Leveraging User Expertise in Collaborative Systems for Annotating Energy Datasets

Hông-Ân Cao, Felix Rauchenstein
Department of Computer Science
ETH Zurich, Switzerland
Email: hong-an.cao@inf.ethz.ch,
rafelix@student.ethz.ch

Tri Kurniawan Wijaya, Karl Aberer
Department of Computer Science
EPFL, Switzerland
Email: {tri-kurniawan.wijaya,
karl.aberer}@epfl.ch

Nuno Nunes
Madeira Interactive Technologies Institute
Funchal, Portugal
Email: njn@uma.pt

***Abstract*—While tasks such as segmenting images or determining the sentiment expressed in a sentence can be assigned to regular users, some others require background knowledge and thus, the selection of expert users. In the case of energy datasets, acquiring data represents an obstacle to develop data-driven methods, due to prohibitive monetary and time costs linked to the instrumentation of households in order to monitor the energy consumption. More so, most datasets only contain pure power time series, despite labels being required to determine when a device is in use from when it is idle (incurring stand-by consumption or being off), and by extension to separate human activities triggering the consumption from the baseline consumption. We build upon our Collaborative Annotation Framework for Energy Datasets (CAFED) to evaluate and distinguish the performance of expert users against that of regular users. Through a user study with curated benchmark annotation tasks, we provide data-driven and efficient techniques to detect weak and adversarial workers and promote users when the contributors' user-base is limited. Additionally, we show that if carefully selected, the seed gold standard tasks can be reduced to a small number of tasks that are representative enough to determine the user's expertise and predict crowd-combined annotations with high precision.**

***Index Terms*—Time series analysis; Data mining; Information search and retrieval; Collaboration; Crowdsourcing; Smart energy; Smart meters; Energy data analytics; Datasets; Algorithms**

I. Introduction

The development of learning algorithms entices the usage of data to improve and evaluate the accuracy of their outcome. Before the spread of online platforms, acquiring ground truth data was tedious as it was difficult and costly to recruit workers to perform specific tasks. These were then often solved by benevolent lab mates and it took considerable time to collect those datasets. Nowadays, the majority of the micro-tasks that are present on Amazon Mechanical Turk or CrowdFlower consist of image and text labeling and have contributed to build large scale datasets that have allowed progress in the fields of computer vision and natural language processing. However, the introduction of a monetary gain instead of the benevolence of fellow researchers or acquaintances to label such data can lead to the abuse of the system to increase workers' remuneration, at the expense of the quality of the data.

While obtaining labels for text or image content can be distributed to a larger audience of workers due to the nature of the tasks themselves, and can piggyback on existing systems such as CAPTCHAs, crowdsourcing tasks for different fields such as labeling genes or locating volcanoes in satellite images would require domain knowledge expertise that is not widely available to the general public. Energy analytics, where data are obtained through the instrumentation of households to obtain power data from dwellings, has benefited from the adoption of smart meters to replace semesterly or yearly reporting. These enabled the release of datasets collected by different research institutes and organizations with household-level aggregated load consumption at finer granularity. However, for the development of human activity-level or more generally event-based algorithms linked to the consumption of energy caused by households' residents, more labels that can be used for training and testing the algorithms are required. This is due to the fact new datasets have to be collected to include more appliances and real-time annotations from the residents: existing datasets have the shortcomings of having either been collected at coarser time granularities, for shorter periods, including few appliances (sometimes having only aggregated household consumption) or simply without event-based labels (appliances' states or human activities). High monetary costs to successfully and reliably carry out data collections have hindered the advances in this domain. They are mostly related to the complexity in instrumenting households: the type of electrical appliances and the electrical wiring can force the sub-metering to be performed at the circuit level, requires expensive hardware and the assistance of certified electricians, preventing the usage of cheaper alternatives such as smart plugs that can be inserted between the appliance's socket and the electrical outlet. Our Collaborative Annotation Framework for Energy Datasets (CAFED)[1] [1] represented the first effort to retrofit labeling on an existing dataset by leveraging the wisdom of domain experts to annotate an appliance as being *active* or *idle*, based on the time series representing its power consumption.

The contribution of this paper consists in i) providing a

[1]https://cafed.inf.ethz.ch

978-1-4673-9005-7/16/$31.00 ©2016 IEEE

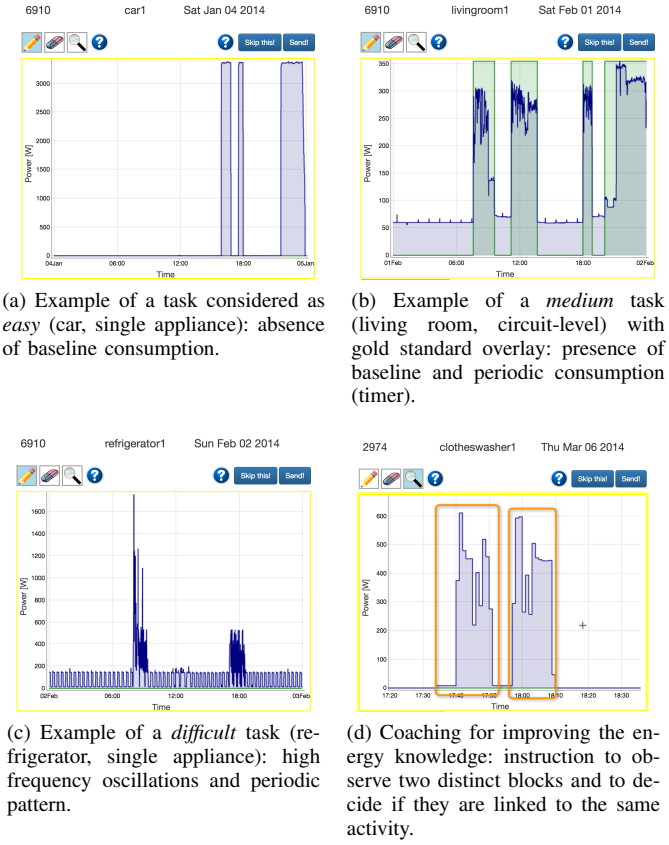
thorough evaluation of the performance of general users in comparison with experts in the labeling of energy time series and ii) showing data-driven approaches to address quality issues with crowdsourcing. Due to the low availability of users that can contribute to such system, we leverage online methods to adaptively evaluate and adjust the score of a user to the difficulty of the task. This salvages as many annotations as possible and promotes promising users, while guaranteeing the quality of the system, by being able to detect weak or adversarial contributors rapidly. Moreover, if the gold standard tasks are well curated, using domain knowledge, and accounting for the tasks' difficulty, few of these tasks are necessary to evaluate the users' expertise levels. These can be used to predict crowd-combined annotations with high accuracy. This shows that the research in the energy domain can benefit from paradigms and advances in big data by leveraging crowdsourcing to collect and consolidate datasets and benefit from the wisdom of the crowd. In the following, we will review the related work in Section II, then we will present the user study for collecting the data in Section III. We will then detail our methods for scoring the users based on their performance in Section IV, present the results of our analysis in Section V and conclude in Section VI.

