Machine Learning in the Wild: The Case of User-Centered Learning in Cyber Physical Systems

Atieh R. Khamesi, Eura Shin, Simone Silvestri
Computer Science Department,
University of Kentucky
Lexington, KY, USA
{atieh.khamesi,eura.shin}@uky.edu, silvestri@cs.uky.edu

***Abstract*—Smart environments, such as smart cities and smart homes, are Cyber-Physical-Systems (CPSs) which are becoming an increasing part of our everyday lives. Several applications in these systems, such as energy management through home appliance identification or activity recognition, adopt Machine Learning (ML) as a practical tool for extracting useful knowledge from raw data. These applications are usually characterized by a sequential stream of data, unlike the classical ML scenario in which the entire data is available during training. For such applications, Stream-based Active Learning (SAL) has been designed as a type of supervised ML in which an expert is asked to label the most informative instances as they arrive. Previous SAL techniques assume that the expert is always available and always labels the data correctly. However, in several applications, such as those mentioned above, the SAL activity interweaves with the everyday life of regular residents, who are often not experts, and may also not always be willing to participate in the labeling process. In this paper, we discuss the importance of user-centered ML, and show how taking into account realistic models of user behavior significantly improves the accuracy and reduces the training period of smart environment applications based on SAL. We consider two use cases, namely appliance identification and activity recognition. Results based on real data sets show an improvement in terms of accuracy up to 55.38%.**

***Index Terms*—Cyber-Physical-Systems (CPSs), Internet of Things (IoT), Smart City, Smart Home, Machine Learning, Active Learning, User-Centered**

I. Introduction

With the increasing technological development in computation and communication capabilities, *Cyber-Physical-Systems (CPSs)*, such as *smart cities* and *smart homes*, are growing in popularity [1], [2]. CPSs consolidate a physical component, like sensors and actuators, to a cyber component, such as software and algorithms, in order to create an integrated intelligent system capable of predicting and reacting to situations [3].

Accordingly, smart cities comprise several layers of technology including various methods of data collection and processing in order to efficiently manage resources such as energy and transportation [4]. A key component of a smart city are *smart homes*, namely homes with sensing, computation and communication capabilities able to provide residents comfort, security, and energy efficiency [5]. Nowadays, certain technologies are ubiquitous to smart homes, such as smart TVs, tablets, smartphones, and wearable devices. A defining characteristic of applications involving these technologies is the use of Machine Learning (ML) methods, as in the examples listed below.

- **Face Recognition:** Nowadays, not only home security systems but also smartphones utilize the face recognition technology to identify the faces of authorized users, for example home's residents or cell phone owner. Such systems deny access and in some cases send the picture of suspicious user to the authorized user [6].
- **Home Appliance Recognition:** For the purpose of fine grain energy management in smart homes, it is essential to recognize energy consumption of each individual appliance. Unlike smart meters which usually provide aggregated energy usage, smart outlets are able to collect data from the single appliance plugged in [7], [8].
- **Voice Recognition Technology:** These days, most of home automation systems are compatible with intelligent personal assistants like, Siri, Alexa or Google Home. To keep the system secure, such technologies should identify the legitimate user's voice. To this aim, voice recognition and natural language processing tools, such as Amazon Transcribe and Azure Custom Speech Service, are applied to recognize user's voice, verify the identity and then mine the meaning and operate the pertinent action.

