A User-Centered Active Learning Approach for Appliance Recognition

Eura Shin[†], Atieh R. Khamesi[†], Zachary Bahr[‡], Simone Silvestri [†] and D. A. Baker[‡]
[†]Department of Computer Science, University of Kentucky, Lexington, KY, USA
[‡]Department of Psychological Science, Missouri University of Science and Technology, Rolla, MO, USA
Email: [†]{eura.shin, atieh.khamesi, simone.silvestri}@uky.edu, [‡]{zjb998, bakerden}@mst.edu

***Abstract*—Smart homes offer new possibilities for energy management. One key enabler of these systems is the ability to monitor energy consumption at the appliance level. Existing approaches rely mainly on data from aggregated smart meter readings, but lack sufficient accuracy to recognize several appliances. Conversely, *smart outlets* are a suitable alternative since they can provide accurate electrical readings on individual appliances. Previous approaches for appliance recognition based on smart outlets use passive machine learning, which are deficient in the flexibility and scalability to work with highly heterogeneous appliances in smart homes. In this paper, we propose a *stream-based active learning approach*, called K-Active-Neighbors (KAN), to address the problem of appliance recognition in smart homes. KAN is an interactive framework in which the user is asked to label signatures of recently used appliances. Differently from previous work, we consider the realistic case in which the user *is not* always available to participate in the labeling process. Therefore, the system simultaneously learns the signatures and also the user willingness to interact with the system, in order to optimize the learning process. We develop an Arduino-based smart outlet to test our approach. Results show that, compared to previous solutions, KAN achieves higher accuracy in up to 41% less time.**

***Index Terms*—Appliance Recognition, Stream-based Active learning, User-Centered Machine Learning, Labeler Abstention.**

I. Introduction

The residential sector is responsible for more than 20% of the total energy consumption of the United States [1], and this amount has been constantly increasing for several decades. As the demand for energy grows, there exists a clear need for the management of energy consumption at the residential level which can be obtained through fine-grained electricity billing, demand-response programs, and electrical power load balancing [2]. A well-recognized key enabler for energy management techniques is the knowledge about when each appliance is used and its consumption [3]–[6]. 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 [7]. To infer individual appliance consumption, load disaggregation techniques are used [8]. However, these techniques lack the sufficient accuracy to enable fine-grain energy management [8].

In this regard, *smart outlets* are a better alternative. These outlets look like traditional wall plugs, but are actually Internet of Things (IoT) devices [9], 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 [10], [11]. We argue that this approach is not practical in a smart home for several reasons: (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 signatures (patterns of current over time), making it hard to perform offline training.

In this paper, we propose the use of *active learning*, and specifically of a *stream-based active learning* strategy. Active learning assumes that the entire dataset is available during training, and an expert user labels a selected subset of instances. Conversely, in stream-based scenarios data is generated over time, when an appliance is plugged in, and the system must decide if manual labeling from the user is needed [12]. However, stream based active 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 [13]. We point out that this is not the case in a smart home, where a user may have a different level of engagement and availability over time, and the number of queries should be limited to prevent overwhelming her. As a result, in this context, in conjunction to learning the appliances' signatures, we must also learn the user's behavior in terms of the likelihood of completing the labeling task, in order to optimize the overall learning process. To the best of our knowledge the problem of user abstention for active querying has only been addressed in [14], [15]. 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 the smart home.

In this paper we propose a stream-based machine learning framework named K-Active-Neighbors (KAN) for appliance recognition that comprehensively learns appliances' signatures and the user behavior in engaging with the labeling task. KAN trades off informativeness of new instances with the likelihood of user engagement in order to improve the accuracy and shorten the convergence of the identification procedure. We develop an Arduino-based smart outlet to test the performance of KAN versus existing approach using several appliances.

978-1-7281-6997-2/20/$31.00 ©2020 IEEE
DOI 10.1109/SMARTCOMP50058.2020.00047

Our results show that we are able to achieve 100% accuracy in up to 66% less time with respect to previous solutions.

