithm for identifying certain electrical appliance load [4], however, without taking into account heterogeneous information sources (for example, user behavioral habits and environment information, etc.). Besides, fine-grained energy monitoring requires to deploy smart power outlets on every device of interest; however it incurs extra hardware cost and installation complexity [1, 5]. Moreover, traditional methods usually fail to provide a continuous learning and optimizing ability during the collection of appliances power usage.

The contributions of this paper are manifolds: first, this paper proposes a lightweight framework for smart building appliance activity learning and detection. It leverages a single smart metering device to establish a continuous learning and detection processing for multiple appliances. In this framework, a transition detection algorithm is designed to effectively reduce the cost of mining model training and optimize the detection accuracy. Second, this paper proposes the concept of appliance fingerprint, and designs a variety of appliance-based and context-based fingerprints to depict fine-grained appliance characteristics. Third, based on the defined fingerprints, the paper proposes a multi-source fingerprint-weighting KNN (FWKNN) classification algorithm and presents a boosting framework for continuous online learning and detection. Finally, a prototype is implemented and demonstrated in IBM Bluemix PaaS cloud platform. Experimental result and analysis prove that FWKNN outperforms other benchmark methods in detection accuracy.

The rest of this paper is organized as follows: related works are summarized in Section II. Section III presents the preliminary definitions and the high-level procedure of the proposed framework, and proposes the transition detection algorithms. Section IV defines the concept of appliance fingerprint and present several designed fingerprints to detect appliance activity. In Section V, the paper proposes the electrical appliance activity mining algorithm - FWKNN. Implementation, experiment and evaluation part are covered in Section VI. The paper is concluded in Section VII.

II. RELATED WORKS

Energy efficiency is becoming a challenging issue due to dramatically increasing energy demands. Sensing and monitoring systems for electricity appliances are expected to play an important role in managing and reducing overall energy consumption. A foundation of such system is the technologies to identify fine grained consumption of each domestic appliance [6]. This information also reveals the resident's energy consumption activities and preference. The pioneering work started by Hart in the beginning of the 1990s [7]. Related technologies and approaches have been extensively studied and reviewed in literatures, see [8] and [9] for example. Major proposed methods fall into two categories. The first kind devotes to the designing of sensing devices and analog/discrete signal processing techniques for load separation. For example, Jiang et. al. [10] developed a wireless networked sensor that measures the power

Meng Ma and Jingbin Zhang is with the School of Electronics Engineering and Computer Science, Peking University, Beijing, China (e-mail: mameng @pku.edu.cn, zhangjingbin@pku.edu.cn).

Weilan Lin is with the School of Software and Microelectronics, Peking University, Beijing, China (e-mail: lin.weilan@pku.edu.cn).

Ping Wang is with the National Engineering Research Center for Software Engineering and the School of Software & Microelectronics, Peking University, Beijing, China (e-mail: pwang@pku.edu.cn). He is the corresponding author of this paper.

Yuchen Zhou and Xiaoxing Liang is with the IBM Research - China, Beijing, China (e-mail: zhouyuc@cn.ibm.com, lxxing@cn.ibm.com).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2776300, IEEE Transactions on Industrial Informatics

consumption at an outlet. Many other devices are designed to sense the real-time electricity consumption, for example Neurio[1] and Sense[2]. Typically systems like ElectriSense [11] and ViridiScope [1] are proposed to analyze and detect the transient physical activity of loads during the load identification by wavelet transform of the time-frequency domain [12]. With the emerging of big data technology, a number of data mining-based approaches are proposed to provide new solutions. In [13], the appliance activity classification task is carried out by Support Vector Machines (SVM) and *k*-Nearest Neighbors methods (KNN). Zhao et al. proposed a training-less solution for non-intrusive appliance load monitoring using a graph-based signal processing method [14]. In [15], Ruzzelli et al. employed the Artificial Neural Network (ANN) training model, and [16] proposed a multiple-class support vector machine (M-SVM) to recognize different appliances. Probability-based data mining approaches are also introduce such as Hidden Markov Model [17-19], Prior Models [20], and also hybrid models [21]. As we can see from the above analysis, these proposed approaches are not taking into account heterogeneous information sources (for example, user behavioral habits and environment information, etc.) for identifying certain electrical appliance load. Moreover, they cannot provide a continuous learning and optimizing ability during the collection of appliances power usage. Besides, some traditional approaches requires to deploy smart power outlets on every device of interest, which incurs extra hardware cost and installation complexity.

III. System Framework

This section introduces the general framework and functions of the proposed solution. Based on the basic preliminary definitions, a transition detection algorithm is designed to optimize the detection efficiency and accuracy.

A. Proposed Framework

This paper proposes a lightweight appliance activity learning and detection framework for smart building. This framework leverages a single smart metering device to establish a continuous learning and detection processing for multiple appliance (See Figure 1).

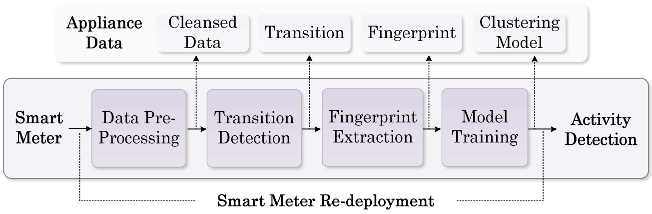


Fig. 1. A high-level procedure of proposed framework

The high-level procedure of this framework is consisting of four major functions, namely *data pre-processing*, *transition detection*, *fingerprint extraction* and *model training/detection*. In this framework, a smart meter device is firstly deployed with arbitrary appliance to collect its electrical consumption data. In the meanwhile, the system performs online data pre-processing to generate cleansed data. Subsequently, all the transitions of appliance will be identified in the next process. Following that, the framework extracts various fingerprint data sequence from original data source and train the mining model with these fingerprints. Finally, the appliance activity can be detected by the classification model. The re-deployment of the smart meter device into another appliance triggers a new learning and detection cycle. Repeating this cycle, the system learns and detects various activity characteristic of multiple kinds of appliances. Therefore, the framework provides a continuous activity learning, detection and self-optimizing ability during the collection of various appliances power usage by a single smart device, which does not incur extra hardware cost and installation complexity. Four major functions of this framework are individually elaborated in the following sections.

B. Data Pre-Processing

In order to demonstrate and discuss the detailed process of proposed method and algorithms, this section defines the preliminary notations for data and time series pre-processing in this paper:

Reading Data. The tuple of raw sensing data is denoted as:

$$R = \{t, P(t), S(t), V(t)\}$$

in which t stands for the timestamp. $P(t), S(t), V(t)$ is the functions of timestamp which return the active power, apparent power and mains RMS on specific time point t.

Time Series. Time series is a sequence of reading data, denoted as:

$$T = \{R_1, R_2, \cdots, R_n\}$$

and satisfies $t_1 \le t_2 \le \cdots \le t_n$. The length is the time period of T, defined as $length(T) = t_n - t_1$.

Sub Time Series. Given a time series T with length n, its sub time series is denoted as:

$$T_w = \{R_p, R_{p+1}, \cdots, R_q\}$$

stands for a continuous subsequence of T. The length of T_w is w and $t_p - t_q = w$.