The tradition supervised ML approach, in which a model is trained on an entire, established data set, is often not appropriate for such applications. In fact, these applications are characterized by a dynamic stream of data, and these data is often highly user-specific, which makes generalizability difficult. As an example, consider the task of home *appliance recognition*, where the goal is to infer the usage of home appliances from the electric signatures collected by smart outlets [9], [10]. Users have different appliances which generate different signatures, new appliances are continuously available on the market, and signatures are generated over time as appliances are used. Clearly, training cannot be done offline as in classic supervised ML, and data needs to be labeled as it arrives. A class of active learning approaches, namely *stream-based active learning* (SAL), have been proposed to deal with these scenarios [11]. According to SAL, an *expert* is available to label the data, and an algorithm is designed to select the most informative samples from the incoming stream to propose to the expert. Usually, a *budget* is used to limit the maximum

978-1-7281-3187-0/20/$31.00 ©2020 IEEE

TABLE I: ML Algorithms

Algorithms	Type	Task
K-Nearest Neighbors	Supervised	Classification
Support Vector Machine	Supervised	Classification
Linear Regression	Supervised	Regression
K-Means	Unsupervised	Clustering
Principal Component Analysis	Unsupervised	Feature extraction

number of samples to label.

In this paper we argue that the classical approach to SAL described above is also not suitable for applications *"in the wild"*, where the expert is just a regular user, as in the case of appliance recognition in smart environments. In these scenarios, users may *not be always available and engaged* in the labeling activity. Therefore, we claim that:

> It is essential to develop SAL and ML approaches for smart environment applications in particular - and Cyber-Physical Systems in general - by taking into account the limits of real non-expert users, through realistic user behavioral models of engagement, participation, and availability.

The remainder of this paper is organized as follows. In Section II, we review the ML techniques in CPSs. Section III studies the impact of real residents on ML techniques. Two use cases of user-centered ML will be explained in Section IV. Finally, Section V overviews the benefit of realistic user modeling and suggests future directions in this field.

II. Machine Learning Algorithms in Cyber-Physical Systems

ML techniques are well-suited for extracting usable knowledge from sensor readings, making them the method of choice in achieving the goals of CPSs. They have been extensively investigated in the literature. Specific to our context, [4] gives an overview of different ML techniques in IoT applications.

A. Classical ML Techniques

ML approaches are mainly categorized into *supervised*, *unsupervised* techniques which usually differ in terms of accuracy, complexity and their input data, referred to as the training set, which is a set of sample data used to prepare the model.

In supervised learning, instances in the training set are associated with labels, which are usually assigned by an expert. The labeling task is usually expensive and time consuming [12]. Unsupervised learning aims to identify similarities amongst the input data and performs so-called *clustering*. The training set in this case is not labeled. Some of the most common algorithms in this category include, k-means [13], hierarchical [14] and probabilistic clustering [15]. Table I provides some algorithms for these two ML categories.

Moreover, a comparison between supervised and unsupervised learning approaches from different perspectives is given in Table II. Briefly speaking, although supervised learning is more complex and more expensive, it generally achieves higher accuracy and reliability.

In the context of smart city applications, ML algorithms are used in order to analyze data from different sources such as traffic cameras, utility consumption, weather forecast etc. Authors in [16] used Regression Trees, Neural Networks and Support Vector Regression (SVR) to present a prediction mechanism for the parking occupancy. In another work, open source data has been adopted to perform real-time predictive analysis for energy management purposes [17].

Moreover, ML techniques have been used for fault detection.For event detection and activity classification in an indoor environment, like an office, home, public mall or airport, [18] compares Principal Component Analysis (PCA) and Canonical Correlation Analysis (CCA) to detect intermittent faults and mask such failures.

In the context of smart health, [19] demonstrated the improvement to mobile eHealth applications achievable through the use of Feed Forward Neural Netowrks, which improved performance with respect to network load, CPU and energy consumption.

One important application in smart homes is to perform energy management by identifying different home electric appliances [20]. Such appliances can be smart, capable of communicating with other devices, or not. To this aim, different ML techniques have been investigated [21]. In case of non-smart appliances, wireless outlets have been designed to collect corresponding data for analysis [22].

In the next section, we review the concept of Active Learning (AL) and explain two families of this technique, namely pool-based and stream-based AL (SAL).

