

Learning from Non-Experts: An Interactive and Adaptive Learning Approach for Appliance Recognition in Smart Homes

JACKSON CODISPOTI, University of Kentucky, USA

ATIEH R. KHAMESI, University of Kentucky Learning, USA

NELSON PENN, University of Kentucky, USA

SIMONE SILVESTRI, University of Kentucky, USA

EURA SHIN, University of Kentucky, USA

With the acceleration of ICT technologies and the Internet of Things (IoT) paradigm, *smart residential environments*, also known as *smart homes* are becoming increasingly common. These environments have significant potential for the development of intelligent energy management systems, and have therefore attracted significant attention from both academia and industry. An enabling building block for these systems is the ability of obtaining energy consumption at the appliance-level. This information is usually inferred from electric signals data (e.g., current) collected by a smart meter or a smart outlet, a problem known as *appliance recognition*. Several previous approaches for appliance recognition have proposed load disaggregation techniques for smart meter data. However, these approaches are often very inaccurate for low consumption and multi-state appliances. Recently, Machine Learning (ML) techniques have been proposed for appliance recognition. These approaches are mainly based on passive MLs, thus requiring pre-labeled data to be trained. This makes such approaches unable to rapidly adapt to the constantly changing availability and heterogeneity of appliances on the market. In a home setting scenario, it is natural to consider the involvement of users in the labeling process, as appliances' electric signatures are collected. This type of learning falls into the category of *Stream-based Active Learning* (SAL). SAL has been mainly investigated assuming the presence of an *expert*, always available and willing to label the collected samples. Nevertheless, a home user may lack such availability, and in general present a more erratic and user-dependent behavior. In this paper, we develop a SAL algorithm, called *K-Active-Neighbors* (KAN), for the problem of household appliance recognition. Differently from previous approaches, KAN jointly learns the user behavior and the appliance signatures. KAN dynamically adjusts the querying strategy to increase accuracy by considering the user availability as well as the quality of the collected signatures. Such quality is defined as a combination of *informativeness*, *representativeness*, and *confidence score* of the signature compared to the current knowledge. To test KAN versus state-of-the-art approaches, we use real appliance data collected by a low-cost Arduino-based smart outlet as well as the ECO smart home dataset. Furthermore, we use a real dataset to model user behavior. Results show that KAN is able to achieve high accuracy with minimal data, i.e., signatures of short length and collected at low frequency.

Additional Key Words and Phrases: Appliance Recognition, Stream Based Active Learning, User Behavior.

Authors' addresses: Jackson Codispoti, University of Kentucky, Lexington, KY, USA, Jackson.Codispoti@uky.edu; Atieh R. Khamesi, University of Kentucky, Lexington, KY Learning, USA, atieh.khamesi@uky.edu; Nelson Penn, University of Kentucky, Lexington, KY, USA, nape226@g.uky.edu; Simone Silvestri, University of Kentucky, Lexington, KY, USA, silvestri@cs.uky.edu; Eura Shin, University of Kentucky, Lexington, KY, USA, eura.shin@uky.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2022 Association for Computing Machinery.
2378-962X/2022/2-ART \$15.00
<https://doi.org/10.1145/3491241>

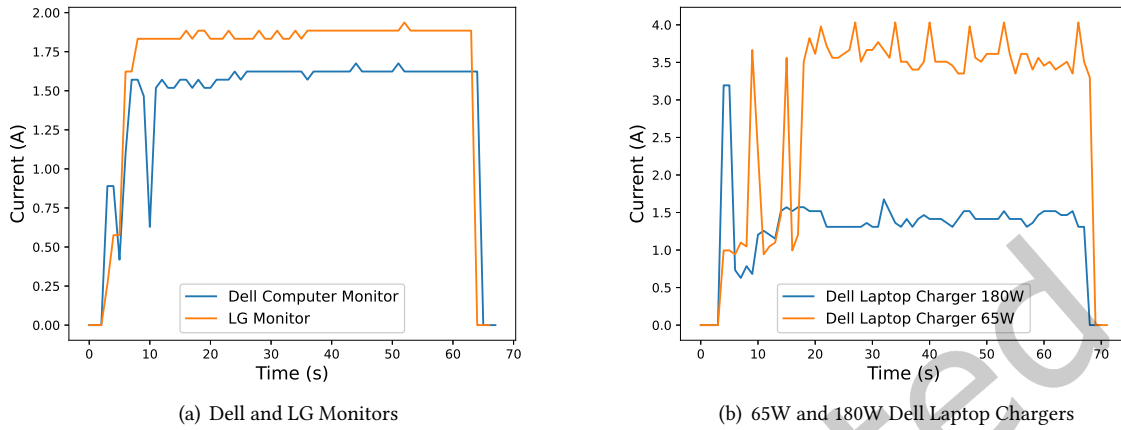


Fig. 1. Signatures comparison between same appliances from different brands

1 INTRODUCTION

In the United States, the residential sector is responsible for more than 20% of the total energy consumption [29], and this amount has been constantly increasing for several decades. As an example, in 2015, over 2.6 trillion kilowatt-hours were consumed [1], which are expected to increase by 13.5% by 2040 according to the U.S. Department of Energy [3]. Some promising approaches introduced to manage energy consumption at the residential level are obtained through fine-grained electricity billing, demand-response programs, and electrical power load balancing [28, 32]. A well-recognized key enabler for such energy management techniques is the knowledge about when each appliance is used and its consumption [5, 9, 17, 22, 27, 38, 39]. The problem of recognizing an appliance from its electric signature is known as *appliance recognition*. Several previous approaches in this context rely on *smart meters*. These meters measure the aggregated energy consumption of all home appliances and communicate such information to a utility company for billing [45]. To infer individual appliance consumption, load disaggregation techniques are used [47]. However, these techniques lack the sufficient accuracy to enable fine-grained energy management, and particularly inaccurate for low-consumption appliances [47].

Smart outlets are a promising alternative for the appliance recognition problem. These outlets look like traditional wall plugs, but are actually Internet of Things (IoT) devices [24], with the capability of monitoring and controlling the power usage of a connected electric appliance. Previous studies on appliance recognition using smart outlets adopt *passive machine learning*, in which a set of labeled data is given to train the model before it is deployed, and then the fixed trained model is used for classification [2]. This approach is not practical in a smart home for several reasons [38]: (i) offline classification is not flexible to new appliances subsequently available on the market; (ii) similar appliances (e.g., different brands) may have very different consumption patterns, making it hard to perform offline training. To illustrate this point, we collected the signatures (patterns of current over time) of two computer monitors, Fig. 1(a), and two laptop chargers, Fig. 1(b). Clearly, training a machine learning model on one monitor (or charger), and then using the trained model to recognize the other monitor (or the other charger) at a later stage (e.g., once the new appliance is available on the market) may result in misclassification. This problem is exacerbated by the high rate of new appliances available on the market.

In a home setting scenario, it is natural to consider the involvement of users in the labeling process, as appliance signatures are collected. This would allow to dynamically train a model tailored to the appliances in each individual home, overcoming the limitations of offline machine learning. This type of learning falls into the category of *Stream-based Active Learning* (SAL) [13]. Differently from *active learning*, where it is assumed that the entire dataset is available during training and an expert user labels a selected subset of instances, SAL assumes that data is generated over time, when an appliance is plugged in, and the system must decide if manual labeling from the user is needed. However, SAL learning has been previously considered only in highly specialized sectors (e.g., image and text recognition) where the user is an *expert* naturally incentivized to participate in the labeling process [36]. We point out that this is not the case in most of the smart home applications, where a user may have a different level of engagement and availability over time. In fact, a home user may present a more erratic and user-dependent behavior, and the number of queries should be limited to prevent overwhelming them. As an example, a recent dataset of Mobile Phone Use (MPU) [25], that we use in this paper to model real user behavior, shows that the users rarely interact with such systems. As a matter of fact, in most cases users respond to the system notifications less than 50% of the time. To the best of our knowledge the problem of user abstention for interactive querying has only been addressed in [35, 43]. These works assume that user responds to queries uniformly at random or that user abstention is primarily influenced by proximity to the decision boundary, which is not a realistic assumption for labeling tasks in a smart home.

In this paper, we propose an *interactive* and *adaptive* SAL approach, called *K-Active-Neighbors* (KAN). We assume that a smart outlet generates a *signature* when an appliance is plugged in and turned ON. Through a smart phone app, a user may be requested to label such a signature. The main challenge addressed by KAN is to jointly learn the user *engagement* and the appliance signatures. To this purpose, KAN provides an *adaptive querying strategy* to determine whether to query the user or not. Such strategy takes into account the *quality* of the signature over time and the user *engagement* with the system. KAN estimates the quality of a signature through three metrics *informativeness* and *representativeness*, calculated on a *K*-Nearest-Neighbor graph, and *confidence score*. Simultaneously, KAN keeps track of the user previous engagement over time (i.e., whether the user responded to the labeling request), in order to infer the likelihood of the user labeling a sample at the current time slot. Since, as the system learns, signatures of high quality tend to become more rare, KAN uses a statistical approach to adaptively adjust its *querying strategy*. This strategy is based on a dynamic tradeoff between quality of a signature and likelihood of user response.

This paper significantly advances commercial solutions as well as state-of-the-art approaches. For instance, Currant [7], a recent commercial smart outlet, is only able to track outlet usage over time, without performing appliance recognition. Additionally, learning from non-expert users introduces specific challenges such as the need of learning quickly, with a small amount of sampled data, and with potential limited engagement of the user in the labeling process. In this paper, we show that even a recently proposed approach based on convolutional neural networks [44] fails in this challenging scenario.