Sliding Time Window. Given a time series T with length n and a sliding time window size w $(w < n)$. The sliding time window is denoted as:

$$W = \{T_{w1}, T_{w2}, \cdots, T_{wn}, \cdots\}$$

which is a set of sub time series of T with increasing order of first occurrence timestamp $t_{w1} < t_{w2} < \cdots < t_{wn}$. Sliding window w is the basic timing processing methods. The transition detection, fingerprint extraction and a classification model training and detection are all based on a sliding window processing mode.

C. Transition Detection

Transition detection in this framework is designed to identify possible occurrence of appliance working status switches (e.g. ON/OFF) in main line or collected appliance-specific time series. Within sliding time window, the detection result of transition detection determines whether the system should perform subsequent time series training and mining. As a result, the accuracy of this process is important for the performance of the completely detection cycle. Theoretically, electrical appliances, especially for high-power electrical appliances, in the

[1] Neurio - Home Energy Monitor. http://neur.io/
[2] Sense - Home Energy Monitor. https://sense.com/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2776300, IEEE Transactions on Industrial Informatics

instant when they are switching their working status, there will be a substantial power change (such as a cliff-like power edge) reflecting on its power. Especially for high-power electrical appliances, such as boiler, kettle and heater, this change is an important target for transition detection. The proposed framework introduces an efficient algorithm for transition detection based on following concepts:

Critical Point* and *Stable Point: Within a time window w, if the power change of an appliance $\nabla(t) = P(t) - P(t-1)$ is larger than 20W and the change rate $\nabla_r(t) = \nabla(t)/P(t)$ exceed threshold $= 10\%$, t is a critical point. Otherwise, t is a stable point.

***Minimum Stable Period* (*MSP*)**: Within a time window w, the minimum stable period requires the existence of a sub time series $T_{MSP} = \{R_i, R_{i+1}, \cdots, R_n\}$, satisfy that all its time point t_i are all stable point.

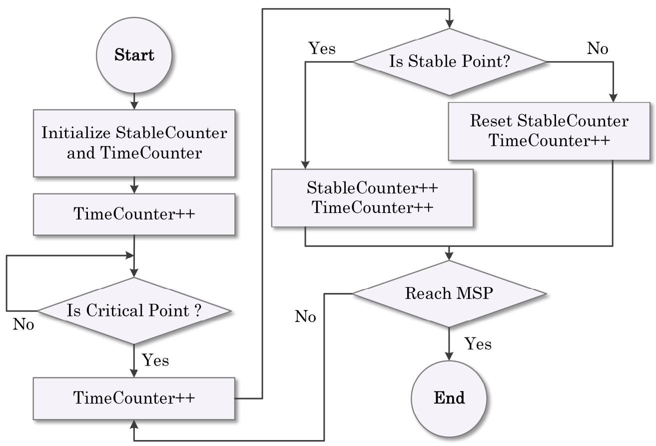


Fig. 2. Transition detection algorithm flow diagram

Figure 2 and Algorithm 1 summarize the proposed transition detection algorithm. The process of the algorithm includes following steps:

Step 1: Search power change in time series, if exists a time point t satisfies $\nabla(t) > 20W$ and the $\nabla_r(t) > 10\%$, mark t as a critical point and trigger the following detection;

Step 2: Search stable point which follows the marked critical point t. Add the *StableCounter* for every found stable point, otherwise reset the counter to 0;

Step 3: If the *StableCounter* reaches MSP p, a transition Tr is detected in the period of $[t, t+p]$, the step value of the transition is $StepValue(Tr) = P(t+p) - P(t)$;

Step 4: Output the transition in the format $Tr = (t, ON/OFF, StepValue)$. If $StepValue > 0$ then the Tr is an ON transition, otherwise the transition is OFF.

Figure 3 is a detected activity instance on television. In the first stage, the transition detection algorithm searches the power change and change rate for critical point (pointed out in Figure 3). When finding a critical point, the next step is the search for stable point, therefore the power pattern in adjustment period will be recognized as disturbance or noise. Once the appliance enters the stable period (when it finds a series of stable points and does not re-enter a transition period within a certain time window), it represents the power edge is valid for a transition detection. Note that the selected threshold $\nabla(t) > 20W$ and $\nabla_r(t) > 10\%$ in algorithm is an empirical value. This threshold should be adjusted according to the electrical environment and the characteristics of the target appliances. A small threshold may lead to frequent false positives in detection results. On the contrary, if the algorithm detects transition with a large threshold, some possible transitions will be lost to catch the actual appliance activities.

Algorithm 1. Transition Detection

Input. Time series T, minimum stable period p.

```
new StableCounter, TimeCounter
for each R_i ∈ T search critical point t_cp
  for each R_j ∈ T, t_j > t_cp search stable point
    if t_sp is stable point then StableCounter++;
      when StableCounter=p, P(t_sp) − P(t_cp) > 0
        Tr = (t_cp, ON, P(t_sp) − P(t_cp))
      else Tr = (t_cp, OFF, P(t_cp) − P(t_sp))
    end
end
```

Output. Transition detection result Tr.

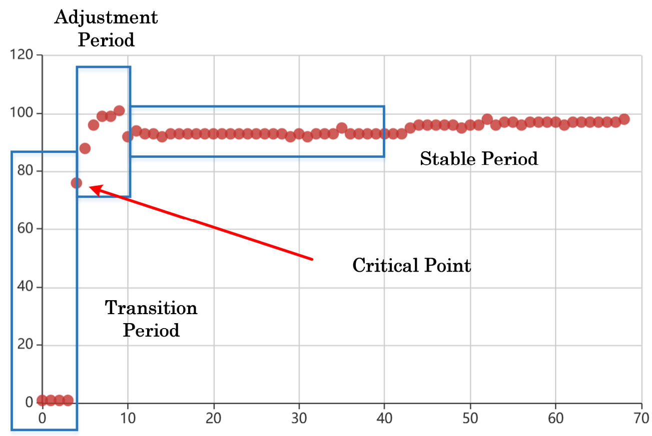


Fig. 3. A transition instance of TV ON activity

IV. Appliance Fingerprint Extraction

This section introduces and defines the concept of appliance fingerprint. Two types of them, appliance-based fingerprint and context-based fingerprint are proposed and a variety of them is defined to depict fine-grained appliance characteristics.

