Knowledge Tracing to Model Learning in Online Citizen Science Projects

Kevin Crowston, Carsten Østerlund, Tae Kyoung Lee, Corey Jackson, Mahboobeh Harandi, Sarah Allen, Sara Bahaadini, Scott Coughlin, Aggelos K. Katsaggelos, *Fellow, IEEE*, Shane L. Larson, Neda Rohani, Joshua R. Smith, Laura Trouille and Michael Zevin

***Abstract*—We present the design of a citizen science system that uses machine learning to guide the presentation of image classification tasks to newcomers to help them more quickly learn how to do the task while still contributing to the work of the project. A Bayesian model for tracking volunteer learning for training with tasks with uncertain outcomes is presented and fit to data from 12,986 volunteer contributors. The model can be used both to estimate the ability of volunteers and to decide the classification of an image. A simulation of the model applied to volunteer promotion and image retirement suggests that the model requires fewer classifications than the current system.**

***Index Terms*— Citizen science, machine learning, training**

I. Introduction

To be successful, online production communities need to sustain a critical mass of skilled and active participants [6], [11], which requires attracting newcomers and helping them learn to be effective participants in the community. In traditional organizations, new members often go through formal training to learn how to contribute. However, the particular characteristics of online communities present at least two challenges to newcomer training. First, many online groups rely on volunteers who contribute in their free time, reducing their willingness to participate in formal training prior to engaging. A second complication is the skewed distribution of contributions seen in most projects: many volunteers contribute only a few times and only a few become sustained contributors [5]. This skew means that requiring training that increases the barrier to entry and delays newcomers' contributions might result in many participants not contributing at all.

To make online communities more effective calls for systems that enable motivated participants to make productive contributions to the community while also supporting an efficient and engaging learning process for newcomers. In this paper, we present the design of a citizen-science project that incorporates machine learning to guide training for new volunteers using real tasks with uncertain outcomes. The specific contribution of this paper is to develop and empirically examine a Bayesian model for tracking volunteer performance to guide training using such tasks.

A. Setting: Gravity Spy

Our study is set in the context of the Gravity Spy [19] citizen science project (http://gravityspy.org/). Citizen science is a broad term describing scientific projects that rely on contributions to scientific research from members of the general public (i.e., citizens in the broadest sense of the term). There are several kinds of citizen-science projects: some have volunteers collect data, while others, including the one we examine in this paper, have volunteers analyze existing data. The interactions between volunteers and the project organizers typically take place via the Web, e.g. on a site that accepts contributed data or that presents data to be analyzed and collects volunteers' annotations (e.g., Zooniverse.org), thus making them examples of online communities.

The Gravity Spy system was developed to support the Laser Interferometer Gravitational-wave Observatory (LIGO). LIGO comprises two detectors that measure minute changes in distance caused by the gravitational waves distorting space as they travel through it. However, the sensitivity that enables LIGO to detect distant astrophysical events also makes it very susceptible to non-astrophysical instrumental and environmental noise, referred to as "glitches". Glitches hamper the detection of gravitational wave events, either by blocking events outright or by increasing the number of potential events to be examined. At LIGO's current sensitivity, detectable

Manuscript submitted 18 July 2018, revised 13 February 2019, 13 May 2019 and 11 July 2019. This work was supported in part by the US National Science Foundation under grant INSPIRE 15-47880. *(Corresponding author: Kevin Crowston.)*

K. Crowston, C. Østerlund and M. Harandi are with the School of Information Studies, Syracuse University, Syracuse, NY 13244 USA (emails: crowston@syr.edu, costerlu@syr.edu and mharandi@syr.edu).

C. Jackson was with Syracuse University School of Information Studies, Syracuse, NY 13244 USA. He is now with University of California, Berkeley, Berkeley CA 94720 USA (email: coreybjackson@berkeley.edu).

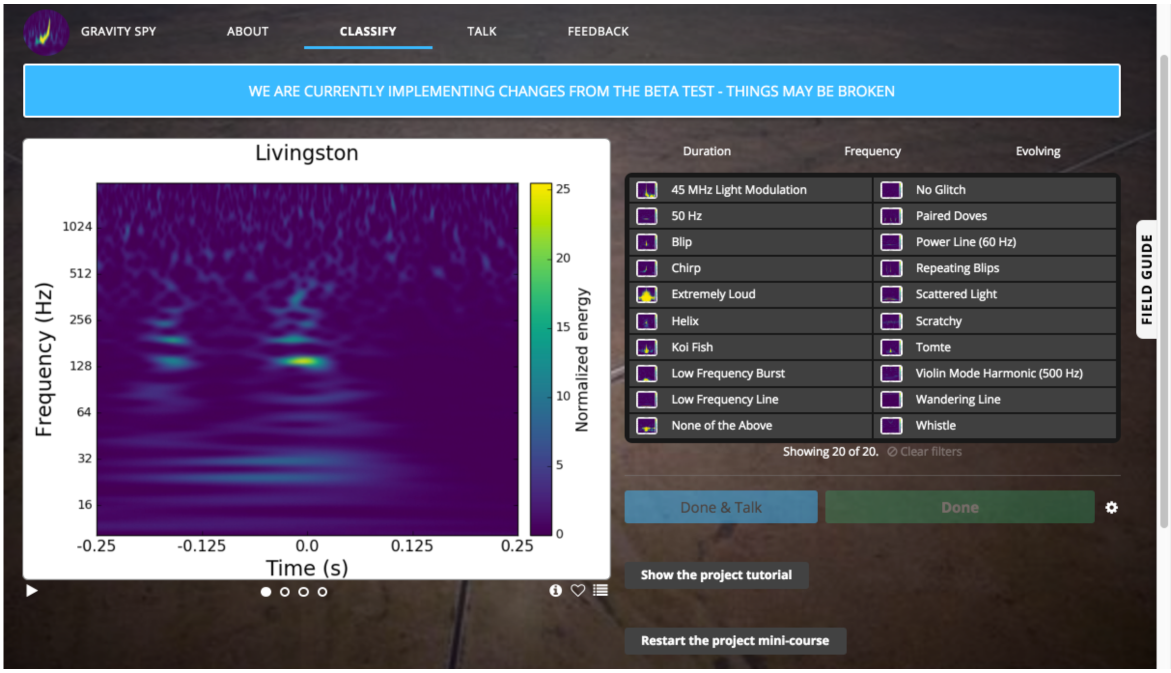
T. K. Lee is with the Department of Communication, University of Utah, Salt Lake City, UT 84112 USA (email: tae.lee@utah.edu).