We perform extensive experiments to test the performance of KAN versus a state-of-the-art approach. To this purpose, we use two datasets of electric signatures. The first dataset is obtained by an Arduino-based smart outlet developed as a part of this paper. The second dataset is from the ECO project [41], which contains signatures collected from a real world smart home. We also model the user behavior realistically by adopting the MPU dataset [25]. Results show that KAN outperforms other approaches, and it is able to achieve high accuracy with minimal data. This makes it particularly suitable to low-cost smart outlets and IoT applications, where resources are limited. In addition, an analysis of the MPU dataset shows that users rarely interact with such systems. Therefore, it is of primary importance to propose an adaptive querying strategy in order to optimize the whole learning process.

In summary, the main contributions of this paper are:

- We propose an algorithm called KAN for appliance recognition in smart homes;
- KAN exploits a notion of quality of a signature based on informativeness, representativeness, and confidence scores;
- KAN learns the user behavior over time to infer the likelihood of the user engaging in the labeling process;
- KAN combines the signature quality with the predicted user behavior to determine an adaptive querying strategy;
- We test KAN on real appliance signatures collected from an Arduino-based smart outlet along with a house from the ECO dataset [41], and use the MPU dataset to realistically model user behaviour ;
- Results show that our approach outperforms state-of-the-art approaches. In addition, KAN is able to achieve high accuracy with minimal data, i.e., signatures of short length and collected at low frequency.

The rest of the paper is organized as follows. In Section 2, we describe the system model and then elaborate on the problem statement. The proposed solution and the experimental results are detailed in Sections 3 and 4, respectively. Section 5 investigates the related works. Finally, Section 6 concludes the paper.

2 SYSTEM MODEL AND PROBLEM STATEMENT

In this paper we consider a *smart residential environment*, also referred to as a *smart home*, equipped with smart outlets with minimal hardware costs. Such outlets are characterized by a relatively low-frequency sampling rate (1-28 Hz) and only collects electric current information. As part of this paper, we develop an Arduino-based smart outlet with these characteristics. The details of the outlet are provided in Section 4.1.1. The outlets are connected to the home energy management system through a wireless interface. The system is also interfaced with a *smart phone app*, which is used to send notifications to the user and to request the user to label data.

When an appliance is plugged into the outlet and turned ON, the electric current data obtained by the outlet results in a time series of RMS amperage values, for a specific duration, e.g., 30 seconds. Such time series defines a *signature*. The n -th collected signature is stored as a vector of current values, denoted by \mathbf{x}_n . Note that, two signatures may be of different lengths, depending on the duration the device is used. We refer to X as the set of signatures observed by the smart outlet. X grows over time as new signatures arrive.

We assume that time is divided into time slots, for example corresponding to the 24 hours per day, denoted as $h = 0, \dots, 23$. As a labeler, the user is assumed to follow a certain response distribution according to his willingness/availability to interact with the system, i.e., respond to the app notifications. In order to model user behavior, we use a Bernoulli distribution at each time slot [40]. As a result, we refer to $H(h)$ as the probability that the user will successfully respond to a query at time slot h . We validate this assumption using real data from the MPU dataset in Section 4.3.1. We refer to D as the number of days during which the framework operates, and to B as the maximum number of queries that can be asked in each day. Budget B and days D are system parameters, and the impact of these values on classification performance is explored in Section 4.3.2.

As commonly described in literature, home electric appliances may be classified according to their working style as follows. Appliances that only achieve an ON or OFF state are type I; multi-state appliances are known as type II; and continuously variable appliances, for which the energy demand fluctuates with the activity of the machine are type III. Generally, type II and III appliances are considered the most challenging by NILM algorithms, as an appliance that operates in multiple states (a hair dryer with a high/medium/low setting) may posses many unique signatures, and thus have a higher chance of misclassification [47]. In this paper, we explicitly consider the problem of signatures of different lengths, by exploiting the Dynamic Time Warping algorithm. Moreover, unlike the NILM algorithms, our proposed model is able to classify appliances with low consumption and multiple working states.

Problem Statement: Without loss of generality, we assume that appliance signatures are generated sequentially over time. Thus, upon the arrival of a signature, \mathbf{x}_n , the system must decide whether to query the user for

the corresponding class label. Intuitively, an upcoming signature should be queried if it represents (i) a new appliance, i.e., it does not match any existing class label, or (ii) similar to an already observed signature but with low confidence.

On the other hand, to avoid overwhelming the user with too many queries, the system is not allowed to exceed the budget of B queries per day. Therefore, the problem is to find an effective querying strategy to (i) maximize the accuracy of recognizing appliances by querying signatures that are most informative; (ii) learn the user response distribution model, and use this distribution to optimize the querying strategy; and (iii) do not exceed the maximum budget constraint.

3 PROPOSED SOLUTION

In this section, we describe the K -Active-Neighbors (KAN) algorithm. KAN adopts the Dynamic Time Warping (DTW) metric to measure the distance between signatures, thus we first introduce this technique.

3.1 Dynamic Time Warping

DTW is a distance measure between two different time series of potentially different length [16]. It has been used in several fields including medicine, industry and finance. Unlike Euclidean distance, DTW exploits dynamic programming to find the optimal alignment between two temporal sequences. This approach minimizes the alignment cost, and returns optimal distance between two sequences of varying length. DTW is an appropriate choice in a smart home scenario because two signatures may likely have different lengths, depending on duration of use. DTW has been used as a passive machine learning classifier in [23]. In this paper, we adopt DTW as a means of constructing a distance matrix to guide the proposed stream-based active learning processes.

According to our experiments, DTW is better able to exploit the natural grouping structure of signatures from the same device. Therefore, the remaining solutions described in this paper all use DTW as the distance between two signatures, say \mathbf{x}_a and \mathbf{x}_b as $DTW(\mathbf{x}_a, \mathbf{x}_b)$. We refer the reader to [34] for more details about DTW.

3.2 K -Active-Neighbors Approach

We propose a K -Nearest-Neighbors-based active learning algorithm, named K -Active-Neighbors (KAN), as the querying strategy. KAN makes use of the *informativeness*, *representativeness*, and *confidence score* of incoming signatures, paired with the observed distribution of user engagement, $H()$, to determine such strategy.

3.2.1 Informativeness and Representativeness. The informativeness of a new sample (i.e., signature) represents the ability of that sample to reduce the amount of error in the classifier through the introduction of needed information. Conversely, representativeness indicates how valuable a sample is in reflecting the underlying structure of the data [10].

Consider an incoming event signature, \mathbf{x}_n , and the set of instances already observed by the smart outlet, $X = \{\mathbf{x}_1, \dots, \mathbf{x}_{n-1}\}$. We let $KNN(\mathbf{x}_n) = \{\mathbf{x}_1^{(n)}, \dots, \mathbf{x}_K^{(n)}\}$ represent the K nearest neighbors of \mathbf{x}_n in X , where DTW is used as the distance metric. The K -Nearest-Neighbors algorithm determines “who affects who” by defining relationships between samples. This may be conceptualized as a directed graph, where an edge is drawn from sample \mathbf{x}_b to sample \mathbf{x}_a only if $\mathbf{x}_a \in KNN(\mathbf{x}_b)$. This concept is demonstrated by a simple example in Fig. 2, where circles depict the already observed instances by the smart outlet, X , and \mathbf{x}_n is represented by the green and blue rectangle in Figs. 2(a) and 2(b), respectively. Moreover, filled and hollow circles illustrate the labeled and unlabeled samples, respectively. In these graphs, directed edges are drawn using the KNN definition, i.e., each sample has exactly K outdegree toward its K nearest neighbors. Accordingly, affecting and non-affecting neighboring relationships for the sample of interest are shown by solid and gray edges, respectively.

Hence, an instance \mathbf{x}_n is *representative* if it receives very few edges overall, meaning it explores a new part of the feature space. We define the number of instances a sample affects, or represents, as $N_R(\mathbf{x}_n)$. This number

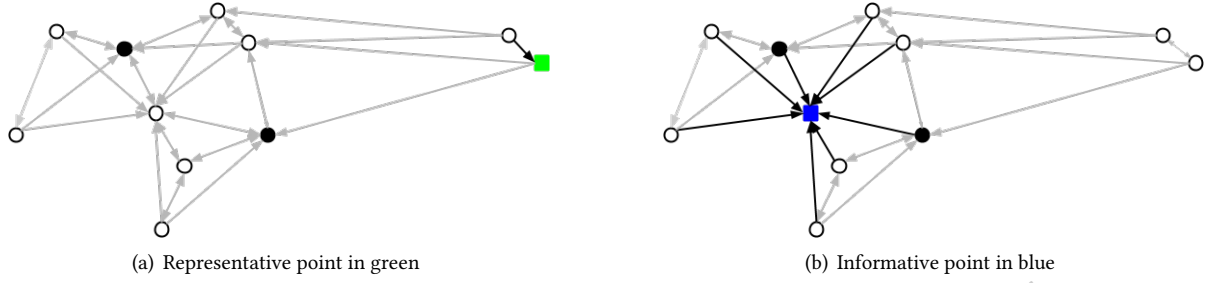


Fig. 2. Example of representative and informative samples

represents the total number of edges \mathbf{x}_n receives, which is also the total number of times $\mathbf{x}_n \in \text{KNN}(\mathbf{x}_i)$ for all $\mathbf{x}_i \in X$. This is formally defined as

$$N_R(\mathbf{x}_n) = \sum_{i=1}^{|X|} \mathbf{1}(\mathbf{x}_n \in \text{KNN}(\mathbf{x}_i)), \quad (1)$$

where, $\mathbf{1}(\cdot)$ represents the indicator function. Fig. 2(a) is an example of a representative sample in green which receives only one edge, meaning $N_R(\mathbf{x}_{\text{green}}) = 1$. This sample “neighbors” very few points, therefore representing an unexplored area of the feature space.