II. Related Work

EM [2] has been used to provide scores to evaluate the quality of the labeling of categorical data to separate between error and bias [3], and to compare expert (geologists') annotations against the performance of an algorithm in the case of images [4]. The performance of experts and non-expert users has been evaluated for natural language processing tasks and using expert annotated data to correct the annotation bias [5]. Probabilistic models have been used for inferring labels for images when the expertise of the annotators is unknown and the difficulty of the task is accounted for [6], [7].

Expanding the realm of tasks that can be solved by crowdsourcing suggests taking steps towards improving the workflow, the design of the tasks to be assigned to the workers, valuing the participation of trusted users, expanding the existing platforms by integrating machine learning and AI techniques to improve quality [8]. Recent work has established that behavioral cues based on the interaction with the platform for information retrieval tasks is more successful at detecting fraudulent interactions when compared to baseline gold standard tasks solved by experts [9]. Splitting the annotations of information retrieval into a training phase and a test phase has been surveyed [10]. Socio-demographic features have also been leveraged to isolate high quality workers for solving multiple choices questions [11]. To improve the quality of the contributions, gamification techniques have shown to be more successful than filtering the workers based on their countries [12].

Energy datasets are mostly constituted of time series of power measurements. The development of algorithms for extracting knowledge such as the states of appliances from these data requires ground truth labels. The monetary costs involved in instrumenting households to obtain energy measurements [13] have hindered the apparition of new data collections to mitigate the absence of events that triggered the energy consumption. Crowdsourcing has not been used extensively for acquiring time series labeling, despite some initiatives for having expert-annotated data such as the CAFED platform [1]. The type of tasks differs from previous crowdsourcing initiatives as they require some background knowledge to be solved successfully and are more straining than the classical image or text labeling micro-tasks, which can be carried out in the form of multiple choice questions or entering a value in a text box and is fitting for existing platforms such as Amazon Mechanical Turk or CrowdFlower, but not for time series data.



(a) Example of a task considered as *easy* (car, single appliance): absence of baseline consumption.

(b) Example of a *medium* task (living room, circuit-level) with gold standard overlay: presence of baseline and periodic consumption (timer).

(c) Example of a *difficult* task (refrigerator, single appliance): high frequency oscillations and periodic pattern.

(d) Coaching for improving the energy knowledge: instruction to observe two distinct blocks and to decide if they are linked to the same activity.

Fig. 1: Tasks' difficulty levels and coaching on CAFED.

III. User Study

For the purpose of this work, we ran a user study to compare the performance in annotating power curves by domain experts and non-experts. The manual labeling consists in indicating on a daily time series that represents a single appliance or a circuit (e.g. data acquired through a power strip or at the room level), when the appliance is *actively* used (being triggered on by a resident) or *idle* (being in stand-by mode or off, and thus only exhibiting baseline power consumption) [1], [14], [15]. We recruited 7 users from different education backgrounds with diverse levels of familiarity with the energy

jargon, with ages ranging from 16 to 50. Our group of users consisted of people with little to no knowledge about the energy domain and some having a scientific background, but not having worked in the field. They were selected for their interest in improving the energy efficiency in households, their motivation for taking part in this experiment and solving these annotation tasks without remuneration. The sample, although being small, offers a plausible approximation of non-adversarial workers available on a traditional crowdsourcing platform with varying degrees of familiarity with the energy domain, diverse education backgrounds, diverse occupations and different age groups. The setup of the experiment is representative of the current situation with research fields where the access to domain experts is limited, but users with general and common knowledge is large. Additionally, we gathered 3 experts who have extensively worked with power data. The experts' contributions were used to create the ground truth or gold standard, which could then be compared to the regular users' labeling. As a benchmark, we selected 30 energy curves from different appliances from the Pecan Street dataset and with varying degrees of difficulty (that would require domain knowledge to solve more accurately) on the CAFED platform[2] [1]. As can be seen in Figure 1, we determined i) *easy* tasks as tasks without baseline consumption, where the *active* consumption would consist in everything above 0 [W] as in Figure 1a, ii) *medium* tasks require additional knowledge such as the presence of baseline consumption, or context (type of appliance or circuit) as in Figure 1b, and iii) *difficult* tasks rely on the detection of periodic patterns with high frequency oscillation such as fridges and the mechanical functioning of an appliance (in the case of a fridge, the compressor or in the case of multi-state appliances such as dishwashers, being able to link the consumption to different stages in the washing process) as in Figure 1c.

The annotation task can be time consuming as the curves have a 1-minute granularity, meaning that 1440 data points per daily curve have to be annotated and that careful annotators are required to meticulously inspect the curves and zoom in and out to decide when to transition from *active* to *idle* and conversely. Each task would therefore be completed within a few seconds to a few minutes, depending on its difficulty level. The curves were labeled both by the experts and the regular users.

We ran a controlled experiment for the regular users' group with different stages, to assess their performance in more details. This consisted of four phases described in the following.

- Phase 1 was a survey to collect background information about the familiarity and knowledge levels of our participants in regards to the energy jargon and the functioning of appliances.
- Phase 2 consisted in the solving of the 30 predefined annotation tasks using the CAFED platform [1]. In order not to influence the participants, they were given just as much information as they needed to interact with the platform itself through a quick start video tutorial, but they were not properly introduced to the problem of energy dataset labeling.
- Phase 3 offered user coaching. We sat down with each user and discussed their experience with Phase 2. Additionally, we properly presented the users with knowledge about energy datasets. In particular, we tutored them to improve how they annotate the time series by making use of existing information displayed on the platform and deepening their knowledge about the electrical consumption of appliances as in Figure 1d. Some examples of what they were made aware of:
  - they should make use of the appliance's type and their knowledge about how it operates;
  - stand-by consumption of different appliances occurs even when the devices are idle;
  - they should identify periodicity by looking at the time axis in the annotation panel and make use of the curves extracted from the same week in the right panel to distinguish extraordinary patterns from normal functioning;
  - fluctuating power consumption depends on the usage and the context (e.g. heater in cold or hot room);
  - electricity patterns can exhibit fluctuations due to metering noise or inner circuits;
- Phase 4 was a repetition of Phase 2, after having acquired the necessary know-how from the coaching session. This round simulated having a group of more experienced users, already knowing how to interact with the platform and having some basic knowledge about energy dataset labeling. In a real-life deployment, this strategy would be implemented when designing the crowdsourcing task by displaying tutorial videos or similarly before having the crowd annotate the data. We expect the second annotation session to have improved the quality of the labeling. This would imply that some guidance is needed in order to get acceptable results.

In order to distinguish experts' from non-experts' work, we incorporated additional features that were collected during the study. Not only did we record the time spent to solve each tasks, but also the mouse movements to highlight the users' interaction with the platform (the usage of the annotation tools, side panel with additional days, etc.), which can also be used in the next section.