S. Allen and L. Trouille are with Adler Planetarium, Chicago, IL 60605 USA (emails: sarah@zooniverse.org and trouille@zooniverse.org).

S. Bahaadini, A. K. Katsaggelos and N. Rohani are with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 606201 USA (emails: sara.bahaadini@u.northwestern.edu, aggk@eecs.northwestern.edu and nedarohani2019@u.northwestern.edu).

S. Coughlin, S. L. Larson and M. Zevin are with the Center for Interdisciplinary Exploration and Research in Astrophysics (CIERA) and Dept. of Physics and Astronomy, Northwestern University, 2145 Sheridan Rd, Evanston, IL 60208 USA (emails: scottcoughlin2014@u.northwestern.edu, s.larson@northwestern.edu and michaelzevin2014@u.northwestern.edu).

J. R. Smith is with the Department of Physics, California State University, Fullerton, CA 92831 USA (email: josmith@fullerton.edu).


WE ARE CURRENTLY IMPLEMENTING CHANGES FROM THE BETA TEST - THINGS MAY BE BROKEN

Showing 20 of 20.

Figure 1. Full Gravity Spy classification interface (http://gravityspy.org/).

astrophysical events are expected to occur only about once a week, while a glitch may occur every few seconds, making a search for true events akin to finding a needle in a haystack.

Similar glitches may have a common cause that can be eliminated if it can be identified, so finding and classifying glitches stand out as core tasks for improving the LIGO detectors. However, with thousands of glitches, the LIGO researchers do not have the manpower to examine them all. Reliance on computers alone has also so far fallen short, as the diversity of glitches defies easy attempts at classification. At present, there are 21 known types of glitches, but many glitches do not fit one of these categories and so may be examples of as-yet-unidentified classes of glitch. Presently, humans are much better at the visual processing needed to identify similar types of glitches. Given these concerns, the project has developed a citizen-science approach to classifying glitches.

When using a citizen-science platform such as Zooniverse, volunteers are presented with images and asked to classify them into one of the known categories. Gravity Spy also provides options of "none of the above" or "no image" for images that in fact do not include an event of interest. The current interface for the Gravity Spy system is shown in Fig. 1: an image of a glitch to be classified is shown on the left as a spectrograph, with time on the x-axis, frequency on the y and intensity represented as colour from blue to yellow, and possible classes on the right. The task for the volunteers to learn is how to identify the correct class of the glitch from the spectrograph.

II. Machine-Learning-Supported Training

To address the training problem faced by citizen-science projects and online production communities more generally, the Gravity Spy system creates a symbiotic relationship between citizen-science volunteers and computer algorithms, each helping the other learn to classify images. Volunteers sort through vast amounts of data to create a dataset that can be used to train machine-learning (ML) algorithms. And conversely, as the ML algorithms learn from this classified dataset, they select images to present that assist humans to learn. The first process is common; the second is the main innovation of the Gravity Spy system.

A. Machine Learning

We start by briefly describing the ML applied to the glitches. In addition to a store of images to be classified, the system includes gold-standard data sets, glitches that have been labelled by human experts. ML models are trained using the gold-standard data (one model for each class of glitch). A description of the ML approach is given in [19]. The trained ML models are then applied to all unlabelled glitches, annotating each with the ML model's level of confidence that the glitch is a member of each class. Often, the confidence level for one of the classes will be much higher than for the others, suggesting that that glitch is a member of that class. But it also possible for none of the confidence levels to be high, meaning that the ML models are not able to classify the glitch or for more than one confidence to be at an intermediate level, meaning that the ML models are uncertain about the classification.

B. Training Citizen Science Volunteers

As with other citizen science project, the Gravity Spy website provides volunteers with a variety of training materials, such as a short tutorial on the site operation and a field guide describing the different glitch classes. The main advance in the Gravity Spy system is that it uses ML results to train new human volunteers. The system moves new volunteers through a sequence of levels in which they are presented with different

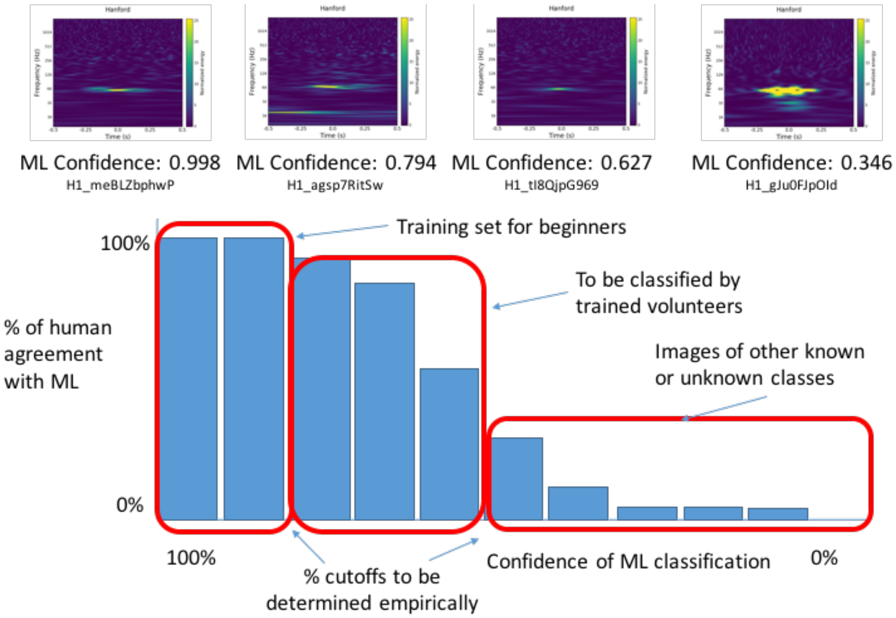


Figure 2. Expected relationship between ML confidence (x-axis) in a glitch belonging to a class and proportion of images assessed by human experts as belong to that class, with examples of glitches in each grouping.

classification tasks intended to improve their ability to classify images [15]. Essentially, the system acts like a tutoring system in picking tasks to help a beginner to learn, but selecting from the natural tasks of the citizen-science project rather than from a predefined set of training instances.

Specifically, a new volunteer is presented with glitches to classify that have been determined by the ML models as being likely to be of one of only two distinctive classes (in the current system, blips and whistles). Volunteers are asked to classify the glitch as being of one of the two classes or "none of the above" (i.e., with a reduced version of the interface shown in Fig. 1). Having only two distinctive classes of glitch to handle makes it easier for the volunteer to learn to distinguish the glitches. Once the volunteer is classifying glitches of the initial classes successfully (as described in [19] and below), the volunteer is advanced to the next training level, in which they see glitches believed by the ML to be of additional classes.