A. Appliance-based Fingerprint

Smart meters are replacing our existing electricity or gas meter, sending information to a display unit and power authorities. Ordinary meters only provide energy consumption measurements for the entire building yet behavioral research suggests that consumers are best able to manage their electricity consumption by given appliance-by-appliance information. In order to depict fine-grained appliance characteristics, the framework introduces the concept of *appliance fingerprint* to describe particular pattern sequence, and divides them into two main types, *appliance-based* and *context-based fingerprint*.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2776300, IEEE Transactions on Industrial Informatics

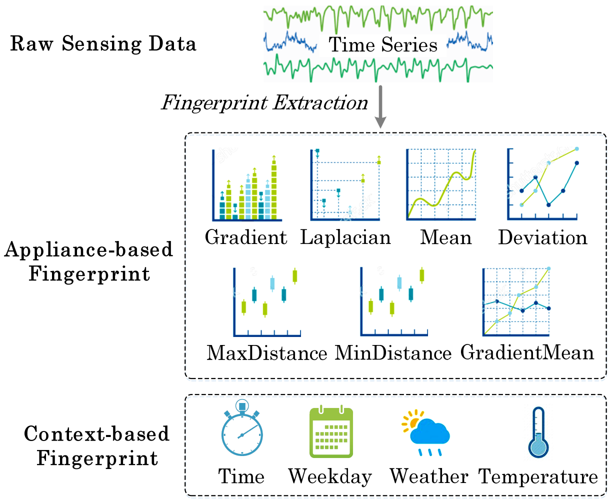


Fig. 4. Appliance fingerprint extraction

As Figure 4 shows, appliance's fingerprint depicts the unique activity characteristics of certain electrical appliance. Appliance-based fingerprints are the patterns that reflect in the consumption data. Additionally, context-based fingerprints reflect the relationship between user, appliance and environment. Naturally, appliances have their own special characteristics. For instance, boiler and fridge show different impacts on overall energy consumption measurements when they are turned ON and OFF. Moreover, different household appliances will be used at different time and environment according to user' s daily routines. As a result, single fingerprint is difficult to fully describe and reflect the characteristics of appliance's activity. The framework proposes and defines the following fingerprints to depict electrical appliance activity patterns.

(1) *Gradient, Gradient Ratio and Laplacian Fingerprint*

The gradient, gradient ratio and laplacian fingerprint (denote as ∇, ∇_r and Δ respectively) evaluate the trends in appliance energy consumption within the sliding window w. These fingerprints can be used to identify specific power spikes or edges and remove the base energy level from other appliances which bias the algorithm result.

$$\left.\begin{aligned} \nabla(t_i) &= P(t_i) - P(t_i - 1) \\ \nabla_r(t_i) &= \nabla(t_i)/P(t_i) \\ \Delta(t_i) &= \nabla^2(t_i) = \nabla(t_i) - \nabla(t_i - 1) \end{aligned}\right\} \forall t_i \in w \tag{1}$$

(2) *Mean and Standard Deviation Fingerprint*

These fingerprints depict the statistic features of certain appliance power usage. They are computed within the sliding window w for the aggregated gradient ∇ and its standard deviation $\overline{\nabla}$.

$$\left.\begin{aligned} \overline{\nabla}(t_i) &= \tfrac{1}{w}\left(\textstyle\sum_{t=t_i}^{t_i+w} \nabla(t)\right) \\ \mathrm{S}_{\nabla}(t_i) &= \sqrt{\tfrac{1}{w-1}\textstyle\sum_{t=t_i}^{t_i+w}\left(\nabla(t) - \overline{\nabla}(t_i)\right)^2} \end{aligned}\right\} \forall t_i \in w \tag{2}$$

(3) *Distances to Local Maximum and Minimum Fingerprint*

These two fingerprints monitor the distance from the current power measurement $P(t_i)$ to its local maximum and minimum (denoted as $d_{max}(t_i)$ and $d_{min}(t_i)$) within the sliding window w. They depict the pattern that demonstrate the local changes with base max and min power measurement removed.

$$\left.\begin{aligned} d_{max}(t_i) &= max\{\forall P(t)\} - P(t_i) \\ d_{min}(t_i) &= P(t_i) - min\{\forall P(t)\} \end{aligned}\right\} \forall t_i \in w \tag{3}$$

B. *Context-based Fingerprint*

In addition to the appliance's own electrical characteristics, the activity of appliances and the context (user and environment) are also closely related. Previous studies mainly focus the recognition of appliance-based fingerprint but ignore the context-based information. In this section, additional fingerprints that reflect the usage pattern of appliance are defined, such as time, day of week, weather and temperature. Time, for example, shows apparent correlation with many kitchen appliances. These appliances are frequently used in the morning and evening of weekdays. However, they are more likely to be used at noon and evening in weekends. Correspondingly, in other time, they are less likely to be used. The usage of other appliances like air conditioning, heating devices is affected by weather and environmental condition. Because these context-based fingerprints have different measurement scales, in order to leverage them in our mining model, a proper classification of these fingerprints is defined as follow:

(1) *Time Fingerprint*

The time fingerprint is a measurement of the hour of an appliance event. It plays an important role in representing the hourly temporal pattern in appliance usage. Based on the correlation between the use of electrical appliances and time period, 24 hours in a day are divided into 8 periods of time (3 hours for each), to distinguish the usage feature in the day time period different characteristics. They are represented by discrete number: 0 - Before Dawn (2:00 to 5:00); 1 - Morning (5:00 to 8:00), 2 - Forenoon (8:00 to 11:00), 3 - Noon (11:00 to 14:00), 4 - Afternoon (14:00 to 17:00), 5 - Evening (17:00 to 20:00), 6 - Night (20:00 to 23:00), 7 - Midnight (23:00 to 2:00). The time fingerprint is defined by the equation:

$$Time(t) = d; d \in [0,7] \tag{4}$$

(2) *Weekday Fingerprint*

As aforementioned, the usage of some appliance is associated with different day in a week. Hence, the framework defines "weekday" as context-based fingerprint to describe the relationship between appliance activity and different weekday. This fingerprint has 7 different values:

$$Week(t) = w; w \in [0,6] \tag{5}$$

(3) *Weather and Temperature Fingerprint*

In order to evaluate the relationship between usage of electrical appliances and environmental conditions, two fingerprints are defined: *Weather* and *Temperature*. Similarly, common weathers are defined as: 0 - Sunny; 1 - Cloudy; 2 - Rainy; 3 - Storm; 4 - Snow. Temperature is divided into five categories: 0 - <0 ºC; 1 - 0 ºC to 10 ºC; 2 - 10 ºC to 20 ºC; 3 - 20 ºC to 30 ºC; 4 - >30 ºC.

$$\begin{cases} Weather(t) = c; c \in [0,4] \\ Temp(t) = t; t \in [0,4] \end{cases} \tag{6}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2776300, IEEE Transactions on Industrial Informatics

V. THE MINING MODEL

This section proposes the mining model for appliance activity learning and detection. It also presents the similarity measurement and fingerprint-based weighting method used in this model.

A. Model Structure

The overall structure of the mining model, are consisted with two main stages (See Figure 5), namely (1) model training; (2) activity detection. When the system detects a transition from the appliance measurement, it triggers the model training process, updates the model weighting parameter, and stores the input fingerprint into metering database. Otherwise, if the transition occurs in the main line, the activity detection module identifies the activity based on historical measurement. The presented model structure establishes an online and incremental supervised learning process. Apparently, the identification of appliance activities is a multi-model classification problem. According to the data and problem characteristic, KNN (*k*-nearest neighbor) algorithm is adopted as the basic mining approach. KNN is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure. The main idea of KNN is to find the closest N samples in the sample feature space. As a result, native KNN is a lazy learner without training process. This paper proposes a fingerprint-weighting KNN algorithm, namely FWKNN, to detect appliance's activity and designs an online learning mechanism to optimize the mining model with new fingerprints. The similarity measurement and the weighting method in FWKNN are defined and discussed in the following sections.