Moreover, \mathbf{x}_n is considered *informative* if it has many incoming edges from *unlabeled* samples. The labeled state of a sample $\mathbf{x}_i \in X$ is denoted by the Boolean function $l_s(\mathbf{x}_i)$. The number of unlabeled instances a sample affects is denoted by

$$N_I(\mathbf{x}_n) = \sum_{i=1}^{|X|} \mathbf{1}(\mathbf{x}_n \in \text{KNN}(\mathbf{x}_i) \ \& \ \neg l_s(\mathbf{x}_i)). \quad (2)$$

The definition N_I considers how many of the received edges come from samples with unknown labels. Thus according to Fig. 2(b), we have $N_I(\mathbf{x}_{\text{blue}}) = 6$.

Consider the set X . We can sort the signatures in X by their score $N_I(\mathbf{x}_i)$ and define the *informativeness score* $I(\mathbf{x}_n) \in [0, 1]$ of the new signature \mathbf{x}_n as the percentile rank (i.e., the normalized position) of $N_I(\mathbf{x}_n)$ in the sorted set. This metric indicates how informative a signature is compared to all of the observed signatures. For example, a new signature that is close to many unlabeled instances, and falls in the 90-th percentile in number of unlabeled neighbors, may be considered *very informative*. In a similar way, we evaluate the representativeness score $R(\mathbf{x}_n)$. By definition, a representative instance is one with few incoming edges. Therefore, we desire the percentile rank of $N_R(\mathbf{x}_n)$ with respect to the frequency distribution of the remaining $\{N_R(\mathbf{x}_i) : \forall \mathbf{x}_i \in X\}$ to be small, i.e., receive few edges. The metric $R(\mathbf{x}_n)$ is defined by the reverse percentile rank, which is the complement of the percentile rank of $N_R(\mathbf{x}_n)$.

We use the above informativeness and representativeness scores to derive the quality of a signature. Nevertheless, such scores are not the only factor in determining the querying decision. In addition to the budget, we consider a confidence score as well as the willingness of the user to answer that query at the current time slot, as explained in the following sections.

3.2.2 Confidence Score. Although the informativeness and representativeness scores give insight about how useful a signature is relative to the topology of the KNN graph, i.e., the feature space, they lack to determine the likelihood of a signature belonging to a certain class label. In other words, the informativeness score $I()$ and the representativeness score $R()$ deal with the relative position of the incoming signature in connection with the

labeled and unlabeled samples, regardless of their class labels. To fill this gap, in this subsection we introduce the confidence score which measures the chance of a signature being of a certain class label among its K nearest neighbors.

Inspired by [4], in this work we define the confidence score as follows. Let $l(\mathbf{x}_i)$ represent the class label of signature $\mathbf{x}_i \in X$ and recall that $\text{KNN}(\mathbf{x}_n) = \{\mathbf{x}_1^{(n)}, \dots, \mathbf{x}_K^{(n)}\}$ is the set of K nearest neighbors of \mathbf{x}_n . We define $\mathcal{A}_{\mathbf{x}_n} = \{l(\mathbf{x}_i^{(n)}) : \forall \mathbf{x}_i^{(n)} \in \text{KNN}(\mathbf{x}_n)\}$ as the set of class labels of the nodes in $\text{KNN}(\mathbf{x}_n)$. The likelihood of signature \mathbf{x}_n belonging to class label a is defined as

$$\mathcal{L}_{\mathbf{x}_n}(a) = \frac{\sum_{i=1}^K \mathbf{1}(l(\mathbf{x}_i^{(n)}) = a) \cdot \frac{1}{\text{DTW}(\mathbf{x}_n, \mathbf{x}_i^{(n)})}}{\sum_{i=1}^K \frac{1}{\text{DTW}(\mathbf{x}_n, \mathbf{x}_i^{(n)})}}, \quad (3)$$

where $\mathbf{1}(\cdot)$ is the indicator function, as mentioned before. Finally, the confidence score of signature \mathbf{x}_n is given by

$$C(\mathbf{x}_n) = \max_{a \in \mathcal{A}_{\mathbf{x}_n}} \mathcal{L}_{\mathbf{x}_n}(a) \quad (4)$$

The confidence score is used in the KAN querying strategy detailed in Section 3.2.4. In the next section we discuss how to learn the user response distribution, which is also integrated in such strategy.

3.2.3 Learning the user response distribution. The KAN algorithm considers the *time* to query the labeler by learning the user response distribution. We recall that in this paper we assume that at each time slot h the user responds to a query with an independent probability $H(h)$. In this section we discuss how this probability is inferred. When a query is sent to the user, there are two possible outcomes, a success (the user labels the data) and a loss (the user ignores the system query). Let $s(h)$ represent the number of successes at time slot h , and let $n(h)$ represent the total number of queries submitted at that time slot. Thus, $H(h)$ is given by,

$$H(h) = \begin{cases} \alpha, & \text{if } n(h) \leq \epsilon_1 \\ \beta, & \text{if } s(h) \leq \epsilon_2 \\ \frac{s(h)}{n(h)}, & \text{otherwise} \end{cases} \quad (5)$$

where α , β , ϵ_1 , and ϵ_2 are initial values introduced to ensure that the empirical probability $\frac{s(h)}{n(h)}$ is considered only after a time slot has been tested a sufficient number of times. This ensures that time slots tested a low number of times have a non negligible probability of being used. The parameters ϵ_1 and ϵ_2 may be derived using any strategy for selecting a sample size to estimate a population mean [14].

3.2.4 Querying Strategy. An efficient querying strategy needs to determine the best time slots throughout the day when (i) the incoming signature is potentially able to improve the accuracy of model and (ii) the user is most likely to respond to the queries in those time slots. Note that, this is not a trivial task. In fact, the amount of potentially good signatures decreases naturally as the system learns more. Therefore, at the beginning most signatures are relevant, and thus it is obviously best to query during time slots with high response probability. Conversely, at later stages good signatures are rare, and thus it may be better to query even if the chance of getting a response is low. For these reasons, we design an adaptive strategy that changes as the system learns over time.

According to our strategy, we first obtain the quality of the signature using its informativeness, representativeness, and confidence scores described in Subsections 3.2.1 and 3.2.2, respectively.

Formally, we define the quality of signature \mathbf{x}_n , denoted by $Q(\mathbf{x}_n)$, as follows.

$$Q(\mathbf{x}_n) = I(\mathbf{x}_n) * R(\mathbf{x}_n) * C(\mathbf{x}_n) \quad (6)$$

where $I(\mathbf{x}_n)$, $R(\mathbf{x}_n)$, and $C(\mathbf{x}_n)$ are the informativeness, representativeness, and confidence scores of signature \mathbf{x}_n , respectively.

Given the remaining budget b , and the quality of the signature \mathbf{x}_n , KAN adaptively decides whether to query the user. This is achieved by predicting whether the incoming signature has better quality and a higher chance of being successfully labeled, compared to the remaining time slots of the day. In order to overcome the changing quality of signatures as the system learns, we calculate the probability that \mathbf{x}_n has higher quality than a randomly sampled signature \mathbf{y} , that is $F_y(\mathbf{x}_n) = P(Q(\mathbf{y}) < Q(\mathbf{x}_n))$. This probability is calculated over a moving window of few days, in order to keep it updated with the changing trends of quality scores.

Accordingly, by assuming that signature qualities are independent on each other, the probability that at least one signature of better quality than \mathbf{x}_n will arrive during a time slot with m signatures is given by

$$A(\mathbf{x}_n, m) = 1 - (F_y(\mathbf{x}_n))^m \quad (7)$$

Thus, we calculate the expected number $N(\mathbf{x}_n, h, m)$ of successfully labelled signatures, with better quality than \mathbf{x}_n , for the remaining time slots of the day after the current time slot h as follows.

$$N(\mathbf{x}_n, h, m) = A(\mathbf{x}_n, m) \times \sum_{i=h+1}^{23} H(i) \quad (8)$$

Note that $H(i)$ is the probability a user responds during the time slot i as described in Eq. (5). In the following, Algorithm 1 summarizes the querying strategy used in KAN.

ALGORITHM 1: Adaptive Querying Strategy Algorithm

Input : Incoming signature \mathbf{x}_n , informativeness score $I(\mathbf{x}_n)$, representativeness score $R(\mathbf{x}_n)$, confidence score $C(\mathbf{x}_n)$, time slot h , remaining budget b , and the number of signatures expected at each time slot of the day m .

- 1 Find $Q(\mathbf{x}_n)$ using Eq. (6)
- 2 Find $N(\mathbf{x}_n, h, m)$ using Eq. (8)
- 3 **if** $N(\mathbf{x}_n, h, m) < b$ **then**
- 4 Query the user
- 5 **else**
- 6 Don't query the user
- 7 **end**

3.2.5 Pseudocode.

The pseudocode of KAN is provided in Algorithm 2. In the code, b refers to the remaining budget, or equivalent queries, for the current day. This value is reset to $b = B$ at the beginning of each day. Lines 2 and 3 of the pseudocode use the definitions of N_R , N_I to respectively derive the number of representative and informative edges for the signature \mathbf{x}_n . Then, the confidence score is obtained in line 4. We initially set the labeled state of \mathbf{x}_n to false in line 5. In line 6, the adaptive querying strategy described in Algorithm 1 is called. Thus, as a result of Algorithm 1, whether the user responds, or not, the label is assigned (line (8)), and, $s(h)$, $n(h)$, b and $l_s(\mathbf{x}_n)$ are updated accordingly (lines 7 – 17). Finally, \mathbf{x}_n is added to the current set of signatures (line 18).