B. Active Learning

As mentioned before, labeling the all instances of a training set is costly and requires an expert. However, in some cases, due to the huge amount of data, it is not practical to label every single instance. Therefore, Active Learning (AL) has been introduced to reduce labeling costs by selecting only the most informative instances for labeling [23], [12]. Briefly speaking, AL algorithms follow a selection criteria to query an expert on the most informative instances. This approach intelligently prevents the training phase from being overwhelmed by uninformative samples.

AL queries instances by taking advantage of past queries, i.e., already known responses (labels)[12]. To determine informativeness of the instance, some works, such as [24], [11], consider Uncertainty Sampling (US) as a selection criteria, in which instances with the highest uncertainty are chosen to be labeled. Some metrics used to measure *uncertainty* are maximum entropy [25], smallest-margin [26] and least confident [27].

[11], [28] studied a new class of AL from data streams. In such scenarios, the data volumes increase continuously. They proposed a classifier ensemble to predict newly arrived samples while only a small portion of stream data are labeled. Finally, [29] studies budgeted SAL where they investigate the possibility of converting existing AL methods to stream-

TABLE II: Comparison

	Supervised Learning	**Unsupervised Learning**
Input Data	Uses Known and Labeled Input Data	Uses Unknown Input Data
Computational Complexity	Very Complex in Computation	Less Computational Complexity
Real Time	Uses off-line analysis	Uses Real Time Analysis of Data
Number of Classes	Number of Classes is Known	Number of Classes is not Known
Accuracy of Results	Accurate and Reliable Results	Moderate Accurate and Reliable Results

based scenarios and provides theoretical guarantees on their performances.

All the above mentioned papers assume the presence of an *always available* expert willing to label the most informative instances. This is not always the case in a smart environments, where a regular resident is asked to perform the labeling. Such users are often not experts and may not always be willing to respond upon query. The problem of not only selecting the best instances, but also adapting to the user willingness to respond, has been first introduced in [28]. However, the authors assume the presence of a random oracle, or an expert that is willing to respond to the queries uniformly at random. Not surprisingly, the authors show that a random sampling strategy is the best in this specific context.

In the following, we describe two families of AL, namely pool-based and stream-based AL. In most AL algorithms [30], [31], [32], [33], a fixed set of unlabeled instances, called a pool, is given to the learner in order to find the most informative instances to label. At this point, the expert gives the corresponding labels to the learner and then an updated classifier is made based on all the labeled samples so far. However, in smart environment applications, sensor measurements are usually collected continuously along the time, which can be interpreted as a big stream of data. Therefore, stream-based AL has been introduced to deal with this sort of data [11].

In the case of sequential data arrival, the learner observes a stream of unlabeled instances, where it should decide whether to ask the expert for labeling of incoming instance and so use this sample for learning process or not.

Several strategies have been proposed for SAL [11], [34], [23], [28] most of which follow a single criterion to pick informative instances. [35], [36], [37] show that this method limits the performance of AL, known as exploitation-exploration dilemma. Specifically in a stream-based scenario, it is destructive to select data for labeling when it does not depute the characteristics of original sample properly [38]. In more detail, authors in [38] propose a reinforcement learning framework for stream-based data in order to learn the optimal strategy during the labeling process. They adopt the feedback from the classifier to refine the selecting criteria.

III. Impact of real users

As mentioned before, unlike classical ML approaches, in various smart environments applications, ML techniques are in direct contact with the regular residents and needs them to perform the labeling task. In this respect, several challenges arise, first, the human mistake is inevitable. In the context of ML, such mistakes can be interpreted as noisy labels. Second, the residents may not always be available or willing to perform labeling.

TABLE III: Summary of datasets

Dataset	**Num. Instances**	**Num. Features**	**Num. Classes**
Appliance [10]	419	35	6
Activity [44]	692	351	12

A. Noisy Labels

In a typical supervised learning scenario, a set of training instances are mapped to a set of labels in a 1-to-1 relationship. Noisy labels occur when one instance is mapped to multiple labels [39], [40], the mapping is incorrect [41], or a combination of the two scenarios. With the growth of crowd-sourcing techniques, such as Amazon Turk, and applications dependent on human interaction, such as in CPSs, noise reduction is becoming increasingly relevant to ML. Although an exploration of label noise is beyond the scope of this paper, we maintain that it is crucial to realistic user modeling and will be included in future work.