IV. Methodology

In the following, we show how the data obtained through the user study are treated and analyzed. We attempt to compare and analyze non-expert against expert users in three different manners. Our goal is to detect weak workers to either ignore their input as soon as possible or ban them from the system. In a system with few potential users, we want to promote good workers, by quantifying their annotation quality and accounting for its fluctuation. We focus on the online scoring

[2]https://cafed-study.inf.ethz.ch/

of users in the order in which the tasks are solved and the classification of non-expert and expert contributions. For these two aspects of our analysis, we present the features that can be extracted from the data. Then, we explore how to combine different users' annotations based on their expertise level and to predict how the wisdom of the crowd's performs against the experts' annotations.

We first describe the features that can be extracted from the data and then explain how they are used in the online scoring, the classification and the prediction. The gold standard is extracted from the annotations provided by the expert contributors by applying majority voting, this allows to consolidate the experts' annotations by enabling consensus [1].

A. Features

We distinguish scoring features, which compare a regular user's work against the gold standard computed from the experts' labeling, from features extracted from single annotation tasks.

1) Scoring: To distinguish the regular users' from the experts' work, we score their contribution against the gold standard produced by the experts. These scores are computed for each user and each task individually. They quantify the worker's quality reflected by their performance in annotating the benchmark curves by looking at the accuracy of the outcome in comparison with the gold standard. Additionally, the variety of scores addresses potential malicious contributions by considering different attack scenarios. In the following, we will formally define the scores that can be extracted by comparing the users' annotations.

a) Hamming distance: The annotated daily curves are binary vectors. To compare different annotators' performances for the same curve, we can turn to distance measures for binary vectors. To compare two binary vectors x and y of dimension d (representing when an appliance is considered *active* as 1 or *idle* as 0), one such measure is to compute their Hamming distance as described in Equation 1.

$$d_H(x, y) = \sum_{k=0}^{d-1} |y_k - x_k| \tag{1}$$

We propose a score based on the Hamming distance: the percentage of correctly annotated minutes per daily curve. The score as described in Equation 2 is computed for each task i and user j.

$$score_{H_{i,j}} = 1 - \frac{d_H(t_{i,j}, g_i)}{d} \tag{2}$$

This score relies on the observation that most appliances are not always *active* (they are either in stand-by mode or off most of the day). This is reflected in how the curves should be annotated, as the majority of the time, the appliance or circuit should be considered *idle*, and the annotation binary vector should have by extension a majority of zeros. If our score took the whole annotation binary vector in consideration, it would allow a user who had provided minimal effort and labeled the curve as all *idle* (thus zeros) to achieve a high score.

To prevent such attack scenario from being unnoticed, we focus on the proportion of *true positives* labeling over the vector's length d in comparison with the manually annotated ground truth provided by the experts, instead of being biased by the *true negatives* proportion. We define the $score_{H_{i,j}}$ for the annotation $t_{i,j}$ for task i and provided by user j and its corresponding gold standard g_i as in Equation 2.

b) Focusing on the classification confusion matrix: We address the case of a user who provides annotations with zeros only, by accounting only for the *true positives* and thus drastically decreasing their score. However, a user who would instead only provide annotations with ones would achieve the highest score of 1. They are indeed guaranteed to correctly classify all of the *active* sections of the curve, although this labeling is comparably as damaging as the all-zeros scenario. To prevent this, we leverage the classification confusion matrix for additional scores. This is why we consider the *true positives* TP as another score.

Additionally, we can leverage i) the *false negatives* FN, where the user annotates an appliance as being *idle*, although it is considered *active* by the experts, ii) the *false positives* FP, where conversely, the appliance is annotated as *active*, despite being marked as *idle* by the experts.

We focus on comparing the parts annotated as *active* by the user and those annotated as *active* in the gold standard. This means using the *true positives* TP, *false negatives* FN and *false positives* FP, to define the ratio of parts correctly annotated as *active*. For the annotation $t_{i,j}$ for task i and provided by user j, we define the $score_{A_{i,j}}$ as in Equation 3.

$$score_{A_{i,j}} = \frac{TP_{i,j}}{TP_{i,j} + FP_{i,j} + FN_{i,j}} \tag{3}$$

We can also make use of traditional machine learning scores such as the precision as defined in Equation 4, the recall as in Equation 5 and the $F1$-score as in Equation 6 for task i's annotation $t_{i,j}$, provided by user j.

$$precision_{i,j} = \frac{TP_{i,j}}{TP_{i,j} + FP_{i,j}} \tag{4}$$

$$recall_{i,j} = \frac{TP_{i,j}}{TN_{i,j} + FP_{i,j}} \tag{5}$$

$$F1_{i,j} = 2 * \frac{precision_{i,j} * recall_{i,j}}{precision_{i,j} + recall_{i,j}} \tag{6}$$

c) WAVE algorithm: The previous scores were defined per task and per user. To compare multiple users' performances for one specific task, we use the weight-adjusted voting algorithm for ensembles of classifiers (WAVE) [16], [17]. The algorithm takes as input an $m * d \times n$ matrix of n vectors v_i, where for each user i

$$v_{i_k} = \begin{cases} 1 & \text{if the user annotated the } k^{th} \text{ data point correctly} \\ 0 & \text{otherwise} \end{cases}$$

and computes weight vectors for the users and the data points to be annotated. This can be related to our case, where each

data point to be annotated in our daily consumption curves corresponds to a *WAVE exercise*. Then follows directly that the input vector v_i that concatenates all of the m annotation tasks, each being a binary vector of dimension d, as a vector with $m * d$ components, is defined as follows:

$$v_{i_k} = \begin{cases} 1 & \text{if the user and the gold standard agree} \\ & \text{on } k^{th} \text{ minute} \\ 0 & \text{otherwise} \end{cases}$$

The WAVE algorithm outputs the following:

- an $m * d$-dimensional weight vector $WAVE_{tasks}$ for all the minutes in all the given annotation tasks, which gives more importance to more difficult minutes;
- an n-dimensional weight vector $WAVE_{users}$ for the n users, which gives more weight to users who label more difficult minutes (in the daily curves to be annotated) correctly.

We will use the user weights $WAVE_{users}$ in the following sections.

2) Data features:

a) Behavioral features: Our platform allows to capture behavioral features relating to the annotation tasks. We are thus using features relating to the users' interactions while annotating each curve.

As can be seen Figure 1, the platform consists of a *tool box*, where the user can choose the pencil to highlight zones considered as *active*, the eraser to correct their annotations and a glass magnifier for zooming in or out. The workbench consists of an *annotation panel* as the yellow area where the curve to be annotated is displayed on the left side, and on the right panel, a *week-viewer* to display the next 7 days of data, in order to determine if the curve to annotate reflects an occasional or the usual consumption pattern.