THEOREM 3.1. *The complexity of KAN is $O(n^2K)$, where $n = |X|$ is the number of already observed signatures by the smart outlet, and K is the number of active neighbors.*

PROOF. The KAN algorithm is called upon the arrival of a new sample (i.e., signature) \mathbf{x}_n . KAN calculates the pairwise DTW distances between \mathbf{x}_n and the n existing samples in X . The complexity of computing DTW is $O(S_d^2)$, where S_d represents the average signature duration [34]. Thus, the complexity of finding pairwise DTW distances between n pairs is $O(nS_d^2)$. Furthermore, for each sample in X , we keep track of its K nearest neighbors in the form of a data structure, e.g., a list or an array. Thus, when \mathbf{x}_n arrives, the time complexity to update the sets of the K nearest neighbors of all n samples is $O(nK)$.

To obtain the informativeness and representativeness scores, for each signature in X , we first create (or update) a list of labeled and unlabeled samples, such that \mathbf{x}_n falls into their K nearest neighbors. This can be completed in $O(n^2K)$. Next, we need to calculate the 90-th percentile of the lengths of such lists, which requires to sort them by length. This incurs in a complexity of $O(n \log n)$.

The complexity of computing the confidence score is the following. Given the pairwise DTW distances and the lists of K nearest neighbors, calculating the likelihood of the new signature \mathbf{x}_n belonging to a specific class label (appliance) is $O(K)$. This should be calculated for all the appliances in the dataset, denote by n_A , to obtain the maximum likelihood. Therefore, the complexity of the confidence score is $O(n_A K) + O(n_A) = O(n_A K)$.

Finally, in line 6 of KAN, Alg. 1 is called. Given $I(\mathbf{x}_n)$, $R(\mathbf{x}_n)$ and $C(\mathbf{x}_n)$, the complexity of Alg. 1 is $O(T)$, where T is the number of time slots in a day, which is usually a small constant (e.g., 24).

Therefore, the total complexity of KAN is $O(nS_d^2) + O(n^2K) + O(n \log n) + O(n_A K) + O(T)$. Since the average signature duration, S_d , and number of time slots per day, T , are significantly smaller than the number of observed signatures, $n = |X|$, the overall complexity is $O(n^2K)$. \square

3.3 Classification

When a sample is not explicitly labeled by the user, either because the user was not queried or because the user did not respond to a query, the label of that sample is estimated using a K -Nearest-Neighbour (KNN) graph

ALGORITHM 2: K -Active-Neighbors Algorithm

Input : Incoming signature \mathbf{x}_n , time slot h , budget B , remaining budget b , $\text{KNN}(\mathbf{x}_n)$.

```

1 if  $b = 0$  then return;
2 Find  $N_R(\mathbf{x}_n)$  to calculate  $R(\mathbf{x}_n)$ .
3 Find  $N_I(\mathbf{x}_n)$  to calculate  $I(\mathbf{x}_n)$ .
4 Find  $C(\mathbf{x}_n)$  using Eq. (4).
5  $l_s(\mathbf{x}_n) = \text{False}$ 
6 Query the user according to Algorithm 1
7 if user has been queried and query is successful then
8   Assign  $l(\mathbf{x}_n)$  according to the user response
9    $s(h)++$ 
10   $n(h)++$ 
11   $b--$ 
12   $l_s(\mathbf{x}_n) = \text{True}$  // labeled sample
13 end
14 if user has been queried and query is not successful then
15    $n(h)++$ 
16    $b--$ 
17 end
18  $X = X \cup \{\mathbf{x}_n\}$ 

```

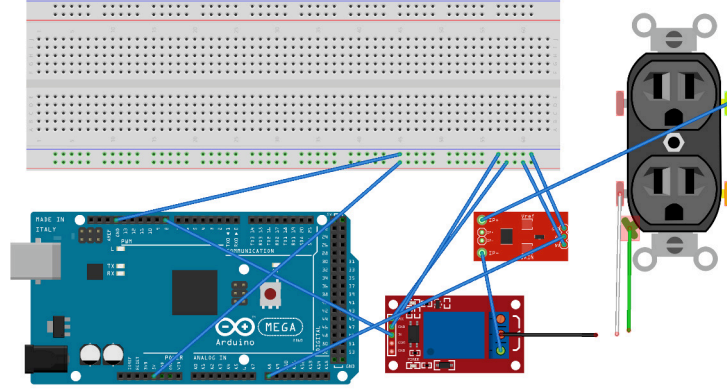


Fig. 3. Schematic of the Arduino-based Smart Outlet

extended with the confidence score and the DTW distance. We adopt the KNN classifier because it is a natural extension of the KAN algorithm, which is designed to choose the most useful “neighbors” in determining a sample’s class label. Like the original KNN algorithm, the confidence score-based DTW-KNN is a lazy algorithm which requires no official training period. This approach is well suited for a dynamic household in which the dataset is constantly evolving.

According to our approach, a signature is assigned the class label given by the label with the highest confidence score among the class labels of its K nearest neighbors. In consistence with the notation introduced in the previous section, signature \mathbf{x}_n is classified as

$$l(\mathbf{x}_n) = \arg \max_{a \in \mathcal{A}_{\mathbf{x}_n}} \mathcal{L}_{\mathbf{x}_n}(a). \quad (9)$$

4 EXPERIMENTAL RESULTS

4.1 Experimental Setup

In this section we provide a comprehensive experimental evaluation of our approach, in comparison with the state-of-the-art solutions. Before showing the results, we describe the smart outlet developed as part of this paper, the appliance electrical signature dataset collected with the outlet, the dataset used to model the user behavior, and the comparison approaches.

4.1.1 Smart Outlet. We developed a low-cost Arduino-based smart outlet, which can be configured to record the current from 1Hz up to 28Hz, and saves this information on a remote server through a wireless interface. An Arduino with a Yún Shield, ACS712 current sensor, and a 5V relay switch was integrated with a regular electrical outlet, as shown in Fig. 3. This platform has been chosen since it is open-source, has a wide array of “shields” that provide enhanced capabilities, has a large resource library, and is very flexible and easy to use. The Yún Shield is the component that allows wireless communication to and from the Arduino, the ACS712 sensor outputs an analog signal proportional to the RMS current of the circuit, and the relay switch simply allows the outlet to be turned on or off autonomously.

4.1.2 Electric Signature Datasets . We use our smart outlet to collect a total of 1570 low-frequency *signatures* sampled at 1 second intervals. This dataset includes 9 different appliances reported in Table 1. We choose these

Table 1. Appliances in the smart outlet dataset

Appliance	Number of Samples	
	Low Freq. (1Hz)	High Freq. (10Hz)
Conair Hair Dryer	83	-
3Com Network Switch	93	108
180W Dell Laptop Charger	112	103
Dell Monitor	184	-
65W Dell Laptop Charger	192	79
Vornado 18W Fan	207	101
1500W Space Heater	211	111
LG Monitor	223	-
Lamp	265	149

Table 2. Appliances in the ECO dataset

Appliance	Number of Samples
Refrigerator	400
Dryer	90
Coffee Machine	400
Kettle	400
Washing Machine	400
PC	112
Freezer	400

appliances because they are commonly available appliances, representative of most households. Besides, the selected appliances test our solution's stability, as they cover challenging characteristics such as low consumption, multi-state, and continuously variable appliances. Additionally, we also collected a total of 651 high-frequency signatures sampled at frequency of 10Hz. These are used to conduct an experiment to evaluate the impact of the signature resolution, i.e., sampling frequency, on the accuracy of our proposed approach.

During data acquisition, each appliance is plugged in the smart outlet and turned ON for roughly 30 seconds before being turned OFF for 20 seconds to allow the current to stabilize. Subsequently, the next signature is collected repeating the above process.

Given the set of appliances' signatures, we generate a *usage schedule*, i.e., the sequence of signatures generated in the smart home, as follows. Without loss of generality, we assume that one appliance signature is observed by the system every time slot. Accordingly, there are $N_s = 24$ signatures observed within a day. Each signature is sampled with replacement from the entire set of signatures. We consider a period of $D = 21$ days.

We also test KAN using signatures from the first house of the ECO dataset [41]. Specifically, the data includes 7 different appliances sampled at 1 Hz, which we list in Table 2, for a total of 2202 signatures. These appliances are found in nearly all homes, have unique usage patterns, and consist of low and high consumption devices alike.

4.1.3 MPU Dataset. To obtain a realistic user behavior model for the user engagement with the querying systems, we use the Mobile Phone Use (MPU) [25] dataset. This dataset is intended to help study patterns of user engagement. The dataset is collected in 4 weeks, on average, for 337 users. It adopts a mood questionnaire that sends notifications to the user 10 to 15 times a day to see when users were most receptive to interacting with

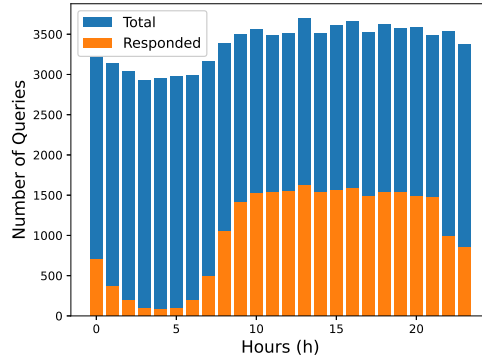


Fig. 4. Total number of queries vs. number of responded ones over all users in MPU

content. In addition to information about user interaction with the mood questionnaire, this dataset also contains the users' mobile phones' information such as screen orientation, battery charge, system notifications, and even the ambient noise level.