B. Abstention from labeling

Unlike most AL approaches in which the expert is always available for labeling, authors in [42], [43] investigate the a pool-based AL where the expert abstains from labeling and also may give noisy labels. In these works, the learner keeps repeatedly querying instances where the expert abstains. Moreover, abstention is modeled by a certain probability which increase as the query points get closer to the decision boundary. However, this method can not be applied to SAL where the instances show up sequentially. Therefore, our work adopts a novel strategy which is distinct from the previous ones.

IV. Case Study

In this section, we investigate two smart environments applications, namely *appliance recognition* and *physical activity monitoring*, using the public data sets, available in [10] and [44], respectively. Both applications are SAL examples which are in direct contact with the residents and require them for labeling.

A. Experimental Design

1) Stream-based Active Learning Scenarios: A smart device may enable the residents to monitor and control energy usage in the smart homes. In order to perform accurate analysis, such smart systems must be robust to different types of electrical appliances. This challenge presents itself in a typical stream-based learning scenario where new appliances are learned by the system as they are used.

In addition, stream-based learning is advantageous in physical activity monitoring because it is able to learn activities that are specific to the user.

2) Data: The public data sets on appliance recognition [10] and activity monitoring [44] are outlined in Table III. The basic features, extracted from the high frequency signal data, include mean, maximum, and variance of each signal.

Additionally, we applied the following transformations to the signals: Fourier Transform, the Power Spectral Density, and auto-correlation. The x-y coordinates at the n highest peaks of these transformations form the feature set. Semantically, these peaks represent the frequency and corresponding amplitude of oscillations in the signal.

The activity data consists of more features than the appliance data because of the complexity of the sensing technology. While the appliance monitor only extracted current and voltage information, the wearables tracked 50 separate inertial measurement units. To account for dimensionality, we set $n = 3$ and $n = 1$ for the appliance and activity sets, respectively.

3) Model: In this model, time is discretely divided into 24 slots corresponding to 24 hours per day, denoted as h. In interacting with the system as labeler, the resident is assumed to follow a certain response distribution according to his willingness/availability. Let $P(h)$ be the probability that the resident will successfully respond to a query at hour h, independent of other time slots. The learner is given D days of training and a budget of B queries to the resident per day. B is a design parameter that can be adjusted to the resident's preference.

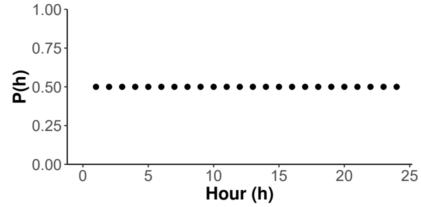
For simplicity, we assume that only one instance arrives per hour. This means that over D days, there are $N = D \times 24$ instances observed by the ML system. In our experiments, the N instances are sampled with replacement from the training set and randomly assigned an hour h.

4) User Distributions: We consider three different residents that follow a uniform, alternating, and Gaussian user distribution, as shown in Figure 1. Finally, for fair comparison, we consider the same *resident abstention rate* for all distributions, such that $P(0 \leq h \leq 24) = 0.5$.

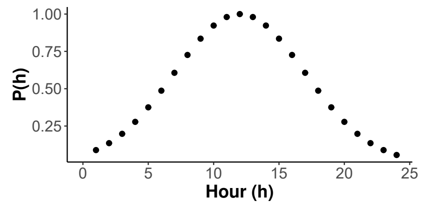
5) Sampling Strategies: Through these experiments, we show that higher accuracy is achieved by modeling the user distribution. We compare the Random Sampling (RS) strategy to a User Aware Sampling (UAS) strategy, where the true user distribution is known to the ML system. Given a user distribution $P(h)$, hour h, random number r in $[0, 1]$, UAS will query the resident if $r \leq P(h)$. The RS strategy will simply choose B hours at which to query the resident for the day.