We can collect the following information for each annotation $t_{i,j}$ for task i and user j:

- $\#seconds_{i,j}$ needed to complete the annotation $t_{i,j}$;
- $\#mousemovements_{i,j}$ over the tool box for annotating $t_{i,j}$;
- $\#mousemovements_{i,j}$ over the annotation panel for annotating $t_{i,j}$;
- $\#mousemovements_{i,j}$ over the week-viewer for annotating $t_{i,j}$;
- $\#milliseconds_{i,j}$ spent over the tool box area for annotating $t_{i,j}$;
- $\#milliseconds_{i,j}$ spent over the annotation area for annotating $t_{i,j}$;
- $\#milliseconds_{i,j}$ spent over the week-viewer area for annotating $t_{i,j}$.

b) Data characteristics: Additionally, we make use of the each task i's difficulty level c_{D_i}.

B. Analysis

In this part, we focus on the data analysis to determine how to characterize the regular users' and experts' work.

1) Online scoring: To guarantee the quality of the data collected in a crowdsourcing system, we need to detect weak and adversarial workers rapidly to either ignore their contribution or to ban them. If the number of contributors is limited, we need to salvage as many annotations as possible, by giving some slack to users for whom we can detect a temporary decrease in the quality of the annotations, if they have proven to be well-performing in the past. For this reason, we should value the user's expertise in regards to the task's difficulty level and consider that bad performance, if temporary, could be overlooked and explainable (weaker knowledge about a specific appliance's functioning, contrasting with solid performance with other types of appliances).

We define the combined score $c_{i,j}$ for user j solving the current task i as in Equation 7, where α, β and γ can be used for giving more or less weight to different other scores.

$$c_{i,j} = F1_{i,j}^{\alpha} * score_{H_{i,j}}^{\beta} * score_{A_{i,j}}^{\gamma} \tag{7}$$

We analyze the evolution of the annotations' quality as the worker is submitting them by computing the current combined score $c_{i,j}$ and acknowledging for the tasks' difficulty level with the coefficient c_{D_i} (0.2 for easy, 0.3 for medium, 0.5 difficult). To account for the past performances, we consider a remembering factor $\alpha_r \in [0, 1]$ for preferring more recent contributions, which applies exponential decay over past annotations. We define the online score $score_{O_{i,j}}$ for the current i^{th} task as the recurrence relation as in Equation 8 with $score_{O_{1,j}} = c_{1,j}$.

$$score_{O_{i,j}} = \frac{c_{D_i}}{\alpha_r + c_{D_i}} c_{i,j} + \frac{\alpha_r}{\alpha_r + c_{D_i}} score_{O_{i-1,j}} \tag{8}$$

2) Classification: To distinguish regular users from experts, we use feature vectors representing each annotation task and use known machine learning classifiers. We describe which features can be used and which algorithms would be suitable in the following.

a) Feature vectors: We want to classify each annotation provided by each user j as either coming from an expert or a regular user. For this, we can build a feature vector from each task i's specific scores as described in IV-A1. Namely, we use the confusion matrix values $TP_{i,j}$, $TN_{i,j}$, $FP_{i,j}$, $FN_{i,j}$, $precision_{i,j}$, $recall_{i,j}$, and the $score_{H_{i,j}}$ and $score_{A_{i,j}}$ scores. Additionally, we use all task specific features described in IV-A2, i.e. interaction features and the tasks' difficulty levels.

b) Classifiers: As our data classes are unbalanced, due to having less expert data than regular user data, we use Adaboost [18] as an ensemble classification method for combining weak classifiers. The weak classifiers that we consider are Naive Bayes, LibLinear, Multi-Layer Perceptrons and Random Trees.

3) Prediction: In this part, we investigate how to combine the wisdom of the crowd for annotating each task and for approaching the expert users' annotation level. This would mean deciding for each data point in a curve to be annotated, what

value it should take, depending on the contribution of multiple annotators. Classical methods of combining the labeling for each curve to annotate t_i from each $t_{i,j}$ contribution by each user j exist, but we take advantage of the user's expertise level to improve the prediction. We want to obtain the k^{th} data point t_{i_k} by combining each t_{i,j_k} provided by each user j.

a) Majority voting: Majority voting is the simplest approach to combine multiple annotations, by choosing the value supplied by the majority of the users among n users as shown in Equation 9.

$$t_{i_k} = \begin{cases} 1 & \text{if } \sum_{j=1}^{n} t_{i,j_k} \geq \frac{n}{2} \\ 0 & otherwise \end{cases} \tag{9}$$

This combination is not robust if the majority of the annotators are unknowledgeable or if an attacker creates multiple accounts and feeds the incorrect labeling multiple times.

b) Weighted Majority Voting: The other approach consists in weighting the majority voting [19], [20] according to the workers' expertise level. This would allow to increase the influence of more experienced users and diminish the influence of weaker users. We are looking to define the weights w_j for each user j, normalized over all n users, to compute the predicted label t_{i_k} as in Equation 10.

$$t_{i_k} = \begin{cases} 1 & \text{if } \sum_{j=1}^{n} w_j * t_{i,j_k} \geq 0.5 \\ 0 & otherwise \end{cases} \tag{10}$$

To reflect the user's expertise level, we can combine existing scores to reflect the users' accuracy in labeling against the gold standard, their performance compared to the other users, and account for the task's difficulty contribution coefficient c_{D_i} (0.2 for easy, 0.3 for medium and 0.5 for difficult tasks), as in Equation 11 for each user j over the m tasks to be solved. Then

$$w_j = \frac{WAVE_{users_j}{}^{\delta} \sum_{i=1}^{m} c_{D_i} F1_{i,j}^{\alpha} score_{H_{i,j}}^{\beta} TP_{i,j}^{\gamma}}{\sum_{j=1}^{n} WAVE_{users_j}{}^{\delta} \sum_{i=1}^{m} c_{D_i} F1_{i,j}^{\alpha} score_{H_{i,j}}^{\beta} TP_{i,j}^{\gamma}} \tag{11}$$

where the constants α, β, γ and δ can be adjusted to emphasize or reduce the impact of some scores.

We will evaluate the prediction from the crowd against the gold standard and observe the robustness of the weights by changing the set of tasks used to build the expertise weights. For this, we proceed with leave-one-out cross-validation and use $m-1$ tasks for training and obtaining the expertise weights w_j to predict the annotations for the left-out task.

V. Results

In the following, we present and discuss the results for the three different axes of our analysis: the online scoring of the users' performances over time, the classification of the annotations as expert or non-expert work and the prediction of an annotation based on the contributors' expertise levels and the performance of others.

A. Online Scoring

Many factors can influence the quality of the annotation for non-adversarial users, such as their focus, their motivation or the task's difficulty level. On the long run, we would like to retain users who perform usually well, accounting for occasional bad results, but we would like to be able to react quickly to weak or malicious workers and expel them from the system.