In summary, the main contributions of this paper are:

1) We propose an algorithm called KAN for appliance recognition in smart homes;
2) KAN minimizes the instances necessary to train this classifier by selecting the most informative instances and learning the user's behavior;
3) We tested KAN on real appliance signatures collected with an Arduino-based smart outlet;
4) Results show that our approach is faster in achieving high accuracy, and also quickly learns the user behavior.

II. System Model and Problem Statement

We assume that time is divided into time slots, for example corresponding to the 24 hours per day, denoted as $h = 1, \ldots, 24$. As a labeler, the user is assumed to follow a certain response distribution according to his willingness/availability to interact with the system. In this paper we use an independent Bernoulli distribution at each hour; nevertheless more complex models can be easily integrated in our framework. As a result, we refer to $P(h)$ as the probability that the user will successfully respond to a query at hour h. 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 IV-C1.

We consider a smart outlet with minimal hardware costs, and as a result, the tool is characterized by a low-frequency sampling rate and collection of only electric current information. When an appliance is plugged in the outlet and turned ON, the electric current data obtained by the outlet results in a time series of amp values, sampled at regular intervals (e.g., 5 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. In addition, some appliances may only ON/OFF states (type I), may multiple states (type II), or continuously variable states (type III) [16]. This clearly affects the resulting signature. We refer to X as the set of signatures observed by the smart outlet. X grows over time as new signatures arrive.

Problem Statement: Without loss of generality, we assume that the 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 appliance label. Intuitively, an upcoming signature should be queried if it represents (i) a new appliance, i.e., it does not match any existing label, or (ii) similar to an already observed signature but with inadequate labels.

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 query strategy to (i) maximize the accuracy of recognizing appliances by querying signatures that are most informative; (ii) learn the user distribution $P(h)$ in each time slot, and use this distribution to optimize the query strategy; and (iii) do not exceed the maximum budget constraint.

III. Proposed Solution

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

A. Dynamic Time Warping

DTW is a distance measure between two different time series of potentially different length [17]. 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 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 [18]. In this paper, we adopt DTW as a means of constructing a distance matrix to guide the KAN's stream-based active learning process.

B. K-Active-Neighbors Approach

We propose a K-Nearest-Neighbors based active learning algorithm, named K-Active-Neighbors (KAN), as the querying strategy. KAN considers *informativeness*, *representativeness*, and the predicted user engagement, $P(h)$, of an incoming signature as the querying criteria.

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 [19].

Consider an incoming event signature, $\mathbf{x}_n$, and the set of instances already observed by the smart outlet, $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n-1}\}$. We let $\mathrm{KNN}(\mathbf{x}_n) = \{\mathbf{x}_1^{(n)}, \ldots, \mathbf{x}_K^{(n)}\}$ represent the K nearest neighbors of $\mathbf{x}_n$ in X, where DTW is used as a 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 $\mathbf{x}_a \in \mathrm{KNN}(\mathbf{x}_b)$.

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 represents the total number of edges $\mathbf{x}_n$ receives, which is also the total number of times $\mathbf{x}_n \in \mathrm{KNN}(\mathbf{x}_i)$ for all $\mathbf{x}_i \in X$. This is formally defined as $N_R(\mathbf{x}_n) = \sum_{i=1}^{|X|} \mathbf{1}(\mathbf{x}_n \in \mathrm{KNN}(\mathbf{x}_i))$.where, $\mathbf{1}(\cdot)$ represents the indicator function.

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 $l(\mathbf{x}_i)$. The number of unlabeled instances a sample affects is denoted by $N_I(\mathbf{x}_n) = \sum_{i=1}^{|X|} \mathbf{1}(\mathbf{x}_n \in \mathrm{KNN}(\mathbf{x}_i) \ \& \ \neg l(\mathbf{x}_i)))$.

The definition N_I considers how many of the received edges come from samples with known labels.