In the initial version of the system, there were four training levels, presenting 2, 5, 9 and 21 glitch classes respectively (i.e., 2, 3, 4 and 12 new glitch classes), as well as "none of the above". Earlier levels include classes that are more common (more volunteers classify at the earlier levels, so more data are needed) and more distinctive (to facilitate learning the distinction between classes). The number of glitches introduced at each level was chosen to gradually increase the number of glitches while keeping the training levels short enough to retain volunteer interest. More recently, the final level was split into two level, each introducing half as many new classes, for reasons that will be explained later in the paper.

In designing the system, we expected the relation between the ML-determined degree of confidence and likelihood of the glitch being of the given class to be as shown in Fig. 2. We expected that nearly all glitches above a certain threshold of ML confidence would be judged by the human experts to be of that class; nearly all below a certain threshold as not of that class; and in the intermediate range of confidence, a mix of in and not in the class. Further, when the ML has a high level of confidence in the classification of the glitches, we expected that these glitches would be exemplary images that would help the volunteer to learn how to identify that class of glitch. Accordingly, the design of the system was to use glitches at the top of the ML range for training, while the others were left to be examined by experienced volunteers.

Once volunteers have completed all rounds of training introducing the classes of glitches, they are considered fully qualified and given images to classify at varying levels of ML certainty in all known classes or even glitches for which the ML has no good classification.

In addition to being helpful to support learning, progress through levels of training also motivates volunteers by appealing to their sense of accomplishment [13], [17]. This motivation is further emphasized in the interface, e.g., by showing the unachieved levels greyed out and through messaging when mastery at the current level is achieved.

III. Modelling Volunteers' Learning

To properly target training requires an estimate of a volunteer's current level of knowledge. However, few current citizen-science projects evaluate volunteers' knowledge level. Those that do generally rely on proxies, such as the number of classifications contributed. To determine when volunteers have mastered the classification tasks and are ready to move to the next level, the Gravity Spy system maintains a model of each volunteer's ability that is updated with each classification.

We are experimenting with different approaches to modelling user ability. In this paper, we propose using Corbett and Anderson's [4] BKT model as a basis for the volunteer model. Bayesian methods are widely used to improve the

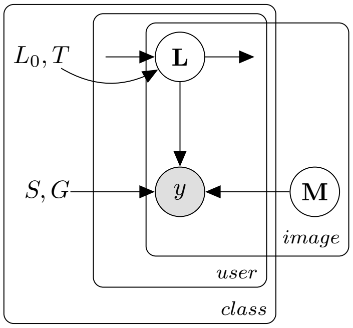


Figure 3. Plate diagram for the Knowledge Tracing model, adding a factor M for confidence in ML classification of the image.

performance of ML systems and human learning [9], [16]. The BKT Model in particular has been applied to model student learning in tutoring system as students practice different skills. The contribution of this paper is to propose and test modifications to this model to fit the ML-driven approach, by accounting for the possibility that the ML classification might be incorrect, rather than the volunteer's classification. Classifications of gold standard data can also be used to update the volunteer model without the uncertainty of the ML classification.

A plate diagram for the proposed model is shown in Fig. 3. The diagram shows that a volunteer's answer *y* for the classification of an image depends on a set of parameters for the volunteer, for the skill of being able to recognize a particular class of image and for the particular image. For each volunteer and each class of glitch, the model maintains an estimate of $p(L_n)$, the probability that the volunteer has learned how to classify after having classified *n* images of this class. $p(L_0)$, the estimate of a volunteer's initial ability, is a parameter.

The estimate is updated in two ways. First, it is updated from the prior estimate of learning in a Markov process that models a volunteer transitioning from not knowing to knowing how to classify. From [4], the formula to update the model's estimate of the volunteer's ability is:

$$p(L_{n+1}) = p(L_n|\text{answer}) + \big(1 - p(L_n|\text{answer})\big)\, p(T) \qquad (1)$$

where $p(L_n)$ is the probability that the volunteer knows how to classify after *n* classifications, answer is the volunteer's observed classification, either agreeing or disagreeing with the ML classification of the image and $p(T)$ is the probability of learning to classify if the volunteer does not already know how. Note that the BKT model does not include forgetting. As a result, it is appropriate for modelling short-term skill acquisition rather than long-term learning [7].

Second, the model updates the estimates of volunteers' ability based on their performance. $p(L_n|\text{action})$, the updated probability that volunteers know how to classify given their answer for the current image (either agreeing or disagreeing with the ML classification), is estimated using Bayesian inference [1]:

$$p(L_n|\text{agree}) = \frac{p(\text{agree}|L_n)\, p(L_n)}{p(\text{agree})} \qquad (2)$$

The components of (2) are defined in (3)–(5). As in the original BKT model in [4], there are two parameters that affect a volunteer's answer when classifying images of a particular class: $p(G)$, the probability of a volunteer getting the answer right without knowing how to classify (guessing) and $p(S)$, the probability of getting the answer wrong even while knowing how to classify (slipping).

$$p(\text{agree}|L_n) = p(M_n)\big(1 - p(S)\big) + p(C)\big(1 - p(M_n)\big)p(S) \qquad (3)$$

$$p(\text{agree}) = p(M_n)\, p(\text{correct}) + p(C)\big(1 - p(M_n)\big)\big(1 - p(\text{correct})\big) \qquad (4)$$

$$p(\text{correct}) = p(L_n)\big(1 - p(S)\big) + \big(1 - p(L_n)\big)p(G) \qquad (5)$$

In these equations, the new parameter $p(M_n)$ is the estimated probability that the particular image seen on this step is of the class identified by the ML classification algorithms. This factor is novel in our model and reflects the fact that rather than a set of exercises for which the system knows the correct answer, we instead have a set of images for which the system believes it knows the correct classification, but could be mistaken. Note that when $p(M_n)$ is 1, $p(\text{agree}) = p(\text{correct})$, the probability that the volunteer's classification is correct, and the model reduces to the standard BKT model. We also need an additional parameter, $p(C)$, the probability that the ML and the volunteer agree by chance when both are wrong.