B. Results

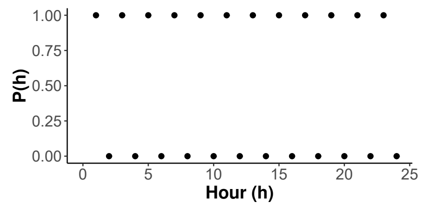
This experiment models a training period of three weeks, i.e., 21 days. We compare the performance of the RS to the UAS strategy for the three defined user distributions. As expected, the uniform distribution looks similar for both RS



(a) Resident 1: Uniform



(b) Resident 2: Gaussian



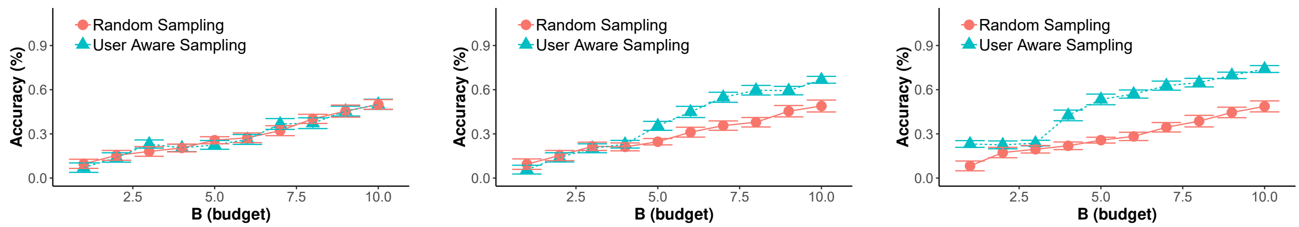
(c) Resident 3: Alternating

Fig. 1: Response distribution $P(h)$ for three different residents

and UAS. This is because random sampling is the uniform distribution; in such distribution, there is not much room for improvement. However, when the user distribution follows a non-uniform pattern, such as alternating or Gaussian, we see queries guided by knowledge of the user's behavior results in higher accuracy.

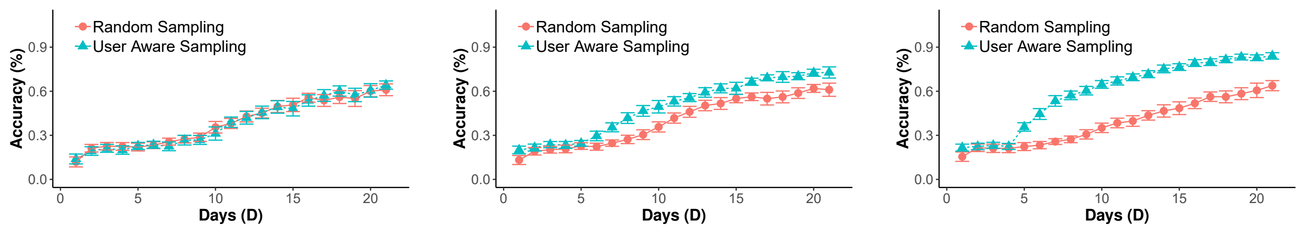
It should be noted that the UAS strategy achieves higher accuracy in *shorter training period* than the RS strategy. In Figure 3, we see that the UAS strategy begins to separate from RS by day 6 of the Gaussian distribution and day 5 of the alternating distribution. With enough days, both strategies are expected to observe enough labeled instances to plateau at similar upper-bound accuracy. However, UAS is able to reach this higher accuracy quicker, which in a real-life smart environment, is more convenient to the residents.