We introduce online scoring to take a user's performance over time into account, and to determine whether or not to keep them in the system in on the fly. We parameterize the online scores described in Equation 8 by evaluating the remembrance factor α_r for different values: 0 (forgetting the past), 0.5 and 1 as in Figures 2 and 3 for expert and users respectively. As can be seen in Figure 2a and Figure 3a, selecting $\alpha_r > 0$ allows to account for the past performances, but an overall score that does not take the task difficulty into account would promote users who have solved easier tasks successfully, but does not guarantee that they will succeed at solving medium or difficult tasks. Choosing a larger $\alpha_r > 0$ as in Figures 2b and 3b, shows a smoother online score that is more resilient to punctual bad labeling. This would allow to be more lenient to users, based on past good performance. We also observe that we can select a threshold for accepting or even promoting users or banning them instead. Additionally, we can clearly distinguish experts' performance from regular users' with some comfortable margin and enables the selection of a threshold that can be tailored to the sensitivity to bad performance.

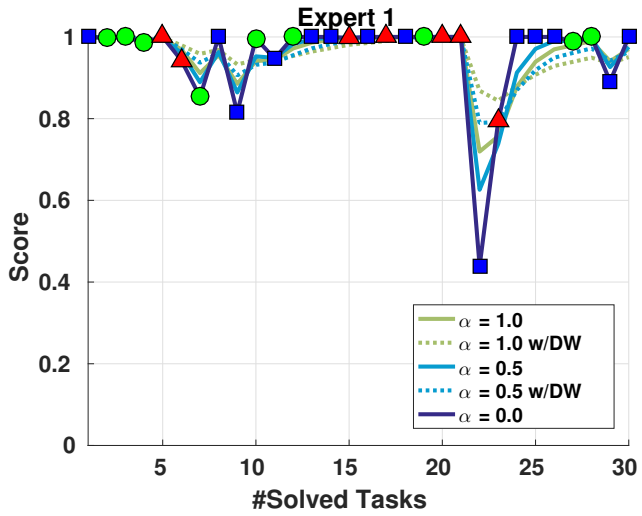
B. Classification

As described previously in IV-B2, we use Adaboost to address the class imbalance due to having more regular users than experts. We classify vectors for each annotation as described in IV-B2. We perform the classification through a 10-fold cross-validation for the datasets consisting of

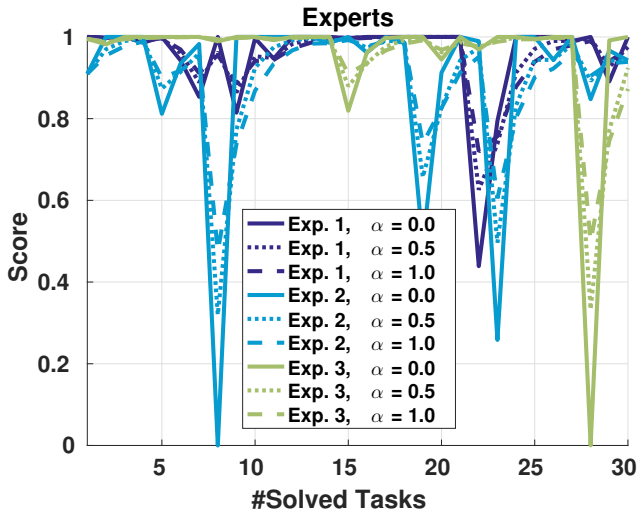
1) the experts' and the users' annotations before receiving the coaching;
2) the experts' and the users' annotations after receiving the coaching;
3) the experts' and the users' annotations before and after receiving the coaching;

and obtain the results shown in Table I.

We notice that the naivest classifier, i.e., the Naive Bayes, performs overall worst than the Random Tree, LibLinear and Multi-Layer Perceptrons. This is due to the class imbalance and that Naive Bayes' weights are smaller for the class with the least representatives [21]. Overall, the best classification scores are achieved before coaching the regular users, as their performance is improved significantly afterwards (we can observe an improvement of 10-20%), and their work becomes comparable to the experts' annotations. The combined dataset (containing annotations before and after the coaching) shows the worst scores overall, and this is due to the incertitude induced by the improved annotations, bringing the non-experts' closer to the experts' contributions.

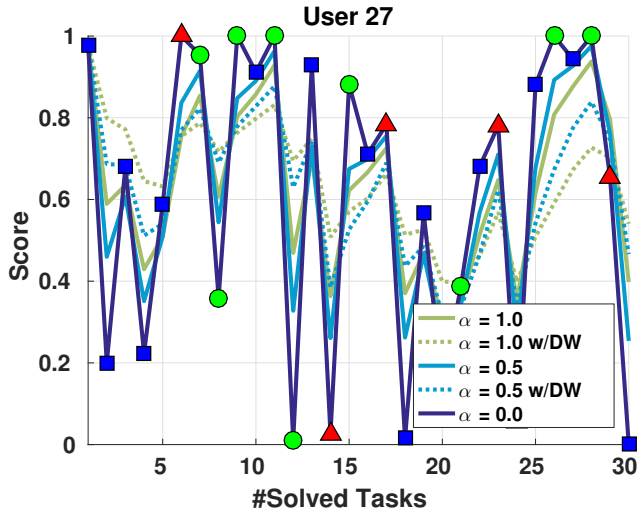


(a) Online scoring for an expert as tasks get completed with $\alpha_r \in 0, 0.5, 1$, per task difficulty and overall (with and without difficulty level weighting). *Easy* tasks as ●, *medium* tasks as ■, *difficult* tasks as ▲.

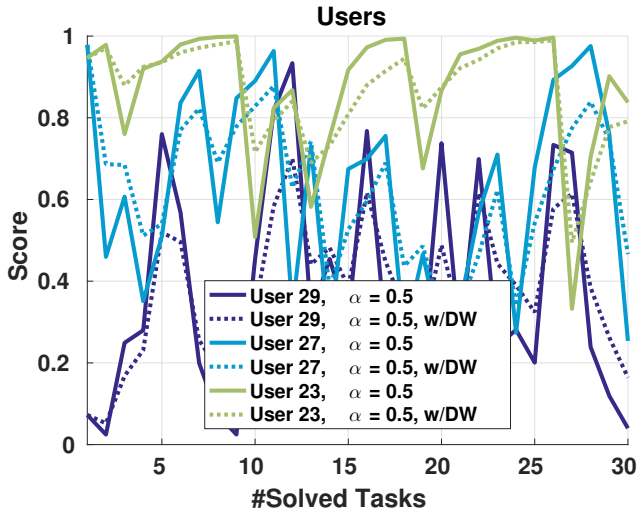


(b) Online scoring for all experts overall and accounting for the tasks' difficulty get completed with different $\alpha_r \in \{0.0, 0.5, 1.0\}$

Fig. 2: Comparison for online scoring between experts



(a) Online scoring for a regular user as tasks get completed with $\alpha_r \in 0, 0.5, 1$, per task difficulty and overall (with and without difficulty level weighting). *Easy* tasks as ●, *medium* tasks as ■, *difficult* tasks as ▲.