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% 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 bring these two criteria together by saying a sample is worth querying if it is highly informative or highly representative of the dataset. A sample's *querying score*, which ranges from $[0, 1]$ is the highest of the two percentile values $I(\mathbf{x}_n)$, i.e., a sample is related to many unlabeled points, and $R(\mathbf{x}_n)$, i.e., a sample represents a sparse area. The querying score is defined as $S(\mathbf{x}_n) = \max\{I(\mathbf{x}_n),\ R(\mathbf{x}_n)\}$. We use a threshold T to determine if the querying score is sufficiently high and a query to the user is needed. Nevertheless, such score is not the only factor in determining the querying decision. In addition to the budget, we consider the willingness of the user to answer that query at the current time slot, as explained in the following section.

2) Learning the user's response distribution: The KAN algorithm considers the *time* to query the labeler by learning the user's response distribution. For every $P(h)$ 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 slot h, and let $n(h)$ represent the total number of queries submitted at that slot. Thus, $P(h)$ is given by,

$$P(h) = \begin{cases} \alpha, & \text{if } n(h) \leq \epsilon_1 \\ \beta, & \text{if } s(h) \leq \epsilon_2 \\ \dfrac{s(h)}{n(h)}, & \text{otherwise} \end{cases} \tag{1}$$

Those hours that have been tested fewer than ϵ_1 trials, will have a probability of α. Hours that have been tested fewer than ϵ_2 trials but did not receive a response will still maintain a minimal probability of $\beta > 0$. The parameters ϵ_2 and ϵ_1 are necessary to ensure that sufficient samples are available to calculate the average $\frac{s(h)}{n(h)}$. The parameters ϵ_1 and ϵ_2 may be derived using any strategy for selecting a sample size to estimate a population mean [20].

If a new signature $\mathbf{x}_n$, generated at time slot h, passes the tests for informativeness and representativeness, and the budget B is not exceeded, the user is asked to label the data with probability $P(h)$. The system observes the user behavior to update $s(h)$, $n(h)$ and $P(h)$.

Algorithm 1: K-Active-Neighbors algorithm

Input : Incoming signature $\mathbf{x}_n$, hour h, budget B, used budget b, $\mathrm{KNN}(\mathbf{x}_n)$, threshold T.

1 Find $N_R(\mathbf{x}_n)$ to calculate $R(\mathbf{x}_n)$
2 Find $N_I(\mathbf{x}_n)$ to calculate $I(\mathbf{x}_n)$
3 $S(\mathbf{x}_n) = \max\{I(\mathbf{x}_n), R(\mathbf{x}_n)\}$
4 **if** $S(\mathbf{x}_n) \geq T \ \ \& \ \ b < B$ **then**
5 Query the user with probability P(h)
6 **if** *user has been queried and query is successful* **then**
7 $s(h)++$
8 $n(h)++$
9 $b++$
10 $l(\mathbf{x}_n)$ = True
11 **end**
12 **if** *user has been queried and query is not successful* **then**
13 $n(h)++$
14 $b++$
15 $l(\mathbf{x}_n)$ = False
16 **end**
17 $X = X \cup \{\mathbf{x}_n\}$
18 **end**

3) Pseudocode: In the following, we provide the pseudocode of our proposed algorithm. where b is the budget used, or queries for the day, at any given instance. This value is reset to $b = 0$ at the beginning of each day. Lines 1 and 2 of the pseudocode use the definitions of N_R, N_I to respectively derive the number of representative and informative edges possessed by the sample $\mathbf{x}_n$. In line 3, the score is determined by percentile rank of neighbors. The condition in line 4 checks whether the sample querying score is higher than the threshold T and would not surpass the allotted budget. If the condition is true, the user is queried with probability $P(h)$ (line 5). Depending whether the user responds, or not, $s(h)$, $n(h)$, $P(h)$ and $l(\mathbf{x}_n)$ are updated accordingly (lines $6-16$). Finally, $\mathbf{x}_n$ is added to the current set of signatures (line 17).

C. Classification

We use the DTW distance metric in guiding the KNN classifier. We adopted a 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 label. Like the original KNN algorithm, the DTW-based KNN is a lazy algorithm which requires no official training period. The DTW-KNN approach is well suited for a dynamic household in which the dataset is constantly evolving. In classification, a signature is assigned the label given by the majority of its K nearest neighbors, and in our experiments we set $K = 3$ to mirror the value of K in the KAN algorithm.