Table IV displays the mean difference between the RS and UAS strategies over the course of 21 days. Graphically, this is the average "distance" between the RS and AUS curves displayed in Figure 3. In Figure 2, we see a clear difference in accuracy between the two strategies starting at values of $B = 3$. For this reason, Table IV and V display the accuracy gain for budget values $B \in [3, 10]$. We perform a t-test on the difference of means between the RS and UAS strategies, such that $h_a : \mu_{\mathrm{UAS}} > \mu_{\mathrm{RS}}$. We apply the Bonferroni correction to account for the eight budgets and three distributions that are being tested. Therefore, a statistically significant result will have $pval < \frac{0.05}{3 \times 8} = 0.002$. As expected, UAS outperforms RS for both the Gaussian and alternating user distributions. Note that in the uniform distribution, UAS is considered a



(a) Uniform (b) Gaussian (c) Alternating

Fig. 2: Appliance data comparison of RS and UAS strategy for different budgets B after one week ($D = 7$)



(a) Uniform (b) Gaussian (c) Alternating

Fig. 3: Appliance data comparison of RS and UAS strategy for $B = 5$ over 21 days

TABLE IV: Appliance data mean difference in accuracies between UAS and RS strategies. Statistically significant values are highlighted.

	Distribution		
Budget	Uniform	Gaussian	Alternating
3	0.90	1.20	16.19
4	0.70	4.6	22.35
5	-0.16	9.28	27.75
6	-0.45	13.04	32.99
7	0.20	18.32	38.53
8	-1.16	23.77	44.29
9	-1.76	24.82	49.73
10	-0.70	28.71	55.58

TABLE V: Activity data mean difference in accuracies between UAS and RS strategies. Statistically significant values are highlighted.

	Distribution		
Budget	Uniform	Gaussian	Alternating
3	-0.74	0.96	17.17
4	0.32	3.93	21.59
5	0.33	8.78	27.56
6	-5.3e-16	14.37	32.51
7	0.80	18.02	38.87
8	-0.86	23.64	43.98
9	-1.10	26.05	49.42
10	-2.6	28.11	55.16

significant improvement on RS for certain budget values. For these cases, the corresponding difference in accuracy is a small positive number. This is likely a result of the significance tests themselves, which are set to identify a positive difference in means, but not the *scale*, or amount, of the difference.

V. Conclusion and Future Works

In this work, we studied the impact of residents engagement in smart environments through ML techniques. Such residents which are supposed to perform labeling, are often not experts, and may also not always available. To demonstrate the importance of realistic user modeling, we provide a case study that compares the performance of Random Sampling (RS) and User Aware Sampling (UAS) for three different user response distributions on two real life data sets. Our results, which include an accuracy boost of up to 55.58%, demonstrate that an accurate user model will result in a smart system that achieves high accuracy at a quicker rate, i.e., shorter training period. According to the result of this work, considering a user-centered ML allows creating a system that learns quicker and enables us to provide an easily deployable, high performing CPS to the residents.

Future works may expand upon user modeling in two main areas. First, it is necessary to establish a strategy to learn the user distribution. Second, the possibility of noisy labels must be incorporated in the model which will introduce many areas for exploration. For example, in this paper we noted that a system trained on RS will achieve an accuracy similar to UAS with enough time. However, with noisy labels, a user-aware system may also consider noise reduction in addition to availability as a querying criteria to the user. While a blind RS strategy will use all the labeled instances it receives, a user-aware model may cherry pick the most representative instances in training, resulting in classification performance that may only be achievable with realistic user-modeling.