In this paper, we focused on the responses to the notifications. Specifically, we are interested in the hours when a notification is posted, and whether or not the user responds to it. We use "Esm_Posted" and "Esm_Timedout" fields of the dataset for our purpose. Finally, we only focus on weekday data to train and test our user response distribution model, since users usually have different response patterns on weekends¹. The response patterns are collected in a time slot by time slot basis and further grouped into days.

Fig. 4 shows the total number of queries and the successfully responded ones from the MPU dataset for all users. Overall, only 30.8% of all queries are successful. This shows the importance of designing an adaptive querying strategy that learns the user behavior and jointly the appliance signatures.

4.1.4 KAN parameters. We set the KAN parameter K , defined in Algorithm 2, equal to 3. We also set $\alpha = \frac{1}{2}$, $\beta = \frac{1}{24}$ in Eq. (5). The chosen α value represents a completely random probability of success, β is the uniform probability a user will respond in any of the time slots of the day, and $\epsilon_1, \epsilon_2 = 0$. We performed a sensitivity analysis to the setting of the parameters, and observed similar performance trends.

4.2 Comparison Approaches

We compare KAN to three different SAL strategies proposed in [26], [38], and [44], respectively.

4.2.1 OBAL and SVM Classifier. We compare KAN with the solution proposed in [26], called Online Batch-based Active Learning (OBAL) algorithm, paired with the Support Vector Machine (SVM) classifier. This algorithm is designed for the identification of relevant textual data from social media. We modified OBAL in order to apply it to the appliance recognition problem.

Similar to our proposed approach, OBAL has a labeling budget and interactively queries the user to obtain labels. OBAL adopts a set of uncertainty strategies to identify items for labeling. To this aim, it exploits the boundary instances, i.e., those that lie close to the classification boundary, for which OBAL is highly uncertain about their class labels. The algorithm decides which successfully labeled samples are boundary vectors through

¹Our approach could be easily modified to learn two different user response distributions, representing week days and weekends, respectively. The querying strategy could use the appropriate distribution, depending on the day of the week.

a series of computational rules. Using these, OBAL continuously updates the boundary information and shifts in the boundary region upon receiving responses from the user. For classification purposes, it trains an SVM classifier only on the boundary vectors recorded.

OBAL is trained on a feature dataset, rather than the raw data. Since OBAL was originally proposed for text analysis, it does not have a specific set of features designed for our context. Therefore, we used the signature's mean, minimum, maximum, standard deviation, kurtosis, and skewness, to create the feature version of our dataset. According to OBAL, if the number of stored vectors exceeds a certain number ω (in our case, we use $\omega = 15$), all non-boundary vectors are dropped to reduce the complexity. In addition, OBAL's SVM classifier is retrained at every labeled signature received.

Although OBAL is originally proposed to distinguish between two classes, it can be extended to multi-class scenarios. However, the number of classes is fixed and should be defined at the beginning. We set this value to the number of appliances in our dataset. However, this implies that introducing new appliances is problematic for this approach.

It is important to note that OBAL requires a minimum number of boundary samples per class in the initial phase, known as *cold start*. This requirement prevents the algorithm to work with a few number of samples per class. In order to have a fair comparison, we provided OBAL with the cold start boundary samples before starting our experiments. This is a clear advantage for OBAL, since our approach starts with *zero knowledge* of the signatures. Nevertheless, as the experiments show, even with such great advantage, our approach is able to outperform OBAL.

4.2.2 Convolutional Neural Networks (CNN) with an always available user. We implement a Convolutional Neural Network (CNN) classification approach recently proposed in [44]. CNNs are generally designed for image classification. Nevertheless, in this work the authors convert the signatures into images, which are then used to train the CNN. Specifically, signatures are trimmed of leading zeros to standardize them across devices. Subsequently, each signature is converted into a 320×320 gray-scale image as detailed in [44]. The image and the corresponding label are used to train the CNN.

Following the VGG-16 design, the model inputs the gray-scale image, then runs it through a series of 2D-convolutional layers and max pool layers, using RELU as the activation function. At the end, there are 3 dense layers and a softmax layer of 1000 is used for the final classification. We train this with the ADAM optimizer provided in Tensorflow.

Since this approach is not designed for learning the user availability, we assumed that the user is *always available*. That is, the user always provides the label when queried. At the beginning of each day with a budget B , the model randomly selects B time slots at which it will query the user. The training occurs at the end of each day for 1 epoch over the entire set of collected labels.

4.2.3 SMARTCOMP. We also compare KAN to the algorithm proposed in the conference version of this paper, denoted as SMARTCOMP [38]. The SMARTCOMP querying strategy only adopts the $S()$ score, defined as $S(\mathbf{x}_n) = \max\{I(\mathbf{x}_n), R(\mathbf{x}_n)\}$, (i.e., it does not use the confidence score) along with the user response distribution. Additionally, this strategy is *not adaptive*. In fact, if the incoming signature at time slot h has the $S()$ score above a threshold, and the budget B allows, the algorithm queries the user with probability $H(h)$. This may penalize the performance at a later stage when high quality signatures are rare. In these cases, the user may not be queried because $H(h)$ is low, even if there is residual budget available. This problem specifically accentuates while using the realistic user behavior model. In this case, the users infrequently respond to the queries which results in low $H(h)$ values.

Finally, SMARTCOMP applies a DTW-based KNN classification method, which is a majority voting among the $K(= 3)$ nearest neighbors using DTW as the distance metric.

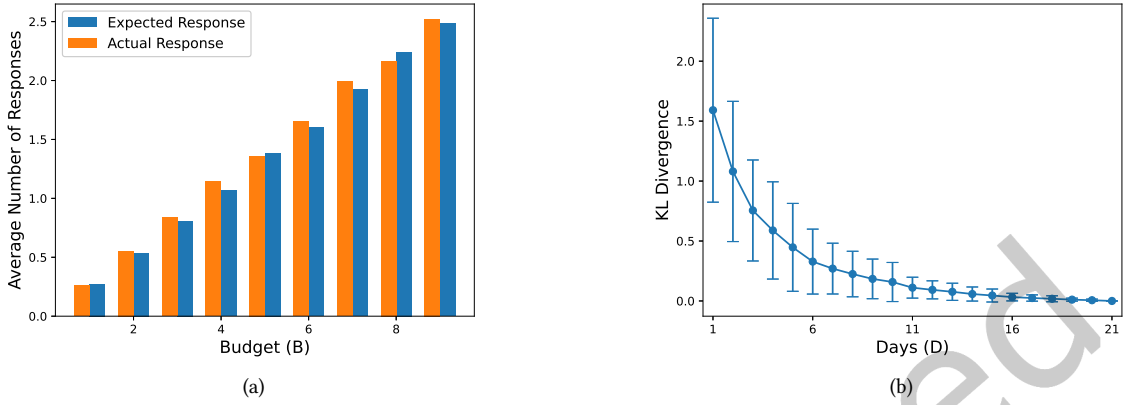


Fig. 5. (a) Actual vs. Experimental number of responses for different budgets, and (b) Average KL Divergence between learned and ground truth (perfect user response) distribution

4.3 Experimental Results

We outline four sets of experiments to test the performance of the proposed algorithm. We first validate the assumption on the binomial distribution to model the user behavior. Subsequently, we study the performance with respect to the available budget, the sensitivity with respect to the signature length and sampling frequency, and finally the impact of mislabeled appliances.

4.3.1 Learning of the user response distribution. In this section, we first validate our assumption on the user behavior and then show the ability of KAN to learn the user response distribution.

As detailed in Section 2, in this paper we assume that the user responds to a query at a time slot h with an independent probability $H(h)$. In order to validate this assumption we perform the following experiment. We first extract the *empirical distribution* from the MPU dataset for each user having at least 21 days of data (see Fig. 4). Next, for each user, we pick B time slots at random over 10 days, and compare the *actual* number of responses with the *expected* number of responses obtained from the empirical distribution. Fig. 5(a) shows the results for different values of the budget B . Clearly, the empirical distribution allows us to closely predict the number of responses provided by the user. It is also interesting to notice that even for a budget of 8, only about 2.2 responses were received on average. This shows the importance of customizing the querying strategy to the user behavior in order to query when the user is more likely to respond.

Next, we study the ability of KAN to learn the user response distribution. To this purpose, we study the Kullback–Leibler (KL) divergence between the empirical distribution up to a certain day D , and the empirical distribution over the entire period of 21 days [21]. The latter represents here the *ground truth* distribution that we would like to learn. We calculate the divergence for each user and average the results. As Fig. 5(b) shows, the learned distribution rapidly converges to the ground truth distribution. In fact, only 11 days are sufficient to learn the distribution with a negligible error. This experiment also supports the validity of our user modeling with independent probabilities. Intuitively, if the empirical distribution in the first D days would differ from the ground truth distribution, the divergence would increase rather than decrease. The figure represents a monotonic and rapid decrease over time, supporting the consistency of the user behavior to our modeling assumption.