(b) Online scoring for a weak, an average and a strong user overall, with difficulty level weighting, for $\alpha_r \in \{0.0, 0.5, 1.0\}$

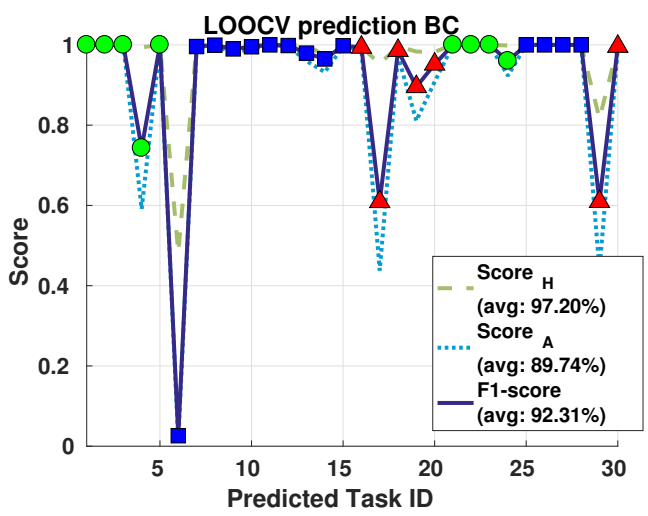
Fig. 3: Comparison for online scoring between regular users

The best scores are achieved for the datasets composed of the annotations obtained before the coaching session and for the SVM implementation in LibLinear and the Random Tree, with F1-scores for classifying the experts of 61% and 86.5% for the regular users, and 61% and 84.7% respectively. The relatively low F1-scores are due to a larger share of *false negatives*, that misclassified the experts due to the similarity in the annotation of the easy tasks and the medium tasks. The share of *false positives* is however an indication of the potential for selecting users, whose contribution should be promoted (if looking at the performance on the medium task solving), but we could explain this by the fact that easy tasks were solved as well as the experts.

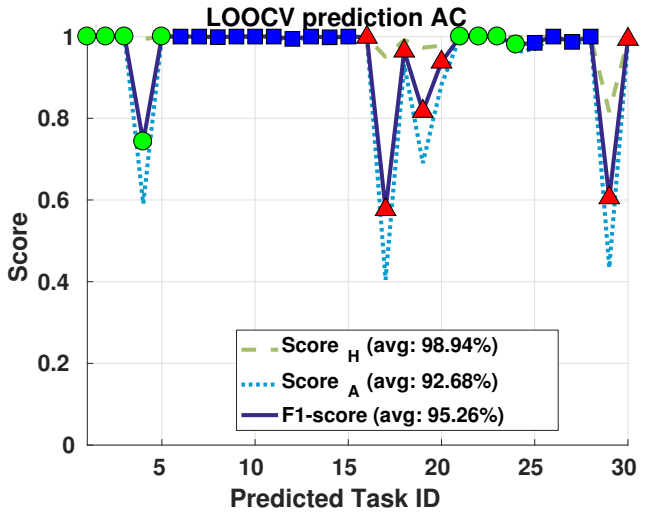
C. Prediction

As described previously in IV-B3, we are investigating the prediction of an annotation by combining the work of several workers, by computing a weighted majority voting based on the expertise weight as in Equation 10. The scoring is obtained by computing the $F1$-score, the activity score $score_A$ and the Hamming score $score_H$ between the resulting crowd-combined annotation vector and the respective gold standard.

We first examine the leave-one-out cross-validation for predicting one annotation based on computing the expertise weights on $m-1$ other tasks as can be seen in Figure 4, where we show the prediction for each task based on using the remaining ones for training. As can be seen in Figure 4a,



(a) Prediction on the set of curves obtained before the coaching with difficulty weighting. Average prediction scores: F1-score 92.3% (dark blue, solid line), $score_A$ 89.73% (light blue, dotted line), $score_H$ 97.2% (green, dashed line). *Easy* tasks as ●, *medium* tasks as ■, *difficult* tasks as ▲.



(b) Prediction on the set of curves obtained after the coaching with difficulty weighting. Average prediction scores: F1-score 95.2% (dark blue, solid line), $score_A$ 92.7% (light blue, dotted line), $score_H$ 98.9% (green, dashed line). *Easy* tasks as ●, *medium* tasks as ■, *difficult* tasks as ▲.

Fig. 4: Prediction using leave-one-out cross-validation for training and obtaining the expertise scores before and after the coaching.

TABLE I: Classification results (precision, recall and F1-score) per class (Expert / User) using Adaboost and weak classifiers (LibLinear, Naive Bayes, Random Tree, Multi-Layer Perceptrons), before coaching (BC), after coaching (AC), with and without coaching (All)

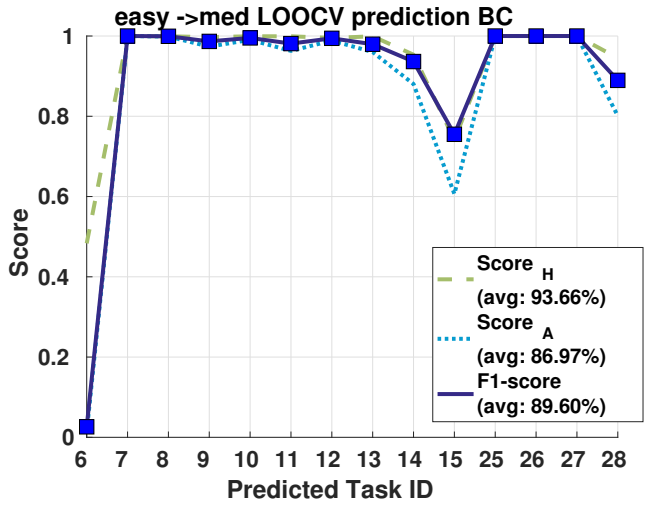
Class	Data	$Prec_E$	Rec_E	$F1_E$	$Prec_U$	Rec_U	$F1_U$
LL	BC	0.734	0.522	0.61	0.818	0.919	0.865
LL	AC	0.567	0.378	0.453	0.733	0.856	0.79
LL	All	0.549	0.311	0.397	0.855	0.941	0.896
NB	BC	0.492	0.644	0.558	0.832	0.714	0.765
NB	AC	0.487	0.422	0.452	0.729	0.778	0.753
NB	All	0.356	0.233	0.282	0.836	0.903	0.868
RT	BC	0.65	0.578	0.612	0.827	0.867	0.847
RT	AC	0.568	0.467	0.512	0.755	0.822	0.787
RT	All	0.553	0.289	0.38	0.852	0.946	0.827
MLP	BC	0.717	0.422	0.531	0.789	0.929	0.853
MLP	AC	0.594	0.422	0.494	0.748	0.856	0.798
MLP	All	0.481	0.278	0.352	0.848	0.931	0.888

by accounting for the difficulty of the task, as more weight is given to users who have solved medium ($c_{D_i} = 0.5$) and difficult tasks better ($c_{D_i} = 0.3$), we obtain high average prediction scores (92.3% for the F1-score, 89.7% for the $score_A$, 97.2% for the $score_H$), for the curves obtained before the coaching. In Figure 4b, with the same weight distribution, the results have improved for annotations made after the coaching, as the average prediction scores (95.3% for the F1-score, 92.7% for the $score_A$, 98.9% for the $score_H$), namely in the annotation of the freezer, which was previously poorly annotated as can be seen in the second dip in Figure 4a.