We now explain (3)–(5). First, the chance of the volunteer agreeing with the ML classification of an image while knowing how to classify (3) is the chance that the ML is correct and the volunteer has not slipped or that the ML is not correct, the volunteer has slipped and by chance both have settled on the same incorrect choice. A simple model of chance agreement is to assume that when incorrect, the ML and volunteer choose other classes independently and with equal probability, which would make $p(C) = 1/(\#\, classes - 1)$. However, classes appear with varying frequency and some classes of glitch are more often confused with each other, so the probability of chance agreement is likely to be different, which is why we have made it a parameter of the model.

The unconditional probability of the volunteer agreeing with the ML classification (4) is the probability that both the ML and the volunteer are correct or that they are both incorrect but choose the same wrong class and so mistakenly agree.

The probability that the volunteer correctly classifies the image (5) is the probability that the volunteer knows how to classify and did not slip or that the volunteer does not know and guessed. Note that a volunteer's answer being correct or incorrect is defined relative to the image's (unknown) true classification and so is often not directly observable in practice.

Finally, the formula to update the estimate in the case of the volunteer disagreeing with the ML model (6) is just the inverse of formula 2: since agreeing and disagreeing are binary decisions, the probability of disagreeing is one minus the probability of agreeing. When volunteers disagree with the ML classification, that answer can be taken as evidence about their ability at the chosen classification instead. The parameters, $p(T)$, $p(S)$, $p(G)$, $p(C)$ and initial ability, $p(L_0)$, can thus be estimated by fitting the model to minimize the prediction error for a dataset of responses or through Bayesian sampling.

$$p(L_n|\text{disagree}) = \frac{\big(1 - p(\text{agree}|L_n)\big)p(L_n)}{\big(1 - p(\text{agree})\big)} \tag{6}$$

The same model (specifically (4)) can be used to predict whether volunteers' classifications of images will agree or disagree with the ML classifications given their ability as estimated from answers on previous classifications.

IV. Fitting the Model to Data

In this section, we present an investigation of the model using data drawn from the Gravity Spy system. The purpose of this analysis is first to check the assumptions of the design of Gravity Spy and second to assess the performance of the proposed model. We retrieved all of the classifications recorded on the system as of 28 December 2018, a total of 3,315,590 classifications from 12,986 users on 232,364 glitches (of a total of 297,376), though analyses used only subsets of this data.

A. ML Confidence vs. ML Correctness

We first checked our assumption about the relation between ML confidence and agreement of volunteer judgements with the ML judgements. This analysis was run on the 61,395 glitches for which we could determine ML correctness. We considered the ML as correct if its classification agreed with the expert classification for gold data or with the consensus of the volunteer classifications for retired images. (Glitches are retired from the current version of the system once a sufficient number of human classifications have been contributed.) For retired images, we took the most frequently chosen class as the consensus classification.

Fig. 4 shows a plot of the fraction of ML classifications that agree with the human classification versus ML confidence plotted from high to low, left to right. We determined the average accuracy in 30 bins of equal numbers of glitches, hence the uneven width of the bins. The data show that the ML is quite accurate when confidence is high, but the accuracy drops off as the confidence decreases. We examined these accuracy curves for each class of glitch separately and found generally similar patterns.

B. ML Confidence vs. Volunteer Correctness

We next checked the assumption that images with higher ML confidence are easier for the volunteers to classify, since this is a key assumption for our training system. We followed the same process as above, computing the average volunteer classification accuracy vs. ML confidence across 1,293,569 classifications for the 61,395 glitches for which we had the ML confidence and could determine if the volunteer classification was correct. We determined volunteer correctness in the same way as above. The resulting plot is shown in Fig. 5. The figure shows that the human accuracy is high (between 88% and 96%) regardless of the ML confidence.

We were concerned that this result might be due to the fact

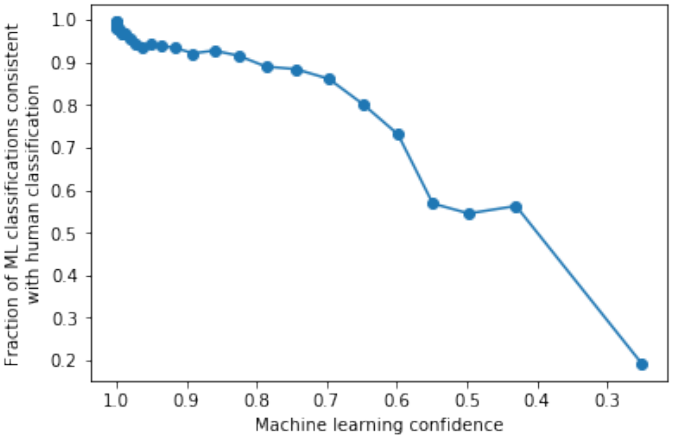


Figure 4. Fraction of ML classifications consistent with human scorers vs. machine learning confidence. N= 61,395.

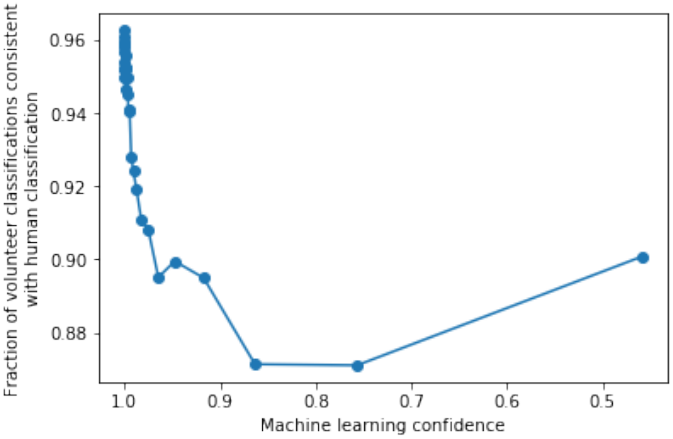


Figure 5. Fraction of volunteer classifications consistent with human scorers vs. machine learning confidence. N=61,395.