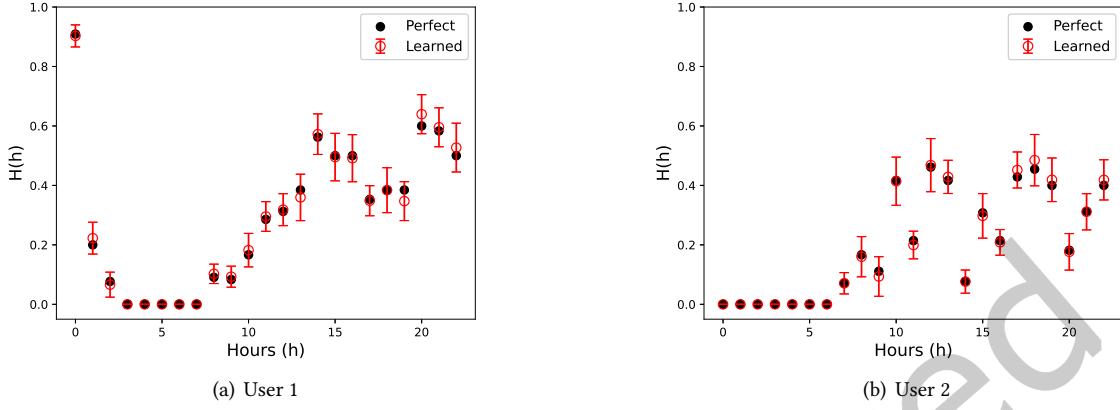


Fig. 6. Comparison of learned and ground truth (perfect user response) distribution for two arbitrary users in the MPU dataset

Finally, we provide an example of two specific users in the MPU dataset, and compare the ground truth distribution to the empirical distribution distribution learned after 17 days. For each user in this experiment, we randomly select 17 days and average it over 25 trials. Figs. 6 (a) and (b) illustrate the results. Clearly, the learned distribution closely matches the true distribution for every hour. The higher error bars for probabilities in the range $[0.2, 0.6]$ are due to the higher standard deviation of the binomial distribution for probabilities in that range.

4.3.2 Accuracy versus time and budget. In this section, we study the accuracy over time of KAN versus the comparison algorithms under different settings of the budget with the smart outlet dataset. In these experiments, the dataset is randomly split into training (80%) and testing (20%) sets. In testing, the *accuracy* is defined as the proportion of instances in the test set for which the label is correctly predicted by the trained classifier.

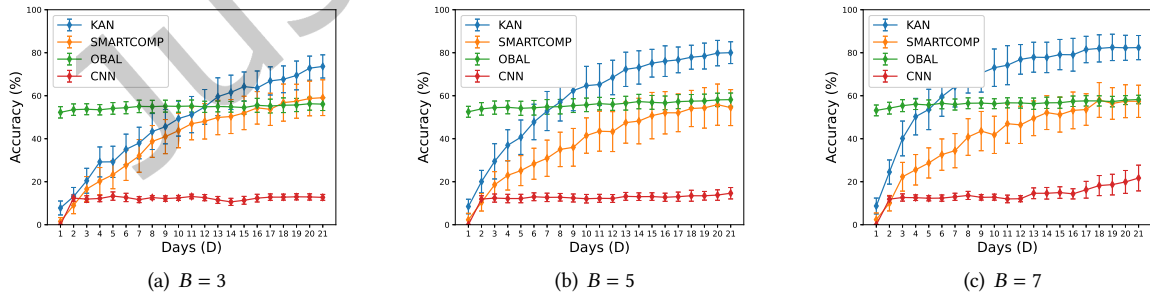


Fig. 7. Accuracy over 21 days of training for four different approaches and budget values of $B \in \{3, 5, 7\}$ plotted with a 95% confidence interval.

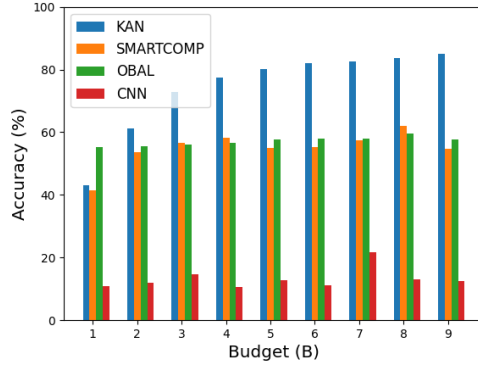


Fig. 8. Accuracy after 21 days of training under different budgets with the smart outlet dataset.

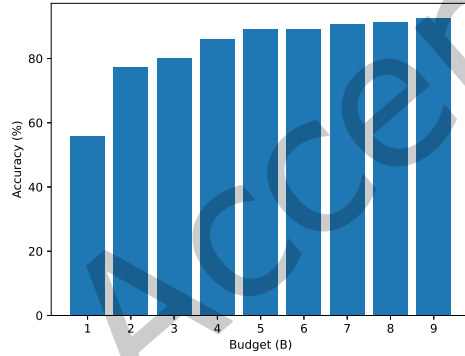


Fig. 9. Accuracy of KAN after 21 days of training under different budgets with the ECO dataset.

Figs. 7 (a)-(c) demonstrate the average accuracy of the four methods over time for three different budget values, namely $B \in \{3, 5, 7\}$. We average the experiments over 20 trials and show the resulting standard deviation as the error bars. As expected, all approaches achieve higher accuracy for higher budgets, since there are more chances to query the user and obtain more labels. However, OBAL even with the required cold start signatures, is not able to achieve high accuracy. This is due to two main reasons. First, OBAL does not explicitly consider the user behavior. As a result, it may decide to query the user even when the user has a low likelihood to provide a label. Consequently, it is only able to acquire few additional labels over time, and the impact of higher budget is minimal. Secondly, the SVM classifier suffers in the domain of electrical signatures. In fact, it is challenging to partition the hyperspace according to the identified features, as they may not capture the complexity of the appliance current signatures. The CNN approach significantly underperforms compared to the other models. This is due to the limited amount of data available to train a machine learning model in the considered residential setting. In fact, even over 21 days of training, with a high budget such as 9, and considering an always available user, the model may collect less than 200 signatures. This is far below the amount needed for a deep neural network to work well. Note that, even if over a very long time period the performance of the CNN model

could improve, this would be at the cost of a prolonged extra burden to the user. As a result, this approach is not apt for a stream based active learning residential scenario with non-expert users. Additionally, CNNs are naturally computationally heavy algorithms which require considerably powerful hardware to be executed. This further limits the applicability of these approaches in a residential scenario. As an example, to execute the CNN algorithm we relied on the High Performance Computing platform of the University of Kentucky. Even though we used such powerful infrastructure, it took several days to complete our experiments. Conversely, KAN could be executed in a few minutes on a regular personal computer.

The SMARTCOMP approach shows an increasing accuracy, however it suffers from the lack of adaptability to the availability of quality signatures over time, which results in less labels acquired from the user and ultimately a lower accuracy. On the contrary, KAN is able to learn the user response distribution and adapt the querying strategy to the different availability of high quality signatures over time. As a result, KAN outperforms both approaches, showing superior performance under all settings of the budget.

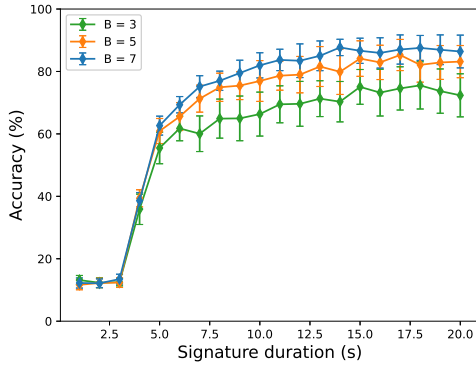
Next, we indicate the impact of different budgets on accuracy. In these experiments, we consider the accuracy after 21 days of training and changed the budget B from 1 to 9 maximum number of queries a day. Results are depicted in Fig. 8. OBAL achieves slightly higher accuracy when the budget is equal to one query a day. This is due to the cold start signatures that this approach is provided, differently from SMARTCOMP and KAN. Nevertheless, as the budget increases OBAL provides significant lower accuracy than the other approaches, especially in comparison to KAN. The CNN approach, consistent with the overall accuracy results, has the lowest accuracy across all budgets. SMARTCOMP in this graph clearly shows its inability to efficiently use the budget. In fact, provided that a signature is sufficiently informative and/or representative, SMARTCOMP queries the users at time slot h with probability $H(h)$. When the budget is high the approach should query the user more often, even at time slots with low values of $H(h)$. However this rarely happens, resulting in a low number of queries and labels, impacting the overall accuracy of SMARTCOMP. Conversely, KAN is able to adjust its querying strategy to the budget value as well as the availability of the quality of signatures. As a result, it achieves a significantly higher accuracy than the other approaches.

To further support the previous results, we now test KAN over the ECO dataset [41]. The results are shown in Fig. 9. KAN performs even better with this dataset due to the different types of appliances available. Specifically, the ECO dataset contains several high consumption devices such as a refrigerator and a freezer in addition to several lower powered devices like a PC and kettle. This means the average power consumption of each signature is more diverse in ECO, making it easier for KAN to distinguish between signatures.

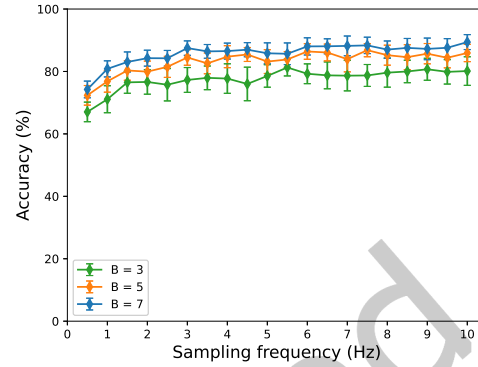
4.3.3 Sensitivity Analysis. As mentioned in Section 4.1.1, the smart outlet collects current signatures of the plugged appliance and transmits it to a server. Clearly, the outlet could be configured to collect the signatures for different lengths of time and frequencies. In these experiments, we investigate the impact of these two design parameters on the accuracy of KAN.