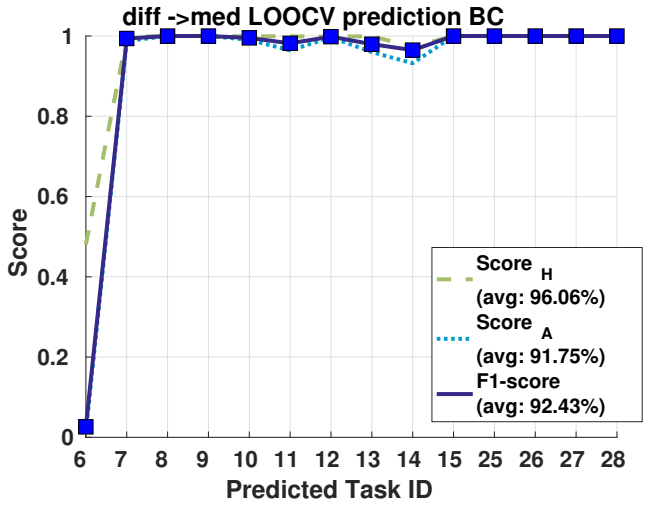
We examine how selecting a training set built on annotated curves of the same difficulty level influences the scores of the combined annotations. For this purpose we provide comparisons for the dataset of the annotations obtained before coaching:

- training on easy curves, then predicting the outcome for medium curves;
- training on easy curves, then predicting the outcome for difficulty curves;
- training on medium curves, then predicting the outcome for difficulty curves;
- training on difficult curves, then predicting the outcome for medium curves.

As can be seen in Figures 5 and 6, being able to solve the easy tasks does not correlate with being able to solve the medium and difficult tasks as can be seen in Figures 5a and 6a. Similarly, while training on medium does slightly improve the solving of the difficult tasks as seen in Figure 6b, drastic improvements are only observed when users that can solve more difficult tasks pull the solving of medium tasks up as can be seen in Figure 5b. We see that the training with the difficult curves shows the highest average prediction scores (92.4% for the F1-score, 91.8% for the $score_A$, 96.1% the $score_H$) and it shows that users that are able to solve those tasks successfully have more advanced knowledge about the energy domain. They can generalize across the annotation tasks, regardless of their difficulty. Also, the training size is the smallest with only 7 benchmark curves. The runner-up configuration, training with medium curves, with average prediction scores of 89.6% for the F1-score, 87.0% for the $score_A$, 93.7% for the $score_H$, failed to generalize for all tasks of annotating the oven and the heater (due to confusion about the baseline consumption), despite having 14 benchmark curves as references for the training. This shows us that choosing discriminating benchmarks can effectively improve

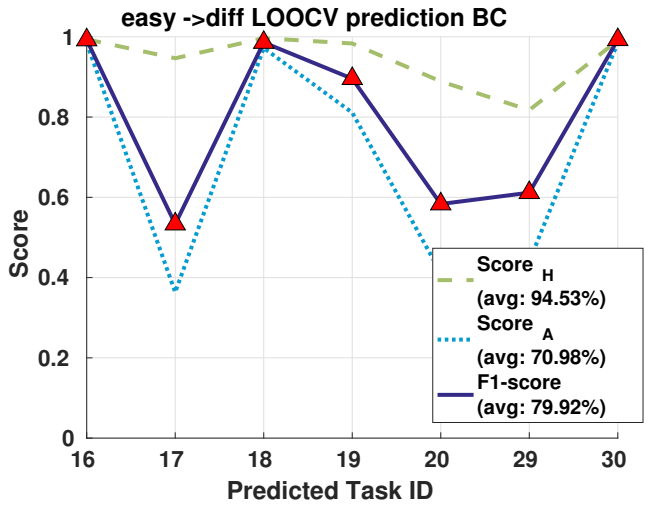


(a) Training on easy, predicting on medium. Average prediction scores: F1-score 89.6% (dark blue, solid line), $score_A$ 87.0% (light blue, dotted line), $score_H$ 93.7% (green, dashed line), difficulty weighting, before coaching. *Easy* tasks as ●, *medium* tasks as ■, *difficult* tasks as ▲.

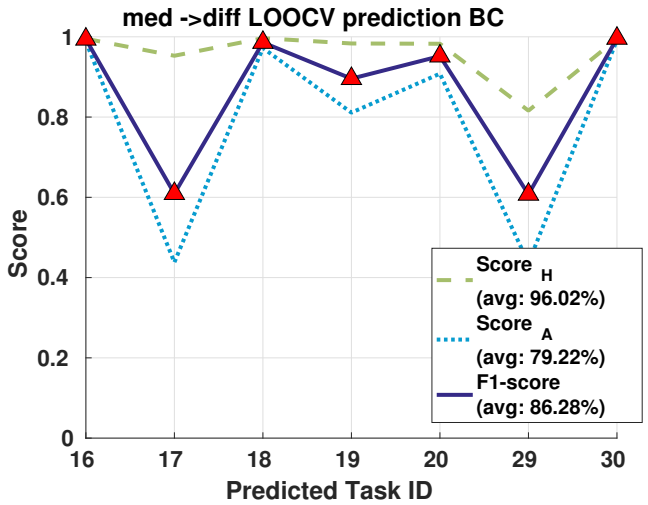


(b) Training on difficult, predicting on medium. Average prediction scores: F1-score 92.4% (dark blue, solid line), $score_A$ 91.8% (light blue, dotted line), $score_H$ 96.1% (green, dashed line), difficulty weighting, before coaching. *Easy* tasks as ●, *medium* tasks as ■, *difficult* tasks as ▲.

Fig. 5: Prediction of medium tasks using leave-one-out cross-validation for training and obtaining the expertise scores.



(a) Training on easy, predicting on difficult. Average prediction scores: F1-score 79.9% (dark blue, solid line), $score_A$ 71.0% (light blue, dotted line), $score_H$ 94.5% (green, dashed line), difficulty weighting, before coaching. *Easy* tasks as ●, *medium* tasks as ■, *difficult* tasks as ▲.



(b) Training on medium, predicting on difficult. Average prediction scores: F1-score 86.3% (dark blue, solid line), $score_A$ 79.2% (light blue, dotted line), $score_H$ 96.0% (green, dashed line), difficulty weighting, before coaching. *Easy* tasks as ●, *medium* tasks as ■, *difficult* tasks as ▲.