VI. Acknowledgements

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

References

[1] M. A. Kafi, Y. Challal, D. Djenouri, M. Doudou, A. Bouabdallah, and N. Badache, "A study of wireless sensor networks for urban traffic monitoring: Applications and architectures," *Procedia Computer Science*, vol. 19, pp. 617 – 626, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S187705091300690X

[2] Y. Qin, Q. Z. Sheng, N. J. Falkner, S. Dustdar, H. Wang, and A. V. Vasilakos, "When things matter: A survey on data-centric internet of things," *Journal of Network and Computer Applications*, vol. 64, pp. 137 – 153, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804516000606

[3] G. Spezzano and A. Vinci, "Pattern detection in cyber-physical systems," *Procedia Computer Science*, vol. 52, pp. 1016 – 1021, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050915008960

[4] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for internet of things data analysis: a survey," *Digital Communications and Networks*, vol. 4, no. 3, pp. 161 – 175, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S235286481730247X

[5] T. YAMAZAKI, "The ubiquitous home," *International Journal of Smart Home*, vol. 1, no. 1, pp. 17–22, 2007. [Online]. Available: https://ci.nii.ac.jp/naid/10026029745/en/

[6] E. Osuna, R. Freund, and F. Girosit, "Training support vector machines: an application to face detection," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1997, pp. 130–136.

[7] V. Dolce, C. Jackson, S. Silvestri, D. Baker, and A. De Paola, "Social-behavioral aware optimization of energy consumption in smart homes," in *IEEE DCOSS*, 2018, pp. 163–172.

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

[9] 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.

[10] T. Picon, M. Nait Meziane, P. Ravier, G. Lamarque, C. Novello, J.-C. Le Bunetel, and Y. Raingeaud, "COOLL: Controlled on/off loads library, a public dataset of high-sampled electrical signals for appliance identification," *arXiv preprint arXiv:1611.05803 [cs.OH]*, 2016.

[11] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," *Machine Learning*, vol. 28, no. 2, pp. 133–168, Aug 1997. [Online]. Available: https://doi.org/10.1023/A:1007330508534

[12] B. Settles, "Active learning literature survey," Tech. Rep., 2010.

[13] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, no. 2, pp. 451 – 461, 2003, biometrics. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320302000602

[14] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, Sep 1967.

[15] T. Hastie, R. Tibshirani, and J. Friedman, "Unsupervised learning," in *The elements of statistical learning*. Springer, 2009, pp. 485–585.

[16] Y. Zheng, S. Rajasegarar, and C. Leckie, "Parking availability prediction for sensor-enabled car parks in smart cities," in *2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, April 2015, pp. 1–6.

[17] W. Derguech, E. Bruke, and E. Curry, "An autonomic approach to real-time predictive analytics using open data and internet of things," Dec 2014, pp. 204–211.

[18] D. N. Monekosso and P. Remagnino, "Data reconciliation in a smart home sensor network," *Expert Systems with Applications*, vol. 40, no. 8, pp. 3248 – 3255, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S095741741201278X

[19] F. Ramalho, A. Neto, K. Santos, J. B. Filho, and N. Agoulmine, "Enhancing ehealth smart applications: A fog-enabled approach," in *2015 17th International Conference on E-health Networking, Application Services (HealthCom)*, Oct 2015, pp. 323–328.

[20] S. Kim, "Appliance recognition unit for home energy management system with upnp network," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2794–2803, Dec 2017.

[21] V. Abeykoon, N. Kankanamdurage, A. Senevirathna, P. Ranaweera, and R. Udawapola, "Real time identification of electrical devices through power consumption pattern detection," 03 2016.

[22] Currant, "Wifi smart outlet [product]," 2018, accessed: 10-31-2018.

[23] Y. Fu, X. Zhu, and B. Li, "A survey on instance selection for active learning," *Knowledge and Information Systems*, vol. 35, no. 2, pp. 249–283, May 2013. [Online]. Available: https://doi.org/10.1007/s10115-012-0507-8

[24] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ser. COLT '92. ACM, 1992, pp. 287–294. [Online]. Available: http://doi.acm.org/10.1145/130385.130417

[25] I. Dagan and S. P. Engelson, "Committee-based sampling for training probabilistic classifiers," in *In Proceedings of the Twelfth International Conference on Machine Learning*. Morgan Kaufmann, 1995, pp. 150–157.