(1) Similarity Function

The following similarity definition and discussion are based on the gradient fingerprint, which is similar with other appliance-based fingerprints except notations. The format of training sample for electrical appliance *a* can be denoted as:

$$x_i = (t_i, T_i = \{\nabla(t_i), \cdots, \nabla(t_{i+w})\}, a, ON/OFF) \tag{7}$$

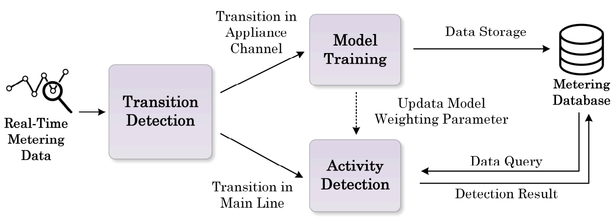


Fig. 5. Mining model structure

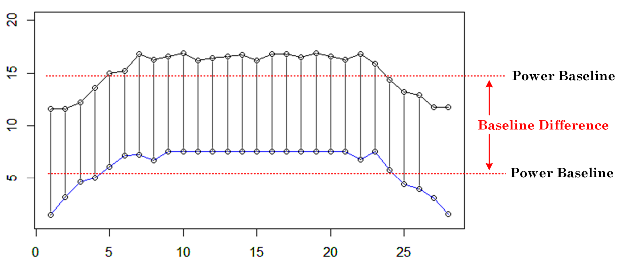


Fig. 6. Distance measurement with baseline difference

in which t_i is the transition timestamp, $T_i = \{\nabla(t_i), \cdots, \nabla(t_{i+w})\}$ is the gradient time series.

Since the load of single appliance can be easily impacted by other appliances or signal noises. For example, as Figure 6 shows, using traditional Euclidean distance, two similar load patterns can have large distance because they have different power baseline. Hence, when calculating the distance, a notion of *baseline difference* is introduced when measuring the distance for two corresponding points, in order to eliminate the impact from noise and other electrical load difference. Given two sample x_i, x_j, the baseline difference between them is defined as:

$$\nabla_{base} = \frac{1}{w}\left|\sum_{k=0}^{w-1}\nabla(t_{i+k}) - \sum_{k=0}^{w-1}\nabla(t_{j+k})\right| \tag{8}$$

Therefore, when calculating distance, the framework uses a time-aligned Euclidean distance measurement for two data samples and subtracts base power difference between them. The calibrated similarity is calculated by the following equation:

$$dist(x_i, x_j) = \sqrt{\sum_{k=0}^{w-1}(|\nabla(t_{i+k}) - \nabla(t_{j+k})| - \nabla_{base})^2}$$
$$= \sqrt{\sum_{k=0}^{w-1}\left(|\nabla(t_{i+k}) - \nabla(t_{j+k})| - \frac{1}{w}\left|\sum_{k=0}^{w-1}\nabla(t_{i+k}) - \sum_{k=0}^{w-1}\nabla(t_{j+k})\right|\right)^2} \tag{9}$$

Although, standardization is commonly used in similarity measurement stage of KNN such as min-max. However, in this framework, various fingerprints have independent KNN models. Therefore, the measurement scales in different KNN models are consistent and standardization is not necessary. For each independent KNN model, given a query instance x_q to be classified, let $x_1, \cdots, x_k$ denote the k instances nearest to x_q, the KNN classification result of x_q can be denoted by the equation

$$f(x_q) \leftarrow argmax_{i \in I}\sum_{j=1}^{k}\delta\left(i, f(x_j)\right) \tag{10}$$

where $i \in I$ is the ID of appliance, $\delta(a,b) = 1$ when $a = b$, otherwise, $\delta(a,b) = 0$. Equation 10 shows that the essential of native KNN algorithm is the search process for the majority vote of k labelled neighbors of x_q. In other words, the query instance x_q is classified with the appliance ID being assigned to the most common class among $x_1, \cdots, x_k$.

(2) Fingerprint Weighting

The irrelevant and noisy fingerprints may influence the neighbor searching for highly relevant fingerprint in native KNN model. As a result, the accuracy of KNN algorithm is likely to deteriorate. To tackle this issue, this section introduces several weighting approaches to improve the accuracy of native KNN. These weighting mechanisms are used to approximate the optimal degree of influence of individual fingerprints using training set. Specifically, when successfully applied relevant fingerprints are attributed with a higher weight value, whereas irrelevant fingerprints are given a lower weight. In FWKNN, three fundamentally approaches are introduced, *distance weighting*, *proportion weighting* and *context-based fingerprint weighting*:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2776300, IEEE Transactions on Industrial Informatics

Distance Weighting: The first refinement of the native KNN classification algorithm is to weigh the contribution of each neighbors according to their distance to the query point x_q, giving a higher weight dw_i to closer neighbors, where the weight is defined by the equation:

$$dw_i = \frac{1}{dist(x_q, x_i)^2} \tag{11}$$

Proportion Weighting: The non-uniform distribution of the training data from different appliances may cause category-sensitive problem in classification. This framework takes advantage of the proportion information and leverages an inverse-proportion weighting strategy, to assign larger weight to the appliance with smaller training data proportion, eliminating the impact from the number of appliance categories. More specific, if there are n different appliances, and their corresponding training data sets are denoted as $S = \{S_1, S_2, \cdots, S_n\}$, for a query point x_q, the framework assigns a weight for its distance to a sample point x_i which belongs to appliance i, where the weight is defined by the equation:

$$pw_i = \frac{1}{S_i / \sum_{j=1}^{n} |S_j|} = \frac{\sum_{j=1}^{n} |S_j|}{S_i} \tag{12}$$

Context-based Fingerprint Weighting: This weighting approach is based on the searching for similar historical context records, including time, day of the week, weather and temperature, to evaluate the probability of an appliance activity at a certain time point. It is based on the assumption that, an appliance is more likely to turn ON or OFF if it is more frequently used than other appliances in similar context. Context-based fingerprint weighting is actually a learning process of user preference by recording the historical context data and computing the probability of given activity. In a time point t, let $\{c_1, c_2, c_3, c_4\}$ denote the time, week weather and temperature fingerprint. The context-based fingerprint weighting for appliance i is an estimation on the probability of its historical activity:

$$fw_i(t) = \prod_{k=1}^{4} \left(1 + \frac{count(i, c_k)}{|S_i|}\right) \tag{13}$$

where $count(i, c_k)$ represents the amount of training samples that the k^{th} context-base fingerprint equals to c_k.

As a summary, above proposed weighting methods are independent to reflect the characteristic of appliance activities. Therefore, by multiplying equation (11) to (13) and replacing the original KNN classification equation (10), the weighted KNN classification result of x_q can be computed by the equation by

$$\hat{f}(x_q, t) \leftarrow argmax_{i \in I} \sum_{j=1}^{k} dw_i \cdot pw_i \cdot fw_i(t) \cdot \delta\left(i, f(x_j)\right) \tag{14}$$

where $f(x_j)$ return the appliance ID of the k-nearest sample instance x_j. It means that the framework searches for the majority vote of k labelled neighbors $x_1, \cdots, x_k$ of the target x_q under the influence of three weighting value dw_i, pw_i and $fw_i(t)$.