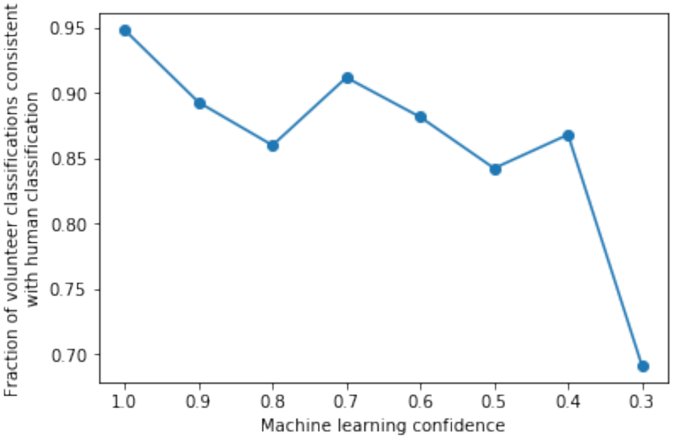


Figure 6. Fraction of volunteer classifications consistent with human scorers vs. machine learning confidence for "gold" data. N=4,071.

that the "correct" classification for a glitch is determined most often from the consensus of volunteer classifications, making the definition circular. To check this effect, we redid the plot for the 4,071 glitches where the correct answer was determined by experts (so-called "gold" data), shown in Fig. 6. For this plot, we used only 10 bins because of the smaller amount of data. This plot shows that the volunteers were generally quite accurate regardless of ML confidence, suggesting that humans and the ML see different things in the data.

Nevertheless, in both this curve and the previous one, volunteer accuracy was high for high levels of ML confidence, confirming that these glitches should be useful for training.

C. *Individual Volunteer Learning*

We next fit the volunteer classification accuracy data against our model. We address in turn the data used, the overall analysis approach and the specific model we developed.

1) *Data*

The first issue was what data to use. We were concerned that data from short-term participants would not be illuminating for the question of how volunteers learn with experience, since those volunteers do not gain experience. Therefore, we dropped 12,831 classifications (less than 1%) from 2,325 volunteers who had contributed 10 or fewer classifications in total and 25,266 classifications that were contributed anonymously. (The fact that 18% of volunteers contributed fewer than 10 classification illustrates the skew in the distribution of the number of classifications performed.) We also dropped classifications of glitches for which we could not determine the correct answer (i.e., not yet retired by the system). Finally, the model's prediction of the probability that a volunteer has not learned decreases with the number of classifications by a factor of $(1-p(T))$ each trial. If $p(T)=0.1$, the chance of not having learned falls below 1% after 44 trials. Therefore, we fit the model against a volunteer's first 60 classification of each class of glitch. We were left with 1,026,652 classifications by 10,655 volunteers on 51,047 glitches.

2) *Analysis Approach*

To fit models to the data, we used the Stan Bayesian analysis system [3]. A Bayesian approach means that rather than being point values, we view model parameters (e.g., $p(T)$) as having a distribution that reflects our uncertainty about their values. In other words, we assume that there are probability distributions for both the observed data and for the parameters of the distributions of the data. For example, we assume that classification data follow a Bernoulli distribution and seek to determine a distribution for the population probability rather than a point estimate of its value. The analysis is Bayesian because we determine the distributions of the parameters by updating an assumed prior distribution with the evidence from the observed data. The prior distributions of the parameters may be based on theory or (as in our case) be "uninformative", e.g., a uniform distribution from 0 to 1 for a probability. Note that this application of Bayes Theory in model estimation is distinct from and should not be confused with its use in the BKT model.

To use Stan, a probability model is written in the Stan programming language describing how observed data depend on parameterized distributions. The model is compiled into a function that computes the log likelihood for a given set of parameter values. Given such a model, one can run an optimizer to find the parameter values that maximize the likelihood. To instead perform a Bayesian analysis, Stan estimates the posterior distributions of the parameters using Markov chain Monte Carlo (MCMC) sampling, i.e., by drawing random samples from the posterior distributions of the parameters (a Monte Carlo estimate). The MCMC algorithm draws samples through a stochastic process where each sample depends on the prior one (i.e., a Markov chain), drawn in a way that the parameters sample the region of highest likelihood. The sampling is seeded with random parameter values drawn from the prior distributions but after some iterations the sampling usually converges on the region of the highest likelihood.

3) *The Model*

Prior work using the BTK model has noted that it is not possible to distinguish empirically between a high initial state of knowledge ($p(L_0)$) and a high rate of successful guessing ($p(G)$) [2], [18]. Reference [18] offered an approach to address the identifiability problem using the forward algorithm. However, we found it difficult to modify the key equations [18, eq. 11–12] to include the uncertainty of an ML classification. We therefore developed an alternative approach to modelling performance.

We noted that the BKT model does not include the possibility of forgetting: learners only transition from not knowing to knowing. As a result, guessing while not knowing is only possible early in a learner's history, before having learned, while slipping is only possible later, after having learned. To capture this transition, we modelled sequential pairs of answers by the same user for the same class of glitch. For two consecutive answers, there are only three possible states of knowledge: not knowing on the first answer and not learning for the second; not knowing at first but learning for the second; and having learned for both.

We develop a model for the probability of pairs of answers in two steps. First, we compute the probability of a learner being in one of the three possible states of knowledge for the two answers after some number of trials as a function of $p(L_0)$ and $p(T)$. For instance, the probability of not knowing then knowing for trials 1 and 2 (i.e., learning between trial 1 and 2) is $(1-p(L_0)p(T))$, not knowing initially but then learning.

Second, for each state of learning, we compute the probability of each of the four possible combination of answers as a function of $p(S)$ and $p(G)$. For instance, if the learner learns only for the second trial, then the probability of answering correctly both times is $p(G)(1-p(S))$, guessing on the first trial when not knowing, then not slipping on the second trial, once having learned. We then sum across the three states of learning to determine the total probability of each of the four possible pairs of answers at each trial given particular parameters.

Finally, to avoid double counting, we only included three of the four counts in the model, since the fourth is determined by the other three, and used only non-overlapping pairs of classifications (i.e., 1st and 2nd, 3rd and 4th, etc.). Pairing the classifications reduced the number of classifications included to a total of 491,428 pairs.

We found that the model did not converge with the default Stan settings, so as suggested by the diagnostics, we increased the target acceptance rate for samples to 0.9, which decreases the sample step size. With these settings, we obtained an estimate of 0.830 for initial level of ability ($p(L_0)$), 0.308 for the probability of learning ($p(T)$), 0.043 for slipping ($p(S)$) and 0.303 for guessing ($p(G)$), with almost no variation across samples. In a Bayesian analysis, the uncertainty in an estimate can be expressed in terms of a high-density interval (HDI), analogous to a confidence interval in a standard analysis. A 95% HDI means the range from the 2.5 percentile to the 97.5 percentile of the parameter values in the samples, i.e., 95% of the sample values are in the HDI. As HDIs are determined by the sampling, they need not be symmetrical, unlike confidence intervals. In this case, the HDIs were the same as the estimates to three decimal places. The estimates for $p(L_0)$ and $p(G)$ were correlated at -0.61, suggesting that these parameters tradeoff somewhat, even though the estimates are precise.