[26] T. Scheffer, C. Decomain, and S. Wrobel, "Active hidden markov models for information extraction," in *Advances in Intelligent Data Analysis*, F. Hoffmann, D. J. Hand, N. Adams, D. Fisher, and G. Guimaraes, Eds. Springer Berlin Heidelberg, 2001, pp. 309–318.

[27] A. Culotta and A. McCallum, "Reducing labeling effort for structured prediction tasks," in *Proceedings of the 20th National Conference on Artificial Intelligence - Volume 2*, ser. AAAI'05. AAAI Press, 2005, pp. 746–751. [Online]. Available: http://dl.acm.org/citation.cfm?id=1619410.1619452

[28] X. Zhu, P. Zhang, X. Lin, and Y. Shi, "Active learning from data streams," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 757–762.

[29] 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. Curran Associates Inc., 2016, pp. 514–522. [Online]. Available: http://dl.acm.org/citation.cfm?id=3157096.3157154

[30] N. Abe and H. Mamitsuka, "Query learning strategies using boosting and bagging," in *Proceedings of the Fifteenth International Conference on Machine Learning*, ser. ICML '98. Morgan Kaufmann Publishers Inc., 1998, pp. 1–9. [Online]. Available: http://dl.acm.org/citation.cfm?id=645527.657478

[31] Y. Baram, R. El-Yaniv, and K. Luz, "Online choice of active learning algorithms," *J. Mach. Learn. Res.*, vol. 5, pp. 255–291, Dec. 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=1005332.1005342

[32] D. D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *In Proceedings of the Eleventh International Conference on Machine Learning*. Morgan Kaufmann, 1994, pp. 148–156.

[33] N. Roy and A. McCallum, "Toward optimal active learning through sampling estimation of error reduction," in *ICML*, 2001.

[34] S. Dasgupta, A. T. Kalai, and C. Monteleoni, "Analysis of perceptron-based active learning," *J. Mach. Learn. Res.*, vol. 10, pp. 281–299, Jun. 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1577069.1577080

[35] H. T. Nguyen and A. W. M. Smeulders, "Active learning using pre-clustering," in *International Conference on Machine Learning*, 2004, pp. 623–630. [Online]. Available: https://ivi.fnwi.uva.nl/isis/publications/2004/NguyenICML2004

[36] S. Huang, R. Jin, and Z. Zhou, "Active learning by querying informative and representative examples," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 10, pp. 1936–1949, 2014.

[37] S. Ebert, M. Fritz, and B. Schiele, "Ralf: A reinforced active learning formulation for object class recognition," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 3626–3633.

[38] Y. Cheng, Z. Chen, L. Liu, J. Wang, A. Agrawal, and A. Choudhary, "Feedback-driven multiclass active learning for data streams," in *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, ser. CIKM '13. ACM, 2013, pp. 1311–1320. [Online]. Available: http://doi.acm.org/10.1145/2505515.2505528

[39] Y. Yan, R. Rosales, G. Fung, R. Subramanian, and J. Dy, "Learning from multiple annotators with varying expertise," *Machine learning*, vol. 95, no. 3, pp. 291–327, 2014.

[40] M.-L. Zhang, B.-B. Zhou, and X.-Y. Liu, "Partial label learning via feature-aware disambiguation," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1335–1344.

[41] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Learning with noisy labels," in *Advances in neural information processing systems*, 2013, pp. 1196–1204.

[42] S. Yan, K. Chaudhuri, and T. Javidi, "Active learning from noisy and abstention feedback," in *53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sep. 2015, pp. 1352–1357.

[43] ——, "Active learning from imperfect labelers," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Curran Associates Inc., 2016, pp. 2136–2144. [Online]. Available: http://dl.acm.org/citation.cfm?id=3157096.3157335

[44] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Wearable Computers (ISWC), 2012 16th International Symposium on*. IEEE, 2012, pp. 108–109.