B. Model Training and Activity Detection

As a lazy learner, the training of KNN is simply the storage of training data samples. In this case, the appliance activity mining model is consisted of serval independent KNN models for different appliance-based fingerprints, denoted as $M = \{M_1, M_2, \cdots, M_n\}$ where M_i is the KNN model for No.i appliance-based fingerprint. In order to dynamically adjust the weight of different sub-classification model and improve the overall accuracy, the framework uses a boosting mechanism to assign higher weight for the sub-classification model with lower error rate. In order to record the history of classification result for each sub-model, a voting matrix $V = (v_{i,j})_{n \times m}$ is defined, where n is the number of appliance-based fingerprint and m is the number of trained appliance. The error rate of model M_i is defined by the proportion of correct classification from all the sample instance of appliance i:

$$error(M_i, A_j) = 1 - \frac{v_{i,j}}{|S_j|}, \quad v_{i,j} \in V \tag{15}$$

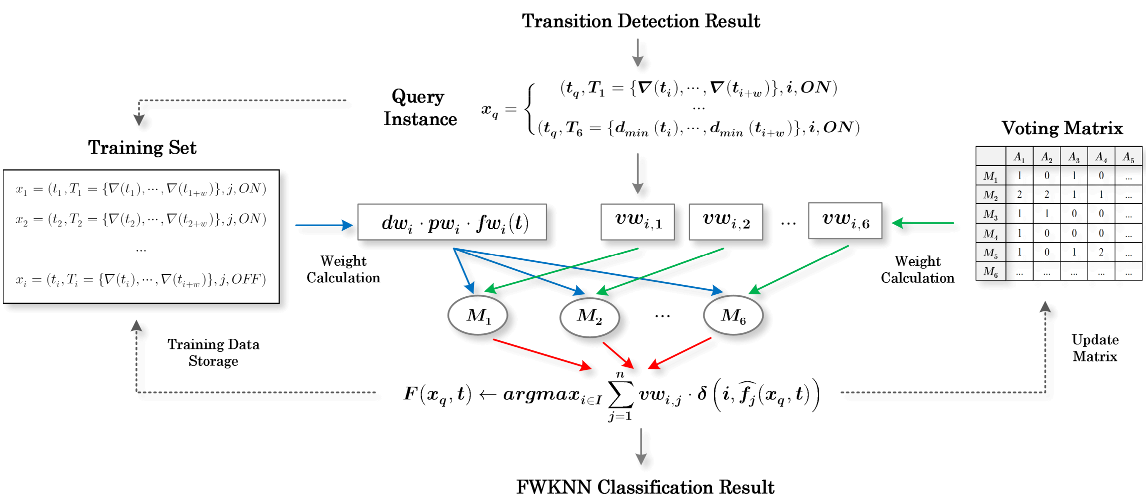


Fig. 7. FWKNN training model and activity learning structure

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2776300, IEEE Transactions on Industrial Informatics

Equation (15) statistics the proportion of incorrect result $v_{i,j}$ made by sub-model M_i for appliance A_j, that is, the error rate of sub-model M_i for the classification of A_j. Based on this error rate function (15), the voting weight of model M_i can be calculated by the equation:

$$vw_{i,j} = \ln\frac{1-error(M_i, A_j)}{error(M_i, A_j)} = \ln\frac{v_{i,j}}{|S_j| - v_{i,j}} \qquad (16)$$

The value range of equation (15) is [0,1]. As a result, when the error rate of certain sub-model M_i to A_j is approaching to 1, which represents a poor classification ability of M_i, equation (16) will assign a low weight to M_i for A_j. On the contrary, when $error(M_i, A_j)$ is approaching to 0, $vw_{i,j}$ is increasing to ∞. Particularly, if $error(M_i, A_j) = 0$, the framework sets $vw_{i,j} = 1$ and $vw_{k,j} = 0$ for any other $k \neq i$.

Figure 7 demonstrates the process of training and activity learning of this boosting model. The data sources include training set S and voting matrix V. In gradient fingerprint, for example, the training data for the appliance ID j is denoted as $x_i = (t_i, T_i = \{\nabla(t_i), \cdots, \nabla(t_{i+w})\}, j, OFF)$. The weight dw_i, pw_i and $fw_i(t)$ of sub-model M_i can be computed according to equation (11)-(13). Afterward, the framework calculates the corresponding voting weight for each sub-model $vw_{i,1}, vw_{i,2}$, $\cdots$,$vw_{i,n}$ based on equation (14). Given an query instance

$$x_q = \begin{cases} (t_q, T_1 = \{\nabla(t_i), \cdots, \nabla(t_{i+w})\}, i, ON) \\ \cdots \\ (t_q, T_n = \{d_{min}(t_i), \cdots, d_{min}(t_{i+w})\}, i, ON) \end{cases} \qquad (17)$$

the classification result of x_q is an optimal parameters problem:

$$F(x_q, t) \leftarrow argmax_{i\in I} \sum_{j=1}^{n} vw_{i,j} \cdot \delta\left(i, \widehat{f_j}(x_q, t)\right) \qquad (18)$$

Every time $F(x_q, t)$ is calculated, the training set and voting matrix will be updated. If $F(x_q, t) = \widehat{f_j}(x_q, t)$ for sub-model M_j, then $v_{i,j} \leftarrow v_{i,j} + 1$. The detailed detection processing includes the following steps (See Algorithm 1 for its details):

Algorithm 1. FWKNN Algorithm

Input. Query instance x_q, time stamp t, training set S, KNN model set M.

```
new Array Q[k]
randomly select {s_j, ..., s_{j+k}} ⊂ S
    insert dw_i · pw_i · fw_i(t) · dist(s_j, x_q) into Q
for each s_i ∈ S calculate dist(s_i, x_q)
    insert dw_i · pw_i · fw_i(t) · dist(s_i, x_q) into Q
end
for each M_j
    calculate vw_{i,j};
    calculate f̂_j(x_q, t);
end
return argmax_{i∈I} Σ_{j=1}^{n} vw_{i,j} · δ(i, f̂_j(x_q, t))
```

Output. Classification result.

Step 1: Input query instance x_q;

Step 2: Initialize k-size descending queue for nearest neighbor. Randomly selected k instances and calculate their distance from x_q as initial value;

Step 3: Traverse the training set and for every sample instance s_i, calculate original distance $dist(s_i, x_q)$ and the fingerprint weight $dw_i \cdot pw_i \cdot fw_i(t)$. Compare and replace the priority queue if the weighted distance is larger;

Step 4: Calculate the voting weight $vw_{i,j}$ for each sub-model;

Step 5: The classification result of x_q is the most voted class among all the sub-models.

VI. Prototype and Experiments

A. Prototype Implementation

A prototype system is implemented with micro-service based architecture (See Figure 8). Such multi-layer structure is designed to handle large-scale energy sensing data with IBM cloud platform as a service (PaaS) Bluemix[3].