D. Volunteer Agreement with ML

Volunteer accuracy can only be determined *post hoc*, once glitches have been retired and the consensus classification determined. The innovation in this paper is to develop a model for tracing learning based on observed agreement between the volunteer and an ML classification. In this section, we discuss fitting this model to the data.

Table I compares the accuracy of the ML and the volunteer classifications. It shows the number of classifications made grouped by whether the volunteer or the ML was correct or not (93% and 95% correct respectively), and in the case that both were incorrect, whether they picked the same incorrect class (agreed or disagreed, 1.1% and 0.4% respectively). In the cases where the volunteer and the model disagreed, it was about 1.5 times more likely for the volunteer to be wrong. When the volunteer and the ML were both wrong, they agreed in 75% of the cases, which is much higher than would be expected by chance. The high agreement for incorrect choices suggests that volunteers and the ML are being similarly mislead.

A key factor in the agreement model is $p(M_n)$, the expected ML correctness. We determined this value for each glitch by mapping the ML confidence of the glitch being classified to the observed level of agreement in the data of the ML to the human classification (i.e., as shown in Fig. 4). The ML confidence levels were binned into 40 bins and $p(M_n)$ was determined as the average correctness of ML classifications of the glitches in the bin. The average correctness was weighted by the number of classifications: the system is designed to show beginning volunteers glitches of higher levels of confidence, so volunteers are more likely to see correctly classified glitches. We computed accuracy curves separately for each class of glitch.

As with the standard BKT model, we fit the model with individual level data using non-overlapping pairs of answers. We again dropped classifications from anonymous and short-term volunteers, used data for which we knew whether the volunteer agreed with the ML and used only the first 60 trials for the fitting, a total of 247,764 pairs of classifications. There are fewer classifications for this fit because some of the classifications were done for gold data on which the ML was not run. Three of the four combinations are fit as a draw from a Bernoulli distribution (the fourth redundant combination was again omitted).

When we fit the model, we obtained an estimate of 0.844 for initial level of ability, 0.093 for rate of learning, 0.015 for slip and 0.529 for guess. We also estimated $p(C)$, the probability of chance agreement, which we found to be 0.116. The 95% HDIs were $0.844^{+0.008}_{-0.009}$, 0.096±0.003, $0.015^{+0.002}_{-0.001}$, $0.529^{+0.020}_{-0.022}$ and $0.116^{+0.043}_{-0.037}$ respectively. The estimates for $p(L_0)$ and $p(G)$ were correlated at -0.88, though again the estimates were precise. These estimates are similar to the standard BKT model for the initial level of ability ($p(L_0)$) (0.844 vs. 0.830) and for slipping ($p(S)$) (0.015 vs. 0.043) but differ for the rate of learning ($p(T)$) (0.093 vs. 0.308) and for guessing ($p(G)$) (0.529 vs. 0.303). It appears that the second model is attributing agreements to successful guesses rather than to learning.

V. Discussion: Uses for the Models

In this section, we discuss two possible uses for the models.

A. Using the Model for Volunteer Promotion Decisions

First, once estimated on an initial dataset, the model can be used to track learning by volunteers, as in the original BKT model. Specifically, the model can be used to decide when to introduce additional tasks (i.e., to promote a volunteer to the next training level). A key parameter here is the required level of performance. Corbett and Anderson [4] used a threshold of 0.95, though without specific justification for that choice. The required level can be set by considering the desired level of classification performance to make the system work efficiently, as we discuss below in the discussion of image retirement.

A simulation of the model given above with $p(L_0) = 0.844$, $p(T) = 0.093$, $p(S) = 0.015, p(G) = 0.529$ and $p(C) = 0.116$ (the values found from fitting the model) found that if volunteers agree with the ML classification on each image, they reach a predicted probability of knowing how to classify of 0.95 after classifying only 2 images even when given images that are only 0.6 likely to be of the given class. Reaching 0.99 probability takes 5 classifications. The quickness of the learning attribution mostly reflects the high predicted initial level of

Table I. ML versus volunteer classification accuracy.

Volunteer correct	**No**		**Yes**	**Total**
ML correct	*Disagreed*	*Agreed*		
No	4,639 0.4%	14,208 1.1%	44,200 3.4%	63,047 5.4%
Yes	66,424 5.1%		1,164,098 90.0%	1,230,522 94.6%
Total	85,271 7.0%		1,208,298 93.0%	1,293,569

knowledge. If a volunteer disagrees initially, more classifications are needed before the result of the model would reach the required level of predicted ability, as is discussed below.

B. *Deciding Image Classifications*

Second, we consider how the models discussed above can be used for image classification. The goal of the Gravity Spy system is to provide information to the LIGO scientists on the classification of glitches. The system uses judgement from multiple volunteers to make the final decisions on classification of images. In many current Zooniverse systems, each item is classified by a fixed number of volunteers (as many as fifteen) to find a consensus. Explicitly modelling the level of confidence in the classification of an image should make much more efficient use of human effort, as images could be classified with only a few human classifications if the ML confidence is high and the volunteers agree with that classification.

To achieve this end, the system can maintain a model of the likely classification of each image that is initialized by the ML model and prior estimates of accuracy vs. confidence (i.e., $p(M_0)$ from above) and updated with each human classification. As with the volunteer model, we are currently experimenting in the project with different approaches to modelling confidence in the classification of images.

The BKT model developed above for volunteers can be used for images as shown in equations 7–9, drawing on elements defined above. In these equations, *n* is also the number of classifications, but in this case, the number of classifications of a particular image done by different volunteers. $p(M_n)$ is probability that ML classification is correct after *n* volunteer classification and agree / disagree refers to the classification, whether the volunteer agrees or disagrees with ML classification of image. Note that this model includes differences in volunteer ability when forming a belief for the classification of images (that is, the elements of the equations incorporate $p(L)$ for the volunteer making the classification).

$$p(M_{n+1}) = p(M_n|\text{agree}) = \frac{p(\text{agree}|M_n)\, p(M_n)}{p(\text{agree})} \tag{7}$$

$$p(M_n|\text{disagree}) = \frac{\big(1 - p(\text{agree}|M_n)\big)p(M_n)}{\big(1 - p(\text{agree})\big)} \tag{8}$$

$$p(\text{agree}|M_n) = p(\text{correct}) \tag{9}$$

If the level of belief in a particular classification crosses a desired threshold, meaning that there is a consensus among the ML models and the human volunteers on the classification, the image can be retired from the system with that classification. Successfully classified images are provided to the science team to use. They can also be added to the gold standard data and used to retrain the ML model for image classification, thus using human judgement to improve the ML model.