In Fig. 10(a), we increase the duration of the signature from 0.5 to 20 seconds and show the resulting accuracy for different budget values, namely $B = \{3, 5, 7\}$. For all the three budgets, the accuracy increases initially and then saturates. Intuitively, when the signature is too short (≤ 3 sec) the available information is not sufficient to discern among different appliances. As the signature length increases, the accuracy rapidly increases, and it plateaus at 12.5sec. This gives an important insight for the design of KAN. In fact, a shorter signature means less data transmitted and processed, and thus a reduced overall complexity of the approach.

Furthermore, the impact of sampling frequency on the performance of KAN is depicted in Fig. 10(b). We collected signatures at different frequencies, namely from 0.5 to 10Hz. As observed, although the accuracy increases initially, it plateaus quickly at about 3Hz. Pairing this result with the impact of signature duration, we



(a) Impact of signature duration on accuracy of KAN



(b) Impact of sampling frequency on accuracy of KAN

Fig. 10. Impact of signature duration and sampling frequency on the accuracy over 21 days for KAN, plotted with 95% confidence intervals.

Table 3. Impact of mislabeling percentage on accuracy for the three algorithms with $B = 5$, $D = 21$

Err. Labels	KAN	SMARTCOMP	OBAL
0%	69.46 ± 4.9	50.07 ± 5.4	43.97 ± 7.3
10%	65.48 ± 4.6	40.78 ± 6.06	45.10 ± 7.3
20%	56.66 ± 5.1	40.16 ± 5.4	43.64 ± 6.0

can conclude that KAN is able to achieve good accuracy with a minimal amount of data. This makes it particularly suitable for low cost smart outlets, as well as for Internet of Things (IoT) architectures, where storage and computational resources are limited.

4.3.4 Impact of mislabeling on classification accuracy. Involving non-expert users in the labeling task does not only imply a lack of availability, but also the occurrence of some erroneously labeled signatures. In the following experiment, we study the robustness of the considered algorithms to low levels of mislabeling, i.e., 0%, 10% and 20%. Since the labeling task is not a difficult, we expect this range to be realistic, also considering that the user can also simply abstain from labeling. The mislabeling percentage is the portion of the training labels that are incorrect. It is worth mentioning that due to poor performance of the CNN approach and its extremely low accuracy, i.e., below 20%, we skip the comparison with this algorithm in this experiment.

For this experiment, we set the strategy parameters to $B = 5$ and $D = 21$. Note that the performance of all strategies within these design parameters and perfect labels, 0% of mislabeling, is equivalent to the point at Day 21 in Fig. 7(b). Table 3 demonstrates results. OBAL shows an apparent higher robustness to mislabeling compared to KAN and SMARTCOMP. This is however due to the cold start that gives this approach an advantage with respect to the others. Nevertheless, KAN outperforms both SMARTCOMP and OBAL in terms of accuracy, achieving 16% higher accuracy than SMARTCOMP, and 13% higher than OBAL, when 20% of data is mislabeled. We expect that the robustness of KAN could be further improved by increasing the value of K for the K -Nearest-Neighbors algorithm.

5 RELATED WORKS

One of the first studies focusing on appliance recognition, equivalently called device identification, proposed Non Intrusive Load Monitoring (NILM) to analyze the voltage and/or current of the total load to determine the energy consumption of individual appliances [15]. The goal is to *disaggregate* and identify individual household appliances from *aggregated* smart meter readings. Along the same lines, other approaches have been proposed for load disaggregation [20, 42, 47]. In particular, the authors of [18] use deep learning neural networks as a building block to design a NILM approach. Recently, to overcome low recognition accuracy of existing NILM techniques, [44] studies a NILM method based on convolutional neural network (CNN). This method is used as the comparison approach in this work and detailed in Section 4.2.2. Furthermore, [42] presents a passive Bayes classification model for recognition of household appliances using power consumption data from intelligent power sockets. This paper incorporates the characteristics of occupant behavior in terms of time and frequency of using the appliances throughout the day. However, the main disadvantage of all these techniques is that they become computationally intensive as the number of devices grows. Moreover, unlike our proposed algorithm, these models require pre-training.

To reduce computational burden, low frequency-sampling smart outlets have been proposed [32]. These outlets have a limited number of devices connected to them, thus reducing the complexity and facilitating the recognition of low consumption devices. In this work, the authors proposed the ANNOT system, which autonomously annotates electricity data using wireless sensor nodes, such as temperature, light and sound sensors, places next to appliances to identify the operating state of an appliance. This approach requires either a specific code for each single node or user involvement to label appliances. Very recently, a commercial smart outlet has been released by Currant [7]. This tool, however, is only able to track outlet usage over time, but does not perform appliance recognition.

A different approach involves understanding the particular electrical signature produced by home appliances that is resultant of their individual circuit design. This signature may be used as a finger print to identify the electric load [19, 33]. Specifically, [33] studies the feasibility of Independent Component Analysis (ICA) to disaggregate a load profile from its composite current signature. Other works consider different signatures based on power consumption patterns, such as phase shift, active and reactive power [2, 30]. Specifically in [2], several offline (pre-trained) machine learning approaches, both supervised and unsupervised, have been examined for real time identification of appliances using the power consumption pattern of electrical devices. In this work we use low frequency smart outlets and rely solely on the current draw to reduce hardware costs to the user.

In order to compare different signatures for classification, the Dynamic Time Warping (DTW) algorithm can be used to obtain the distance between two time-series, or event signatures. In [23], the authors used a comparative DTW learning algorithm to classify devices based on current data. Note that, unlike [23] which uses DTW as the classifier, we adopt DTW only as a means of constructing a distance matrix to guide a KNN classifier.

The works mentioned above consider offline classification. Such approach is not suitable for a smart home context, since the user's own appliances may have very different signatures than that of the offline training data.

Active learning is a technique which trains a model with minimum cost by using data obtained from labeling only the most informative instances [12]. An active learning scheme can be pool-based, i.e., all training samples are available in a "pool," and the system must decide which of these samples should be labeled by the expert user for maximal information gain. This approach intelligently prevents the training from being overwhelmed by uninformative samples by taking advantage of information available from past queries, i.e., already known labels [36]. To determine informativeness of the instance, some works, such as [11, 37], consider Uncertainty Sampling (US) as a selection criteria, in which instances with the highest uncertainty are chosen to be labeled

[6, 8, 31]. Some other metrics to measure *uncertainty* include maximum entropy [8], smallest-margin [31] and least confident [6].

Alternatively, [46] studied a new class of active learning from data streams. In such scenarios, the data volumes increase continuously, and new information arrives in “batches.” They proposed a classifier ensemble to decide which portion of the incoming stream data should be labeled. Accordingly, [13] studies budgeted stream-based active learning, and investigates the possibility of converting existing active learning methods to stream-based scenarios while also providing theoretical guarantees on their performances. In a recent work, [26] developed an algorithm for crisis management applications using data streams from social media. Given a labeling budget, this algorithm adopts a set of uncertainty strategies to determine the items for labeling.

All the above mentioned active learning papers assume the presence of an *always available* oracle (expert) willing to label the most important samples. This is not the case in a smart home scenario, in which a user is performing the labeling and may abstain from labeling or simply not be willing to respond at any time. The problem of not only selecting the best instances, but also adapting to the user availability to respond in terms of abstention rate, has been first introduced in [43], who use the decision boundary as a guide to predicting user abstention rates. However, their strategy is not applicable on labeling tasks which are more vulnerable to abstention. In [35], the authors assume the presence of a uniformly random oracle, meaning that they will always respond to queries with a fixed probability. Not surprisingly, the authors show that a random sampling strategy is the best in this specific (unrealistic) context.

In this work, we advance the state-of-the-art research by proposing an interactive and adaptive learning approach which is a user-centered stream-based active learning method. Our approach not only picks the best instances to improve the appliance recognition performance, but it also learns the user availability to interact with the system and provide labels for the queried instances.

6 CONCLUSIONS

In this work, we address the problem of creating interactive and adaptive machine learning algorithms that specifically take into account the user’s behavior in determining learning choices. To this aim, we introduce the KAN algorithm, which takes as input the appliance signatures obtained from a low-cost and low-frequency smart outlet. KAN is characterized as a stream-based active learning algorithm that incorporates signature quality and user behavior, in terms of user response distribution.

We exploited the Mobile Phone Use (MPU) dataset to obtain a realistic user behavior model for user engagement with the querying systems. This dataset shows that regular users rarely interact with such systems, motivating the importance of designing machine learning algorithms that jointly learn user behavior through the classification task.

We compared the proposed algorithm with two state-of-the-art stream based active learning approaches, called Online Batch-based Active Learning (OBAL), and Convolutional Neural Networks (CNN). In the experiments, we used a dataset of electrical signatures of common household appliances collected using an Arduino-based smart outlet along with signatures from the ECO dataset. Experiments show the superiority of KAN in a variety of scenarios. Furthermore, our approach is able to work with appliance signatures of short length and low frequency, making it particularly apt to Internet of Things applications.

ACKNOWLEDGMENTS

This work is supported by the National Institute for Food and Agriculture (NIFA) under the grant 2017-67008-26145, the NSF grant EPCN 1936131, and the NSF CAREER grant CPS-1943035.