IV. Experimental Results

A. Experimental Setup

In the following experiments, we explore the how budget, user responses within a day, and mislabeling affect different learning strategies. In this paper, we make design choices to isolate these characteristics, which may not account for all residential scenarios. However, our experimental framework may be used to explore alternative design choices in the future, such as user response behavior, the budget, and types/runtimes of different appliances.

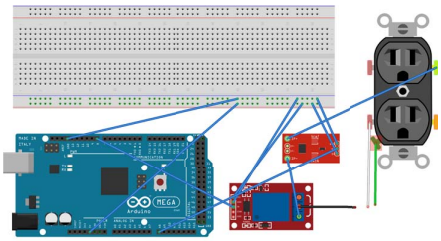
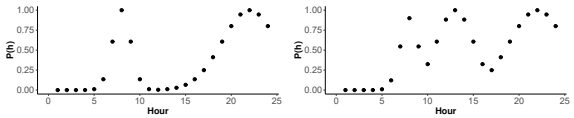


Fig. 1: Schematic of the Arduino-based Smart Outlet



(a) User 1 (b) User 2

Fig. 2: Fixed user response distributions

1) Smart Outlet: We implement a low-cost Arduino-based smart outlet, which captures the current signal at 0.2Hz, i.e., one sample every 5 seconds, and saves this information on a remote server. 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. 1. 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 Yun 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.

2) Dataset: We use our smart outlet to collect a total 380 *signatures* including 5 *appliances* of different types, namely a laptop (type III), iPhone (type I), LED lamp (type I), fan (type II) and hair dryer (type II), because they are common, portable households appliances. Though our selected appliances are not exhaustive, they test our solution's stability, as they cover challenging characteristics such as low consumption appliances. During data acquisition, each appliance is plugged in our the smart outlet and turned ON for one minute before being turned OFF. Type II appliances have been turned ON in different states during the data collection. Appliance signatures are restricted to one minute for comparability when testing the impact of other attributes (budget, user response, mislabeling) in the experiments below. In reality, DTW will allow us to capture and compare appliances of varying run times, such as a toaster or an entire washing machine cycle.

The collected data is then used to train and test the models as described in Section IV-B. In each trial of the experiment, the data is randomly split into training (80%, 304 instances) and testing (20%, 76 instances) 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.

We consider a training period of $D = 21$. We also assume that three appliance signatures are observed by the system every hour, from a potential pool of five appliances. Accordingly, there are $N = 24 \times 3 = 72$ signatures observed within a day. These N signatures are sampled with replacement from the entire training set and randomly assigned an hour h to form a usage schedule. To mimic how residents would use appliances in the same manner every day, our generated usage schedule remains the same for every day of training throughout a trial of the experiment. Finally, we set the KAN parameters K and t, defined in Algorithm 1, equal to 3 and 0.9, respectively. We also set $\alpha = \frac{1}{2}$, $\beta = \frac{1}{24}$ in Eq. (1). The chosen α value represents a completely random probability of success, β is the uniform probability a user will respond in any of the hours of the day, and $\epsilon_1, \epsilon_2 = 0$. We performed a sensitivity analysis and observed similar trends with similar settings of these parameters.

3) User Modeling Parameters: To evaluate the robustness of our approach over different user response patterns, we consider two distributions, displayed in Fig. 2, which are created as a mixture of independent Gaussian distributions. User 1 has a typical 9A.M. to 5P.M. job and is available to answer queries in the mornings and evenings surrounding these times. User 2 is available throughout the day and hence follows a schedule composed of more Gaussian components.

B. Comparison Approaches

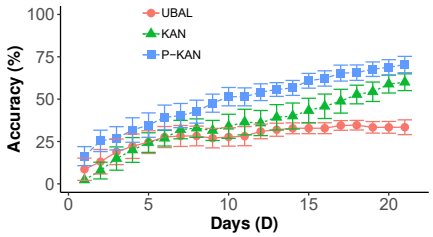
We compare three different stream learning strategies.