Contrariwise, if after some number of human classifications there is no consensus, then the image can be labelled as none of the above. The efficiency of the process depends on the accuracy of the human labelers. If volunteers slip too often (for example), it is hard to learn from their answers. Fortunately, the data suggest that volunteers are largely accurate.

We simulated image retirement decisions using the model parameters estimated from fitting the model. If volunteers of beginner ability always agree with the ML classification, a glitch can move from 0.6 to 0.999 likely to be of a particular class after 2 classifications (many fewer than the current static number required). If the initial ML confidence is only 0.05 (i.e., if the initial ML classification is for a different class), 3 consistent answers are sufficient for retirement. The quick retirements reflect the high level of belief in the knowledge of the volunteers (i.e., a high value for $p(L_0)$).

C. *Simulating Promotion and Retirement with Real Data*

The above simulations have assumed that volunteers always agree with the ML classification, which is not the case. Therefore, we simulated promotion and retirement decisions using the actual pattern of agreements or disagreements in the classification data (using all of the data). We set $p(M_0)$ to the predicted ML accuracy for the ML's predicted class (as discussed above) or 0.05 for other classes (approximately the average ML error rate). In these simulations, we tracked volunteer performance on both their choice and the ML-predicted class when these disagreed. The required level of performance for both volunteers and images was set to 0.99.

The results from this simulation for volunteer promotion decisions are shown in Table II and compared to the actual performance of the current system. Classification is the count of classifications of glitches that must be trained to be promoted to that level. (The mean and standard deviation reported for the actual data are of 95% trimmed data to remove outliers, e.g., problems that might have been created as the promotion system

Table II. Comparison of simulated promotion decisions to actual promotions in the current system, showing number of new classes to be learned, mean number of classifications needed to achieve 0.99 learning, and number and fraction of volunteers promoted. (Actual data are 95% trimmed.)

		Simulated				**Actual classifications**		
Level	**Classes**	**classifications**		**N**	**%**			**N**
		Mean	**SD**			**Trimmed mean**	**Trimmed SD**	
2	2	5.2	4.6	11,504	88.6%	33.7	29.6	6,429
3	3	12.6	18.4	4,975	38.3%	68.8	60.6	3,741
4	5	43.4	66.1	2,226	17.1%	213.3	178.1	1,697

was being worked on.) In the current system, each level also includes instances of glitches trained in the previous levels, which about doubles the total number classifications that have to be done at each level. Nevertheless, it can be seen that the proposed system would promote more volunteers more quickly on average, with the exception of level 5. The averages are reasonable for the amount of work expected of volunteers, though with high variance.

The number of classifications required is not linear in the number of classes to be learned: 2.6 instances per class to reach level 2, but 25.7 instances per class for the one volunteer who reached level 5. This effect is partly due to the increased difficulty of the classes, partly to the potential confusion with more new classes and largely due to the need to see a sufficient number of the glitches of all of the classes in one level before promotion to the next. It can be shown that the expected number of classifications needed to see all of the classes grows as the square of the number of classes (assuming classes are equally likely and uniformly distributed). Promotion from level 1 to 2 requires learning only 2 classes; from level 4 to 5, 12 classes, implying 36 times as many classifications on average. To reduce the time to promotion, level 4 of the system was recently split into two levels each with half the new classes. The simulation shows that this change would result in 313 volunteers being promoted to the intermediate level and 10 to the top level vs. only 1 with the original configuration.

Considering glitches, the simulation retired 140,512 of the 228,415 non-gold glitches seen (61.5%), with an average of only 2.7 classification each (sd = 2.9). This measure of system performance is likely an underestimate, as if the glitches had actually been retired, other glitches would have been shown to volunteers, perhaps enabling them to gain enough classifications to also be retired.

79.2% of the retired glitches were retired as the class chosen by the ML, in 2.0 classifications on average (indicating that the initial estimate of the ML accuracy and the volunteer ability were high). Retirements that disagreed with the ML took 5.3 classifications on average. The small number of classifications needed to overturn the prior ML classification suggests that a high level of expertise has been estimated for the volunteers.

VI. Conclusion

In this paper, we have presented the design of a system that uses ML classifications of images to guide training for human volunteers in a citizen-science project. The goal of the training is to help volunteers more quickly learn how to classify images while making productive contributions to the project.

The model presented in this paper fits the observed learning well and does demonstrate learning overall. Of course, to properly test the value of the training regime will require an experiment comparing the performance of volunteers with and without training. We hope to report on such an experiment soon. We further expect that the training will also motivate users to contribute more. If the system works as expected, the training approach presented here that should be of interest to other citizen-science projects.

A further important benefit of the training approach described here is that because the ML cannot be certain of the classification, having a volunteer confirm the classification—even a beginner still being trained—is still useful to the project. This approach contrasts with training that is either entirely preset or that relies exclusively on gold-standard data. In those cases, the work done by the volunteer as part of the training does not directly advance the project's work. As many volunteers report that they are motivated by the fact that they are contributing to science [14], it is important to ensure that the work done is real to maintain volunteer interest.

The system described above also offers an interesting platform for further experimentation. First, the training system described above has a large number of parameters (e.g., how many and which classes to introduce at each level, the ML certainty cutoffs or the right mix of images of different certainties at different points in the process). Experimentation will be useful to determine the optimal settings. For example, we can test the benefits and tradeoffs of advancing volunteers more quickly: quicker advancement might be good for motivation but negative for performance (and vice versa).

As well, the system enables us to experiment with other factors that affect volunteer performance, e.g., the kinds of motivational messages provided or information on the novelty of images. A particularly interesting set of questions are around the effects of feedback that can be provided to volunteers based on the ML certainties. Again, it is possible that there are tradeoffs involved, e.g., that letting a volunteer know the result of the ML evaluation might be useful feedback to improve performance but also potentially demotivating if the ML and the volunteer disagree or volunteers feel that their contributions are unnecessary given the capabilities of the ML. A further problem is that this approach to feedback runs the risk of training the human volunteers in the idiosyncrasies of the ML, thus reducing the benefit of having diverse kinds of classifiers in the system. These effects need to be carefully considered by introducing such feedback.

Future research could explore extensions to BKT model such as modelling forgetting [8] or making individual estimates of the model parameters (e.g., [12]). A first step is to look for evidence that volunteers' accuracy in fact drops after a break. A possibility to handle such forgetting between sessions is, at the start of each session, to restart volunteers who have not yet mastered some classes to the default estimated initial level for the classes still to be learned. Those who are in the highest level of the system, having learned all the classes, might also have their estimated abilities reduced after a long gap in contribution.