REFERENCES

- [1] 2015. *RECS Survey Data*. <https://www.eia.gov/consumption/residential/data/2015/>.
- [2] Vibhatha Abeykoon, Nishadi Kankanamdurage, Anuruddha Senevirathna, Pasika Ranaweera, and Rajitha Udawapola. 2016. Real Time Identification of Electrical Devices through Power Consumption Pattern Detection. In *First International Conference on Micro and Nano Technologies, Modelling and Simulation*.
- [3] U.S. Energy Information Administration. April 2015. *Annual Energy Outlook 2015 with projections to 2040*. [http://www.eia.gov/forecasts/aeo/pdf/0383\(2015\).pdf](http://www.eia.gov/forecasts/aeo/pdf/0383(2015).pdf).
- [4] Joaquim Arlandis, Juan Carlos Pérez-Cortes, and Javier Cano. 2002. Rejection strategies and confidence measures for a k-nn classifier in an ocr task. In *Object recognition supported by user interaction for service robots*, Vol. 1. IEEE, 576–579.
- [5] Hsuan-Hao Chang, Wei-Yu Chiu, Hongjian Sun, and Chia-Ming Chen. 2018. User-centric multiobjective approach to privacy preservation and energy cost minimization in smart home. *IEEE Systems Journal* 13, 1 (2018), 1030–1041.
- [6] Aron Culotta and Andrew McCallum. 2005. Reducing Labeling Effort for Structured Prediction Tasks. In *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2 (Pittsburgh, Pennsylvania) (AAAI'05)*. AAAI Press, 746–751.
- [7] Currant. 2018. WiFi Smart Outlet [Product]. Accessed: 10-31-2018.
- [8] Ido Dagan and Sean P. Engelson. 1995. Committee-Based Sampling For Training Probabilistic Classifiers. In *In Proceedings of the Twelfth International Conference on Machine Learning*. Morgan Kaufmann, 150–157.
- [9] Valeria Dolce, Courtney Jackson, Simone Silvestri, Denise Baker, and Alessandra De Paola. 2018. Social-Behavioral Aware Optimization of Energy Consumption in Smart Homes. In *2018 14th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. 163–172.
- [10] Bo Du, Zengmao Wang, Lefei Zhang, Liangpei Zhang, Wei Liu, Jialie Shen, and Dacheng Tao. 2017. Exploring representativeness and informativeness for active learning. *IEEE transactions on cybernetics* 47, 1 (2017), 14–26.
- [11] Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. 1997. Selective sampling using the query by committee algorithm. *Machine learning* 28, 2 (1997), 133–168.
- [12] Yifan Fu, Xingquan Zhu, and Bin Li. 2013. A survey on instance selection for active learning. *Knowledge and information systems* 35, 2 (2013), 249–283.
- [13] Kaito Fujii and Hisashi Kashima. 2016. Budgeted stream-based active learning via adaptive submodular maximization.. In *NIPS*, Vol. 16. 514–522.
- [14] Charles Miller Grinstead and James Laurie Snell. 2006. *Grinstead and Snell's introduction to probability*. Chance Project.
- [15] George William Hart. 1992. Nonintrusive appliance load monitoring. *Proc. IEEE* 80, 12 (1992), 1870–1891.
- [16] Eamonn Keogh and Chotirat Ann Ratanamahatana. 2005. Exact indexing of dynamic time warping. *Knowledge and information systems* 7, 3 (2005), 358–386.
- [17] Atieh R Khamesi, Simone Silvestri, Denise A Baker, and Alessandra De Paola. 2020. Perceived-value-driven optimization of energy consumption in smart homes. *ACM Transactions on Internet of Things* 1, 2 (2020), 1–26.
- [18] Jihyun Kim, Thi-Thu-Huong Le, and Howon Kim. 2017. Nonintrusive load monitoring based on advanced deep learning and novel signature. *Computational intelligence and neuroscience* (2017).
- [19] Seong-Joong Kim. 2015. Appliance recognition unit for home energy management system with UPnP network. *IEEE Systems Journal* 11, 4 (2015), 2794–2803.
- [20] J Zico Kolter and Tommi Jaakkola. 2012. Approximate inference in additive factorial hmms with application to energy disaggregation. In *Artificial Intelligence and Statistics*. 1472–1482.
- [21] Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics* 22, 1 (1951), 79–86.
- [22] Shih-chiang Lee, Gu-yuan Lin, Wan-rong Jih, and Jane Yung-jen Hsu. 2010. Appliance recognition and unattended appliance detection for energy conservation. In *Workshops at the twenty-fourth aaai conference on artificial intelligence*.
- [23] Jing Liao, Georgia Elafoudi, Lina Stankovic, and Vladimir Stankovic. 2014. Non-intrusive appliance load monitoring using low-resolution smart meter data. In *IEEE international conference on Smart grid communications (SmartGridComm)*. 535–540.
- [24] Olivier Monnier. 2013. *White paper: A smarter grid with the Internet of Things*. Technical Report. Texas Instruments. <http://www.ti.com/lit/ml/slyb214/slyb214.pdf>
- [25] Martin Pielot, Bruno Cardoso, Kleomenis Katevas, Joan Serrà, Aleksandar Matic, and Nuria Oliver. 2017. Beyond interruptibility: Predicting opportune moments to engage mobile phone users. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 1–25.
- [26] Daniela Pohl, Abdelhamid Bouchachia, and Hermann Hellwagner. 2018. Batch-based active learning: Application to social media data for crisis management. *Expert Systems with Applications* 93 (2018), 232–244.
- [27] Atieh R. Khamesi, Eura Shin, and Simone Silvestri. 2020. Machine Learning in the Wild: The Case of User-Centered Learning in Cyber Physical Systems. In *2020 International Conference on COMMunication Systems & NETworks (COMSNETS)*. IEEE, 275–281.

- [28] Atieh R. Khamesi and Simone Silvestri. 2020. Reverse auction-based demand response program: A truthful mutually beneficial mechanism. In *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 427–436.
- [29] Michael Ratner and Carol F Glover. 2014. *US energy: Overview and key statistics*. Vol. 40187. Congressional Research Service Washington, DC.
- [30] Antonio G Ruzzelli, C Nicolas, Anthony Schoofs, and Gregory MP O’Hare. 2010. Real-time recognition and profiling of appliances through a single electricity sensor. In *2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*. IEEE, 1–9.
- [31] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 2001. Active Hidden Markov Models for Information Extraction. In *Advances in Intelligent Data Analysis*. Springer Berlin Heidelberg, 309–318.
- [32] Anthony Schoofs, Antonio Guerrieri, Declan T Delaney, Gregory MP O’Hare, and Antonio G Ruzzelli. 2010. Annot: Automated electricity data annotation using wireless sensor networks. In *2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*. IEEE, 1–9.
- [33] Sunil Semwal, Deepak Joshi, RS Prasad, and Dogga Raveendhra. 2013. The practicability of ICA in home appliances load profile separation using current signature: A preliminary study. In *2013 International Conference on Power, Energy and Control (ICPEC)*. IEEE, 756–759.
- [34] Pavel Senin. 2008. Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA* 855, 1-23 (2008), 40.
- [35] Elson Serrao and Myra Spiliopoulou. 2018. Active Stream Learning with an Oracle of Unknown Availability for Sentiment Prediction.. In *IJAL@ PKDD/ECML*. 36–47.
- [36] Burr Settles. 2009. *Active learning literature survey*. Technical Report. University of Wisconsin-Madison Department of Computer Sciences.
- [37] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. 1992. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*. 287–294.
- [38] Eura Shin, Atieh R. Khamesi, Zachary Bahr, D. A. Baker, and Simone Silvestri. 2020. A User-Centered Active Learning Approach for Appliance Recognition. In *IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE.
- [39] Simone Silvestri, Denise A Baker, and Valeria Dolce. 2017. Integration of Social Behavioral Modeling for Energy Optimization in Smart Environments. In *Proceedings of the 2nd International Workshop on Social Sensing*. ACM, 97–97.
- [40] Ashutosh Timilsina, Atieh R Khamesi, Vincenzo Agate, and Simone Silvestri. 2021. A Reinforcement Learning Approach for User Preference-aware Energy Sharing Systems. *IEEE Transactions on Green Communications and Networking* (2021).
- [41] Silvia Santini Wilhelm Kleiminger, Christian Beckel. 2015. Household Occupancy Monitoring Using Electricity Meters. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2015)*. ACM.
- [42] Da Yan, Yuan Jin, Hongsan Sun, Bing Dong, Zi Ye, Zhaoxuan Li, and Yanping Yuan. 2019. Household appliance recognition through a Bayes classification model. *Sustainable Cities and Society* 46 (2019), 101393.
- [43] Songbai Yan, Kamalika Chaudhuri, and Tara Javidi. 2016. Active Learning from Imperfect Labelers. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (Barcelona, Spain) (NIPS’16)*. Curran Associates Inc., USA, 2136–2144.
- [44] Dongsheng Yang, Xiaoting Gao, Liang Kong, Yongheng Pang, and Bowen Zhou. 2020. An event-driven convolutional neural architecture for non-intrusive load monitoring of residential appliance. *IEEE Transactions on Consumer Electronics* 66, 2 (2020), 173–182.
- [45] J. Zheng, D. W. Gao, and L. Lin. 2013. Smart Meters in Smart Grid: An Overview. In *IEEE Green Technologies Conference (GreenTech)*. 57–64.
- [46] Xingquan Zhu, Peng Zhang, Xiaodong Lin, and Yong Shi. 2007. Active learning from data streams. In *7th IEEE International Conference on Data Mining (ICDM)*. 757–762.
- [47] Ahmed Zoha, Alexander Gluhak, Muhammad Ali Imran, and Sutharshan Rajasegarar. 2012. Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey. *Sensors* 12, 12 (2012), 16838–16866.