1) UBAL Sampling and Decision Tree Classifier: We compare our approach to the solution proposed in [14], the Uniform Budgeted Active Learning (UBAL) algorithm, paired with the Decision Tree (DT) classifier. UBAL randomly chooses B time slots in a day to query the user. To the best of our knowledge, the sampling algorithm outlined by [14] is the only other algorithm in literature that considers the problem of addressing user abstention for active querying.

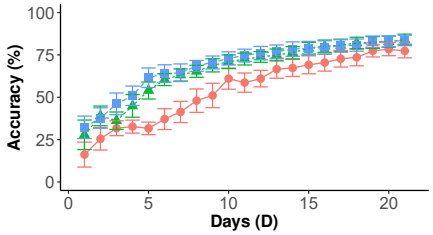
The DT is trained on a feature set extracted from the raw time series data, which includes standard time series quantifiers such as mean, minimum, maximum, standard deviation, kurtosis and skew. Moreover, the transition time, i.e., the time taken for an appliance to reach a steady state when turned ON, is also included in this dataset. Thus, the dataset contains seven features. On a five-fold cross validation of the this 380 instance feature set, we find that DT classifier achieves an accuracy of 96.4%, which was comparable to our other tested classification methods, such as Neural Networks (95.9%), logistic regression (93.5%), and support vector machines (78.9%), with untuned hyper-parameters for all models.

2) KAN: In this work, we propose the K-Active-Neighbors query strategy. We adopt this algorithm along with the DTW-based KNN classification method, collectively referred to as "KAN" for notation simplicity. On a five-fold cross validation of the entire 380 instance dataset, we find that the DTW-KNN classifier achieves an accuracy of 96.5%.

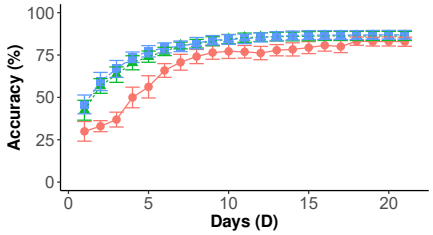
3) Perfect KAN: We also include an adaptation of the KAN strategy where the perfect user response distribution, $P(h)$ for $h = 1, \ldots, 24$, is known and given to the KAN algorithm. This is referred to as the "perfect KAN", or P-KAN. This may be considered the upper bound when testing the strategies.



(a) $B = 1$



(b) $B = 3$



(c) $B = 6$

Fig. 3: Results of three active learning strategies on User 2 distribution for $B \in \{1, 3, 6\}$ plotted with a 5% confidence interval.

C. Experimental Results

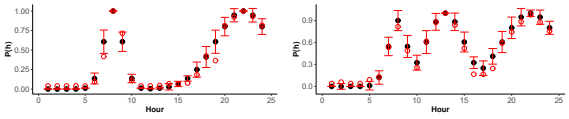
We outline three experiments that test the robustness of our proposed algorithm with respect to varying budgets, user response distributions, and mislabeled appliances.

1) Impact of budget: In the first experiment, we examine the impact of the budget B on the performance of the three strategies described in Subsection IV-B. Fig. 3 displays the average testing accuracy of the three methods over the considered period for different budget values on User 2's distribution. The results look nearly identical for User 1, so we omit the related figures due to space limitations. Each bullet corresponds to the accuracy of the classifier on the testing data at the end of each day. To account for the randomness in the sampling methods, every experiment is repeated and averaged across 10 trials.

As observed, the accuracy of the KAN and P-KAN methods is noticeably higher than the UBAL strategy. In fact, for higher budgets, the KAN algorithm is able to ignore redundant instances and learn the user distribution faster. As a result, the number of days necessary for the KAN algorithm to reach a higher accuracy, such as 80%, consistently takes less time than for UBAL, even as the budget increases (see Table I).For

TABLE I: Average number of days to reach 80% accuracy

Budget	UBAL	KAN	P-KAN
1	> 21	21 ± 0	20 ± 1
3	18 ± 1	15 ± 2	14 ± 2
6	12 ± 2	7 ± 2	8 ± 2



(a) User 1 (b) User 2

Fig. 4: Comparison of average learned user response distribution with perfect user response distributions over 25 trials with 5% confidence intervals at $D = 21$, $B = 6$.

example, given $B = 3$, the KAN algorithm reaches more than 60% accuracy five days before than UBAL. In Fig. 3b, we see that the UBAL strategy also has larger confidence intervals, showing that random sampling does not always provide *quality* labels, which leads to volatile classifiers with respect to accuracy. Also according to Table I, KAN achieves similar performance to P-KAN for all budgets, suggesting that our strategy is effective a learning the user distribution similar to the one provided to P-KAN.