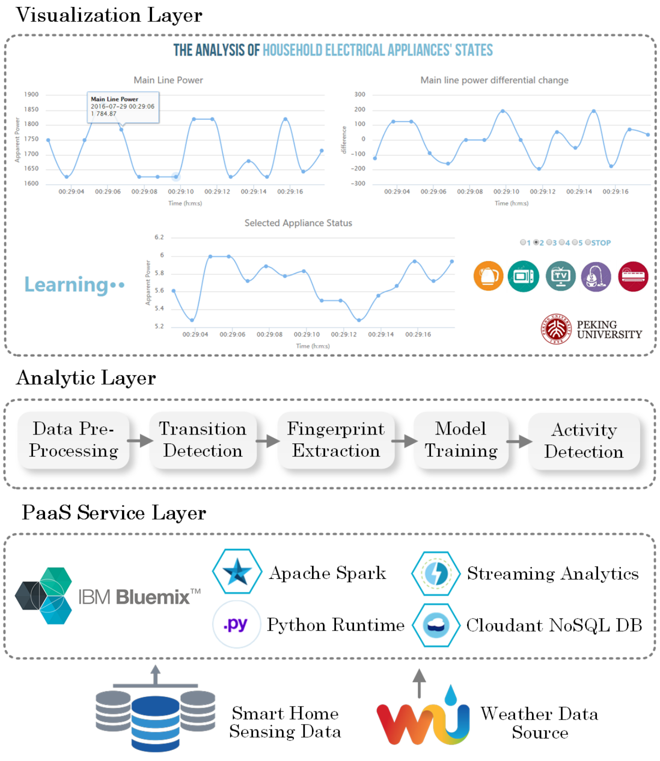


Fig. 8. System prototype

The smart building sensing data is collected and uploaded into cloud storage services Cloudant NoSQL database. The context-based fingerprint is obtained from external sources. For example, the system collects weather information from weather services such as weather underground[4] and fed into the system via REST interfaces. Different analytic components are implemented in the *Analytic Layer*. Proposed fingerprint extraction and FWKNN algorithm are built and implemented with Spark Streaming Analytics cloud service in Bluemix. In system

[3] IBM Bluemix. http://www.ibm.com/cloud-computing/bluemix/
[4] Weather Underground. https://www.wunderground.com/

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2776300, IEEE Transactions on Industrial Informatics

front-end, a dashboard web application in the *Visualization Layer* provides different interactive user interfaces to explore the results from the analytic layer results. The prototype system uses Python runtime in Bluemix to build a user interface. This interface displays real-time main line consumption and fingerprint measurements. This demo also displays the result of appliance activity learning and detection.

B. *Experiments and Evaluation*

(1) *Experiment Design*

The proposed appliance activity learning and detection algorithm are evaluated using UK-DALE dataset [5]. UK-DALE is an open-access dataset from the UK recording a domestic appliance-level electricity usage at a sample rate of 16 kHz for the whole-building and at 1/6 Hz for 111 individual appliances. It is collected from five houses, one of which was recorded for 655 days, which provides a longest duration among the public datasets at this diverse sample rate. Table 2 describes the data columns of UK-DALE dataset for its 1 Hz and raw data for main line (outlet power) and 1/6 Hz raw data for different channels (individual appliances). There are four columns in each 1Hz raw data file recording the whole-building main power demand every second: T - Unix Timestamp; P - Active power; $|S|$ - Apparent power; V_{rms} - Main line RMS (root mean squared) voltage. For 1/6 Hz data of separate appliance channels, there are only two columns: T and P. Several typical appliances in UK-DALE dataset, boiler, laptop, washing machine, dishwasher, kettle and fridge are selected for experiment.

Table 2 Data format of UK-DALE dataset

	Col 1	Col 2	Col 3	Col 4
1Hz Raw Data	T	P	$\lvert S\rvert$	V_{rms}
1/6Hz Raw Data	T	P		

Notation	Details	Notation	Details
T	Timestamp	$\lvert S\rvert$	Apparent power
P	Active power	V_{rms}	Mains RMS Voltage

In experiment, FWKNN is compared with state-of-the-art Decision Tree Learner (DTL) [22], Support Vector Machine (SVM) [13] and Hidden Markov Model (HMM) [18] algorithms. Ten different periods of data series are selected and all experiments are performed with 1-month data for training and its subsequent 1-month data the testing. Three different experiments are designed to evaluate the effectiveness and efficiency of proposed method. In the first experiment, we compare the result of FWKNN algorithm under different parameter k. Two types of electrical appliance are selected, boiler stands for high-power appliance and laptop for low-power. The second experiment aims to evaluate the performance of different algorithms - FWKNN, DTL, SVM and HMM, using all six kinds of electrical appliances. The last experiment compares FWKNN algorithm with different input fingerprints to verify the effect of multi-fingerprint weighting and the contribution of context-based fingerprint to the algorithm accuracy.

(2) *Performance Metric*

The evaluation uses the following metrics to evaluate our method and compare with other benchmarks:

True Positive (*TP*): if the query instance x_q is classified to appliance i, and appliance i has actually switched, the detection result is true positive.

False Positive (*FP*): if the query instance x_q is classified to appliance i, however appliance i has not switched, the detection result is false positive.

False Negative (*FN*): if the query instance x_q is not classified to appliance i, while appliance i has switched, the detection result is false negative.

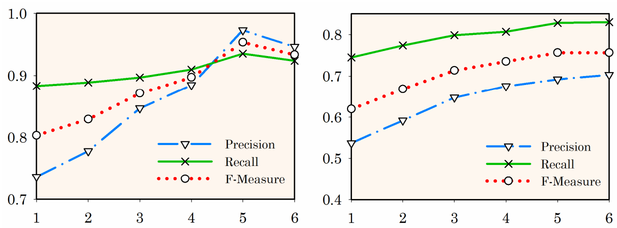
Precision (*PR*): the proportion of the correct detected activity, calculated by $PR = TP/(TP+FP)$, representing the fraction of the correctly identified activity of the appliances.

Recall Rate (*RE*): the proportion of activity that were correctly identified, as calculated by $RE = TP/(TP+FN)$. Represents the ratio between the correctly detected activity of the appliances and their total number of actual activities.

F-Measure (*FM*): weighted harmonic mean of precision and recall, calculated by $FM = 2*(PR*RE)/(PR+RE)$, where both contributions of the precision and the recall are equally weighted in the F-Measure coefficient.