Fig. 6: Prediction of difficult tasks using leave-one-out cross-validation for training and obtaining the expertise scores.

the prediction accuracy and if done carefully, requires few curves.

VI. Conclusion

We have analyzed the quality of regular users' against experts' work in the domain of energy time series datasets labeling by collecting data through a user study. The users had varying degrees of familiarity with the energy jargon or the functioning of electrical appliances. We have quantified the discrepancy based on the difficulty of the tasks and have shown that improvements can be achieved if the users are trained to pay attention to certain details when annotating different appliances or circuit-level data. We showed that the classification does not provide enough discrimination between regular users and experts, but it can be combined with the online scoring of the annotations to provide an effective way for detecting when to promote a user or to discard weaker users. Moreover, if we leverage the difficulty of the annotations and carefully curate the difficult tasks, we can use a small number of seed benchmark tasks to improve the prediction quality significantly, which would reduce the work load on the expert users.

We can further the analysis by looking at the evolution of the quality of the annotations as time progresses and measure

how much time an annotation should take before straining the annotator too much. As we have observed an improvement in quality by coaching the users, we could also schedule the benchmark tasks as to take into account their difficulty and evaluate the order of apparition of the tasks to be solved and their impact on the quality of the annotations. Additionally, the CAFED platform contains a user engagement component and dispenses badges based on achievements. One user reportedly provided over 175 annotations (for a total of 3 hours in a row) due to the motivation of acquiring more badges as had been previously shown for text labeling gamification [12]. We still need to examine how the quality of the data is stirred as the badges are allocated. Additional data from the survey and behavioral features could be leveraged for improving the expertise levels of the users. We have underlined the necessity for the designers of a collaborative system for labeling data where domain knowledge is required, to make use of more domain-specific information to craft the challenge benchmark questions to vet the quality of the workers.

REFERENCES

[1] H.-Â. Cao, T. K. Wijaya, K. Aberer, and N. Nunes, "A Collaborative Framework for Annotating Energy Datasets," in *Proc. BigData '15*. Santa Clara, CA, USA: IEEE, Oct 2015, pp. 2716–2725.

[2] A. P. Dawid and A. M. Skene, "Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm," *Applied statistics*, vol. 28, no. 1, pp. 20–28, 1979.

[3] P. G. Ipeirotis, F. Provost, and J. Wang, "Quality Management on Amazon Mechanical Turk," in *Proc. HCOMP '10*. Washington, DC, USA: ACM, Jul 2010, pp. 64–67.

[4] P. Smyth, U. Fayyad, and M. Burl, "Inferring Ground Truth from Subjective Labelling of Venus Images," in *Proc. NIPS '94*. Denver, CO, USA: MIT, Dec 1995, pp. 1085–1092.

[5] R. Snow, B. O. Connor, D. Jurafsky, A. Y. Ng, D. Labs, and C. St, "Cheap and Fast—But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks," in *Proc. EMNLP '08*. Stroudsburg, PA, USA: ACL, Dec 2008, pp. 254–263.

[6] P. Welinder, S. Branson, S. Belongie, and P. Perona, "The Multidimensional Wisdom of Crowds," in *Proc. NIPS '10*. Vancouver, BC, Canada: Curran Associates, Dec 2010, pp. 1–9.

[7] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan, "Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise," in *Proc. NIPS '09*. Vancouver, BC, Canada: Curran Associates, Dec 2009, pp. 2035–2043.

[8] A. Kittur, J. V. Nickerson, M. Bernstein, E. Gerber, A. Shaw, J. Zimmerman, M. Lease, and J. Horton, "The Future of Crowd Work," in *Proc. CSCW '13*. San Antonio, TX, USA: ACM, Feb 2012, pp. 1301–1317.

[9] G. Kazai and I. Zitouni, "Quality Management in Crowdsourcing Using Gold Judges Behavior," in *Proc. WSDM '16*. San Francisco, CA, USA: ACM, Feb 2016, pp. 267–276.

[10] J. Le, A. Edmonds, V. Hester, and L. Biewald, "Ensuring Quality in Crowdsourced Search Relevance Evaluation: The Effects of Training Question Distribution," in *Proc. SIGIR '10*. Geneva, Switzerland: ACM, Jul 2010, pp. 17–20.

[11] H. Li, B. Zhao, and A. Fuxman, "The Wisdom of Minority: Discovering and Targeting the Right Group of Workers for Crowdsourcing," in *Proc. WWW '14*. Seoul, South Korea: ACM, Apr 2014, pp. 165–175.

[12] C. Eickhoff, C. G. Harris, A. P. de Vries, and P. Srinivasan, "Quality through Flow and Immersion: Gamifying Crowdsourced Relevance Assessments," in *Proc. SIGIR '12*. Portland, OR, USA: ACM, 2012, pp. 871–880.

[13] J. Feminella, D. Pisharoty, and K. Whitehouse, "Piloteur : A Lightweight Platform for Pilot Studies of Smart Homes," in *Proc. BuildSys '14*. Memphis, TN, USA: ACM, Nov 2014, pp. 110–119.

[14] H.-Â. Cao, T. K. Wijaya, and K. Aberer, "Estimating Human Interactions with Electrical Appliances for Activity-based Energy Savings Recommendations," in *Proc. BuildSys '14*. Memphis, TN, USA: ACM, Nov 2014, pp. 206–207.

[15] ——, "Estimating Human Interactions With Electrical Appliances for Activity-based Energy Savings Recommendations," in *Proc. BigData '16*. Washington, DC, USA: IEEE, Dec 2016.

[16] H. Kim, H. Kim, H. Moon, and H. Ahn, "A Weight-adjusted Voting Algorithm for Ensemble of Classifiers," *Journal of the Korean Statistical Society*, vol. 40, no. 4, pp. 437–449, Dec 2011.

[17] A. Kim, M. Kim, and H. Kim, "Double-bagging Ensemble Using WAVE," *Communications for Statistical Applications and Methods*, vol. 21, no. 5, pp. 411–422, Sep 2014.

[18] R. E. Schapire, *The Boosting Approach to Machine Learning: An Overview*. Springer, 2003, pp. 149–171.

[19] N. Littlestone and M. Warmuth, "The Weighted Majority Algorithm," *Information and Computation*, vol. 108, no. 2, pp. 212–261, Feb 1994.

[20] T. TIAN and J. Zhu, "Max-Margin Majority Voting for Learning from Crowds," in *Proc. NIPS '15*. Montreal, QC, Canada: Curran Associates, Dec 2015, pp. 1621–1629.

[21] J. D. Rennie, L. Shih, J. Teevan, and D. Karger, "Tackling the Poor Assumptions of Naive Bayes Text Classifiers," in *Proc. ICML '03*. Washington, DC, USA: AAAI, Aug 2003, pp. 616–623.