For the value of $B = 1$, i.e., in Fig. 3a, all three strategies perform unsatisfactorily, with the average accuracy around 0.75, 0.74 and 0.68 for the P-KAN, KAN and UBAL strategies, respectively. The reason is the limited size of the acquired labeled dataset, considering that the user may not respond to the single query of the day.

Note that, in a few cases KAN and P-KAN may show a non-monotone accuracy in consecutive days, for example for P-KAN in Fig. 3a between day 10 and 11. This is because the KNN classification method will assign a class based on the "majority vote" of its neighbors and is vulnerable to noise in small training sets. In our experiments, where $K = 3$, if the KAN strategy does not have three instances of a given class in the training set, it will include at least one incorrect vote.

2) Learning of the user response distribution: We also explore the ability of the current KAN strategy to learn the user distribution.

Fig. 4 provides a comparison between the learned user distribution and the perfect one. First, note that the learned distribution is very close to the true distribution for every hour, with a 95% confidence. The error bars are considerably larger for mid-range $P(h)$ values, i.e., values within a $[0.25, 0.75]$ interval. This implies that our strategy can distinguish hours in which the user is *always* or *never* willing to respond to queries, but less precise at learning mid-range $P(h)$ values. This can be explained from a statistical standpoint, because the standard deviation is expected to decrease as P_{success} approaches 0 or 1 in a binomial distribution. Moreover, this clarifies the large intervals associated with the learned distribution in Fig. 4b.

3) Impact of mislabeling on classification accuracy: Upon involving users in labeling task, the occurrence of some errors

TABLE II: Impact of mislabeling percentage on accuracy for User 2 with $B = 3$, $D = 21$

Err. Labels	UBAL	KAN	P-KAN
0%	77.31 ± 0.04	84 ± 0.03	83.52 ± 0.03
10%	73.16 ± 0.03	74.47 ± 0.04	77.57 ± 0.03
20%	66.26 ± 0.06	67.47 ± 0.05	71.05 ± 0.05

is inevitable. However, we expect minimal error from the user in labeling appliances, as this is not a difficult task and the user can simply abstain from labeling. In this experiment, we observe the strategies' robustness to low levels of mislabeling, i.e., 0%, 10% and 20%. The mislabeling percentage is the portion of the training labels that are incorrect.

For this experiment, we set the strategy parameters to $B = 3$ 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. 3b.

Table II demonstrates that UBAL is less affected by mislabeling when compared to the KAN methods. With a 10% noise level, UBAL loses 4.2% while KAN loses 9.5% in accuracy. This is likely because the K-Nearest-Neighbors algorithm is susceptible to noise when many instances are mislabeled, especially given our small value of $K = 3$. We expect that for larger values where $K > 3$, the KAN algorithms will be more robust to noise. As mentioned, we do not expect high amounts of error in this simple appliance labeling task, and expect abstention to be the larger component to this active learning scheme, for which the KAN algorithm is shown to be effective.

V. Conclusions

In this work, we address the challenge of creating 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, low-frequency smart outlet. KAN is characterized as a stream-based active learning algorithm that incorporates user behavior, in terms of query response distribution, in forming a realistic testbed for these smart outlet algorithms. We demonstrate that KAN, paired with a DTW-based KNN classifier, requires a much shorter training period than the previously established state of the art user-centric sampling method [14] to produce high accuracy classifiers. On average, with a budget of 6 queries per day, our system is able to reach 80% accuracy in seven days, while the uniform sampling method method requires fifteen days. This reduction in training period is attributed to the rate at which we learn the user's response distribution along with the highly representative training set achievable through KAN.