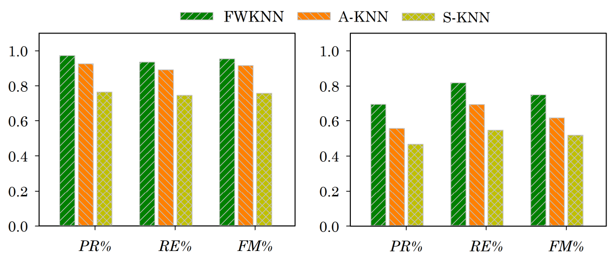
(3) *Experimental Result and Analysis*

Different k Value. Figure 9 shows the comparison result on FWKNN algorithm performance (*Precision*, *Recall* and *F-Measure*) under different *k*-values for high-power appliance, boiler (See Figure 10(a)) and low-power laptop (See Figure 10(b)). It is clear that, under different parameters *k*, the detection precision for high-power appliance is significant better than low-power appliance. For example, in this experiment, when set *k*=5, FWKNN algorithm has 97.3% accuracy for boiler activity detection, however only has 69.2% accuracy for laptop. Moreover, the detection accuracy is significantly lower



(a) Different *k* vaule for Boiler (b) Different *k* vaule for Laptop

Fig.9. Performance of FWKNN with different *k* value



(a) Different Fingerprint for Boiler (b) Different Fingerprint for Laptop

Fig. 10. Performance of FWKNN with different fingerprint

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2776300, IEEE Transactions on Industrial Informatics

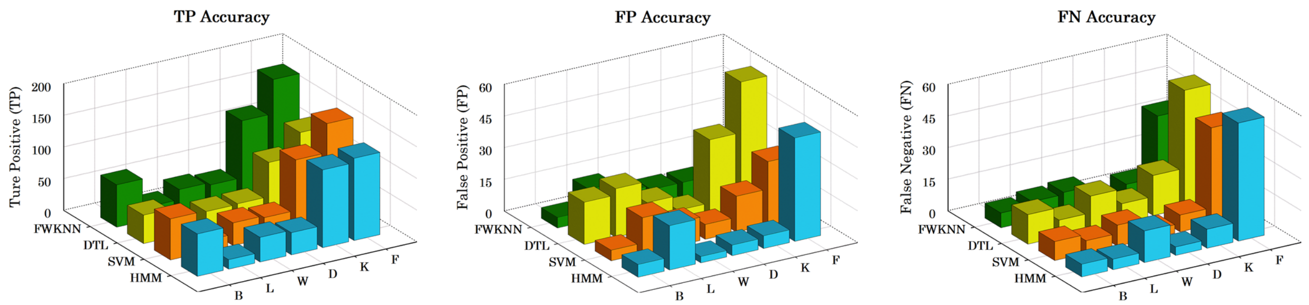


Fig. 11. TP, FN, and FP for different methods on boiler (B), laptop (L), washing machine (W), dishwasher (D), kettle (K) and fridge (F).

Table 3. Experimental result of precision

	Precision (*PR*%)			
	FWKNN	DTL	SVM	HMM
Boiler	97.30	76.36	91.67	94.00
Laptop	69.23	30.30	60.00	41.67
Washing Machine	88.68	58.97	61.90	92.86
Dishwasher	90.24	61.29	65.71	50.00
Kettle	97.81	76.15	86.92	94.96
Fridge	89.95	55.97	73.15	71.22
Average	91.08	62.28	76.28	78.23

Table 4. Experimental result of recall

	Recall (*RE*%)			
	FWKNN	DTL	SVM	HMM
Boiler	93.51	63.64	74.32	61.84
Laptop	81.82	55.56	62.07	75.00
Washing Machine	92.16	56.10	63.41	72.22
Dishwasher	92.50	59.38	71.88	51.72
Kettle	99.26	74.11	93.39	85.61
Fridge	92.49	53.94	55.90	50.77
Average	93.35	60.64	69.39	64.14

Table 5. Experimental result of f-measure

	F-Measure (*FM*%)			
	FWKNN	DTL	SVM	HMM
Boiler	95.36	69.42	82.09	74.60
Laptop	75.00	39.22	61.02	53.57
Washing Machine	90.38	57.50	62.65	81.25
Dishwasher	91.36	60.32	68.66	50.85
Kettle	98.53	75.11	90.04	90.04
Fridge	91.20	54.94	63.37	59.28
Average	92.16	61.11	72.22	69.55

for boiler than laptop at all k values. In term of the recall rate, the results are both relatively good as we designed a transition detection algorithm before FWKNN to eliminate redundant calculation. Specifically, the average recall rate for boiler is 89.2% compared to the 78.4% for laptop. It can be concluded from the F-Measure results that FWKNN has relatively better performance and avoids redundant calculation with the algorithm parameter k=5. Hence, the following experiments use this parameter value.

Different Appliance. Figure 9 and Table 3-5 summarize the results of the second experiment. In this study, we compare the algorithm performance for FWKNN and other three benchmark algorithms DTL, SVM and HMM. Figure 10 presents the result for six different appliances in terms of their TP, FP and FN. Similar to the first experiment, this experimental result also reflects that the precision and recall rate results are better for high-power appliance. For example, the boiler and kettle in UK-DALE dataset has the highest detection precision around 97%. They also have the highest recall rate and F-Measure among all the tested appliances. Note that the result of fridge is relatively worse for all four algorithms as it produce more FP and FN than other appliances. Owing to the status of this kind of appliance is actually not simply two status - ON and OFF, if it is using a fixed-frequency compressor, but a multi-level working status. As a result, the amount of false positive and false negative are much higher than other traditional appliances.

Different Algorithm. In the third experiment, when comparing four different algorithms, it can be found that the performance of DTL algorithm is significantly worse than other three algorithms because decision tree has a relatively simple structure, therefore it cannot have high identification accuracy for various different appliances. According to the weighting average (based on the number of classification) of *F-Measure* result, FWKNN has the best performance with 91.08% precision, 93.35% recall rate, and 92.16% F-Measure, followed by SVM and then HMM. Consistent with the previous results, the classification accuracy for low-power appliance is not as good as high-power appliances. For example, the detection precision for laptop are below 70% for all four algorithms. Although FWKNN has the highest experiment result for identifying the activity of laptop, 69.23% precision, 81.82% recall rate, and 75.00% F-Measure. Therefore, although the activity identification accuracy for low-power appliances needs to be improved for FWKNN and other algorithms, FWKNN still outperform other benchmarks in its average performance.

Different Fingerprint. Let A-KNN denote the FWKNN algorithm running without context-based fingerprints and S-KNN denote the native KNN algorithm running on $\nabla(t_i)$. In the last experiment, two different appliances, boiler and laptop are evaluated for precision, recall rate and F-Measure. Figure 10 shows the experimental result. The average precision of reaches 97.3% for boiler and 69.2% for laptop, compared with the 92.5%, 55.5% of A-KNN and 76.5%, 46.5% of native KNN. By comparing the result of FWKNN and A-KNN, it can be seen

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2776300, IEEE Transactions on Industrial Informatics

that the precision, recall rate and F-Measure result of FWKNN algorithm are all significantly higher than A-KNN. It reflects that the introducing of context-based fingerprints into FWKNN effectively improves its performance for both high-power and low-power appliances. In terms of the comparing with native KNN model and FWKNN, it also shows significant advantage in detection accuracy. Therefore, the result proves the advantages of the proposed boosting framework upon multiple appliance-based fingerprints.

VII. Conclusions and Future Works

This paper proposes the concept of appliance fingerprint and a variety of appliance-based and context-based fingerprints. Additionally, the paper proposes a multi-source fingerprint--weighting KNN algorithm - FWKNN, and a boosting framework for continuous online fingerprint learning and detection. A prototype is implemented based on IBM PaaS cloud platform, and evaluate the proposed model and algorithm on UK-DALE dataset. Experimental result and analysis prove that FWKNN outperforms other benchmark method in detection accuracy. Future work of this paper includes the following aspects: (1) propose a multi-state detection model to establish and characterize the multiple states switching of appliance, improving the accuracy of the activity detection algorithm for multi-status appliance; (2) experiments and evaluations of other practical household electricity datasets, analyze the effectiveness of FWKNN algorithm; (3) implement a system for intelligent appliance activity identification and continuous model improvement using smart mobile devices and user responses.

Acknowledgements

This work is supported by National Natural Science Foundation of China (Grant no.61701007), National Key R&D Program of China (Grant no.2017YFB1200700), China Postdoctoral Science Foundation (Grant no.2016M600865) and IBM Shared University Research Project.

References

[1] 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*, 2009, pp. 245-254.

[2] A. Zoha, A. Gluhak, M. A. Imran, and S. Rajasegarar, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors,* vol. 12, no. 12, pp. 16838-16866, 2012.

[3] N. Batra *et al.*, "NILMTK: an open source toolkit for non-intrusive load monitoring," in *Proceedings of the 5th international conference on Future energy systems*, 2014, pp. 265-276.

[4] L. K. Norford and S. B. Leeb, "Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms," *Energy and Buildings,* vol. 24, no. 1, pp. 51-64, 1996.

[5] J. Kelly and W. Knottenbelt, "The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes," *Scientific data,* vol. 2, 2015.

[6] Y. Kim, T. Schmid, Z. M. Charbiwala, J. Friedman, and M. B. Srivastava, "NAWMS: nonintrusive autonomous water monitoring system," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, 2008, pp. 309-322.

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

[8] Y. Du, L. Du, B. Lu, R. Harley, and T. Habetler, "A review of identification and monitoring methods for electric loads in commercial and residential buildings," in *2010 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2010, pp. 4527-4533.

[9] Y. F. Wong, Y. Ahmet Sekercioglu, T. Drummond, and V. S. Wong, "Recent approaches to non-intrusive load monitoring techniques in residential settings," in *IEEE Symposium on Computational Intelligence Applications In Smart Grid (CIASG)*, 2013, pp. 73-79.

[10] 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*, pp. 253-264.

[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*, 2010, pp. 139-148.

[12] H.-H. Chang, "Non-intrusive demand monitoring and load identification for energy management systems based on transient feature analyses," *Energies,* vol. 5, no. 11, pp. 4569-4589, 2012.

[13] M. Figueiredo, A. De Almeida, and B. Ribeiro, "Home electrical signal disaggregation for non-intrusive load monitoring (NILM) systems," *Neurocomputing,* vol. 96, pp. 66-73, 2012.

[14] B. Zhao, L. Stankovic, and V. Stankovic, "On a training-less solution for non-intrusive appliance load monitoring using graph signal processing," *IEEE Access,* vol. 4, pp. 1784-1799, 2016.

[15] A. G. Ruzzelli, C. Nicolas, A. Schoofs, and G. M. O'Hare, "Real-time recognition and profiling of appliances through a single electricity sensor," in *IEEE Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks (SECON),* 2010, pp. 1-9.

[16] L. Jiang, S. Luo, and J. Li, "An approach of household power appliance monitoring based on machine learning," in *International Conference on Intelligent Computation Technology and Automation*, 2012, pp. 577-580.

[17] J. Liao, G. Elafoudi, L. Stankovic, and V. Stankovic, "Non-intrusive appliance load monitoring using low-resolution smart meter data," in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2014, pp. 535-540.

[18] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Using hidden markov models for iterative non-intrusive appliance monitoring," 2011.

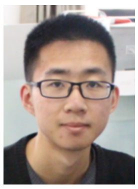
[19] S. Makonin, F. Popowich, I. V. Bajić, B. Gill, and L. Bartram, "Exploiting hmm sparsity to perform online real-time nonintrusive load monitoring," *IEEE Transactions on Smart Grid,* vol. 7, no. 6, pp. 2575-2585, 2016.

[20] O. Parson, S. Ghosh, M. Weal, and A. Rogers, "Non-intrusive load monitoring using prior models of general appliance types," 2012.

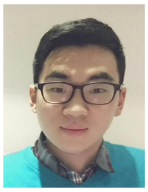
[21] J. M. Gillis, S. M. Alshareef, and W. G. Morsi, "Nonintrusive load monitoring using wavelet design and machine learning," *IEEE Transactions on Smart Grid,* vol. 7, no. 1, pp. 320-328, 2016.

[22] K. Basu, V. Debusschere, S. Bacha, U. Maulik, and S. Bondyopadhyay, "Nonintrusive Load Monitoring: A Temporal Multilabel Classification Approach," *IEEE Transactions on Industrial Informatics,* vol. 11, no. 1, pp. 262-270, 2015.

Meng Ma (S'14-M'16) is a post-doctoral research fellow of School of Electronics Engineering and Computer Science at Peking University, China. He obtained his doctorate degree in Computer Science from Peking University in 2016. His research interests include intelligent computing and sensing, complex event processing, internet of things and semantic computing. He has published 18 papers at venues like *IEEE Internet of Things Journal* and *Knowledge-Based Systems*. One of his papers has received the Best Paper Award from the 2013 *IEEE International Conference on RFID-Technologies and Applications (RFID-TA).*



Weilan Lin received a B.S. degree in software engineering from Beijing Institute of Technology, China, in 2016. He is currently pursuing a master degree in software engineer at Peking University. His research interests include intelligent computing and sensing, knowledge discovery, internet of things and semantic computing.

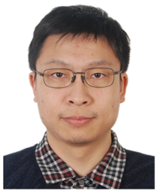


Jingbin Zhang received a B.S. degree in computer science from Beijing Normal University, China, in 2016. He is currently pursuing doctorate degree in computer science at Peking University. His research interests include internet of things, IoT middleware, context-aware computing and service computing.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2776300, IEEE Transactions on Industrial Informatics

Ping Wang (M'88-SM'16) is a professor of National Engineering Research Center for Software Engineering at Peking University, China. He obtained his doctorate degree in Computer Science from the University of Massachusetts in 1996. He is the director of Intelligent Computing and Sensing Laboratory (iCSL), Peking University. Dr. Wang has authored over 50 papers at venues like IEEE TDSC, ACM CCS, IEEE ICWS, IEEE IoT-J. He was the recipient of one Best Paper and two Outstanding Paper Awards from different professional societies.

Yuchen Zhou (S'98-M'00-SM'06) is a Senior Technical Staff Member and the Manager of AI Perception, IBM Research - China. He is leading the research on acoustic analytics and multi-sensor collaborative perception. Before that, he initiated and led the direction of SOA Policy and Cloud Enabled Industry Solutions in IBM SWG SOA Advanced Technologies. Zhou is a senior member of IEEE and ACM, a member of Technical Committee, CCF Embedded System Community, a member of IBM Academy of Technology, and an IBM Master Inventor. He also served as a member of W3C SML WG and TOG SOA Governance WG. Zhou published 1 book and over 20 papers, and filed more than 30 patents.



Xiaoxing Liang (M'16) is a research scientist of IBM China Research Lab, Beijing, China. He obtained his master degree in Computer Science from Beihang University in 2007. His research interests include data analytics for internet of things, big data processing platform, cloud based services. He has led the design and implementation of several innovative projects in IBM China Development Lab, Baidu, IBM China Research Lab. He has published 7 papers at venues like IEEE ICWS, IEEE ICCC, IEEE UIC, IEEE Cloud, IEEE SERVICES.