Acknowledgment

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] M. Ratner and C. F. Glover, *US energy: Overview and key statistics*. Congressional Research Service Washington, DC, 2014, vol. 40187.

[2] A. Schoofs, A. Guerrieri, D. T. Delaney, G. M. P. O'Hare, and A. G. Ruzzelli, "Annot: Automated electricity data annotation using wireless sensor networks," in *7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, June 2010, pp. 1–9.

[3] S.-C. Lee, G.-Y. Lin, W.-R. Jih, and J. Y.-J. Hsu, "Appliance recognition and unattended appliance detection for energy conservation," in *Proceedings of the 5th AAAI Conference on Plan, Activity, and Intent Recognition*, ser. AAAIWS'10-05. AAAI Press, 2010, pp. 37–44. [Online]. Available: http://dl.acm.org/citation.cfm?id=2908558.2908564

[4] S. Silvestri, D. A. Baker, and V. Dolve, "Integration of social behavioral modeling for energy optimization in smart environments," in *Proceedings of the 2nd International Workshop on Social Sensing*. ACM, 2017, pp. 97–97.

[5] H. Chang, W. Chiu, H. Sun, and C. Chen, "User-centric multiobjective approach to privacy preservation and energy cost minimization in smart home," *IEEE Systems Journal*, vol. 13, no. 1, pp. 1030–1041, March 2019.

[6] A. R. Khamesi, S. Silvestri, D. A. Baker, and A. D. Paola, "Perceived-value-driven optimization of energy consumption in smart homes," *ACM Transactions on Internet of Things*, vol. 1, no. 2, pp. 1–26, 2020.

[7] J. Zheng, D. W. Gao, and L. Lin, "Smart meters in smart grid: An overview," in *IEEE Green Technologies Conference (GreenTech)*, April 2013, pp. 57–64.

[8] 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. 16 838–16 866, 2012.

[9] O. Monnier, "White paper: A smarter grid with the internet of things," Texas Instruments, Tech. Rep., Oct. 2013. [Online]. Available: http://www.ti.com/lit/ml/slyb214/slyb214.pdf

[10] V. Abeykoon, N. Kankanamdurage, A. Senevirathna, P. Ranaweera, and R. Udawapola, "Real time identification of electrical devices through power consumption pattern detection," in *First International Conference on Micro and Nano Technologies, Modelling and Simulation*, March 2016.

[11] A. R. Khamesi, E. Shin, and S. Silvestri, "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, 2020, pp. 275–281.

[12] K. Fujii and H. Kashima, "Budgeted stream-based active learning via adaptive submodular maximization," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. USA: Curran Associates Inc., 2016, pp. 514–522. [Online]. Available: http://dl.acm.org/citation.cfm?id=3157096.3157154

[13] B. Settles, "Active learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2009.

[14] E. Serrao and M. Spiliopoulou, "Active stream learning with an oracle of unknown availability for sentiment prediction." in *IAL@ PKDD/ECML*, 2018, pp. 36–47.

[15] S. Yan, K. Chaudhuri, and T. Javidi, "Active learning from imperfect labelers," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. USA: Curran Associates Inc., 2016, pp. 2136–2144. [Online]. Available: http://dl.acm.org/citation.cfm?id=3157096.3157335

[16] S. Semwal, D. Joshi, R. S. Prasad, and D. Raveendhra, "The practicability of ica in home appliances load profile separation using current signature: A preliminary study," in *International Conference on Power, Energy and Control (ICPEC)*, Feb 2013, pp. 756–759.

[17] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and information systems*, vol. 7, no. 3, pp. 358–386, 2005.

[18] 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.

[19] B. Du, Z. Wang, L. Zhang, L. Zhang, W. Liu, J. Shen, and D. Tao, "Exploring representativeness and informativeness for active learning," *IEEE transactions on cybernetics*, vol. 47, no. 1, pp. 14–26, 2017.

[20] C. M. Grinstead and J. L. Snell, *Grinstead and Snell's introduction to probability*. Chance Project, 2006.