The models suggest two additional ways to improve system performance. First, the system can pick images for the volunteers to classify that will be particularly informative for improving the ML models (e.g., images that have confidence levels between the cutoffs), a process called active learning.

Second, since the system is tracking each volunteer's ability, it can also assign tasks based on ability (e.g., assigning harder tasks to more capable volunteers). However, as [10] point out, when picking an item to be classified in a crowdsourcing setting, the number of existing classifications should be considered. If the item already has many human classifications, another classification will not reduce the ML model uncertainty.

The contribution of this paper has been to discuss how machine learning can be used to support learning in a citizen science project and to present a Bayesian model for tracking learning progress in this setting. The system thus implements a redesigned relationship between the technology of the system and the human volunteers to facilitate learning by both.

Acknowledgment

We thank the many citizen scientists whose engagement have made Gravity Spy possible and our collaborators on the project. Our gratitude also goes to the review team whose comments have improved this paper.

References

[1] R. S. J. d. Baker, A. T. Corbett, and V. Aleven, "More accurate student modeling through contextual estimation of slip and guess probabilities in Bayesian knowledge tracing," in *Proc. Intelligent Tutoring Systems*, 2008, pp. 406–415. doi: 10.1007/978-3-540-69132-7_44

[2] J. E. Beck and K.-m. Chang, "Identifiability: A fundamental problem of student modeling," in *Proc. Int. Conf. User Modeling*, 2007, pp. 137–146

[3] B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell, "Stan: A probabilistic programming language," *J. Statist. Softw.,* vol. 76, 2017. doi: 10.18637/jss.v076.i01

[4] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Model. User-Adapted Interact.,* vol. 4, pp. 253–278, 1995

[5] K. Crowston and I. Fagnot, "Stages of motivation for contributing user-generated content: A theory and empirical test," *Int. J. Hum. Comput. Stud.,* vol. 109, pp. 89–101, 2018. doi: 10.1016/j.ijhcs.2017.08.005

[6] N. Ducheneaut, "Socialization in an open source software community: A socio-technical analysis," *Comput. Supp. Coop. Work,* pp. 323–368, 2005

[7] J. Kasurinen and U. Nikula, "Estimating programming knowledge with Bayesian knowledge tracing," in *Proc. ACM SIGCSE Conf. Innovation Technology Computer Science Education*, Paris, France, 2009, pp. 313–317. doi: 10.1145/1562877.1562972

[8] M. Khajah, R. V. Lindsey, and M. C. Mozer, "How deep is knowledge tracing?," *arXiv preprint arXiv:1604.02416,* 2016

[9] M. M. Khajah, B. D. Roads, R. V. Lindsey, Y.-E. Liu, and M. C. Mozer, "Designing engaging games using Bayesian optimization," in *Proc. Conf. Human Factors Computing Systems*, Santa Clara, California, USA, 2016, pp. 5571–5582. doi: 10.1145/2858036.2858253

[10] C. H. Lin and D. S. Weld, "Re-active learning: Active learning with relabeling," in *Proc. AAAI Artificial Intelligence*, 2016

[11] S. Malinen, "Understanding user participation in online communities: A systematic literature review of empirical studies," *Comput. Hum. Behav.,* vol. 46, pp. 228–238, 2015

[12] Z. A. Pardos and N. T. Heffernan, "Modeling individualization in a Bayesian networks implementation of knowledge tracing," in *Proc. Int. Conf. User Modeling Adaptation Personalization*, 2010, pp. 255–266

[13] N. R. Prestopnik, K. Crowston, and J. Wang, "Gamers, citizen scientists, and data: Exploring participant contributions in two games with a purpose," *Comput. Hum. Behav.,* vol. 68, pp. 254–268, 2017

[14] M. J. Raddick, G. Bracey, P. L. Gay, C. J. Lintott, P. Murray, K. Schawinski, A. S. Szalay, and J. Vandenberg, "Galaxy Zoo: Exploring the motivations of citizen science volunteers," *Astron. Educ. Rev.,* vol. 9, pp. 010103–18, 2010. doi: 10.3847/AER2009036

[15] B. D. Roads and M. C. Mozer, "Improving human-machine cooperative classification via cognitive theories of similarity," *Cogn. Sci.,* 2016. doi: 10.1111/cogs.12400

[16] J. B. Tenenbaum, "Bayesian modeling of human concept learning," in *Advances in Neural Information Processing Systems*. vol. 11, M. Kearns, S. Solla, and D. Cohn, Eds. Cambridge, MA: MIT Press, 1999, pp. 59–65.

[17] R. Tinati, M. Luczak-Roesch, E. Simperl, and W. Hall, "An investigation of player motivations in Eyewire, a gamified citizen science project," *Comput. Hum. Behav.,* vol. 73, pp. 527–540, 2017. doi: 10.1016/j.chb.2016.12.074

[18] B. van de Sande, "Properties of the Bayesian Knowledge Tracing Model," *J. Educ. Data Min.,* vol. 5, pp. 1–10, 2013

[19] M. Zevin, S. Coughlin, S. Bahaadini, E. Besler, N. Rohani, S. Allen, M. Cabero, K. Crowston, A. Katsaggelos, S. Larson, T. K. Lee, C. Lintott, T. Littenberg, A. Lundgren, C. Oesterlund, J. Smith, L. Trouille, and V. Kalogera, "Gravity Spy: Integrating Advanced LIGO detector characterization, machine learning, and citizen science," *Classical Quantum Gravity,* vol. 34, 2017. doi: 10.1088/1361-6382/aa5cea

Kevin Crowston received his Ph.D. in 1991 in Information Technologies from the Sloan School of Management at the Massachusetts Institute of Technology.

He is currently a distinguished professor and the associate dean for research at the School of Information Studies, Syracuse University, Syracuse, NY US. From 1992 to 1994 he was a program director for the Cyber-Human Systems & Human-Centered Computing Program of the United States National Science Foundation**.** He is an author of more than 130 journal and referred conference papers. He currently serves as the Editor-in-Chief of the journal *ACM Transactions on Social Computing* and co-EIC of *Information, Technology & People*.

Prof. Crowston is a member of the Academy of Management, Association for Computing Machinery (ACM), Association for Information Systems, IFIP Working Groups 8.2 and 2.13 and the Society for Industrial and Organizational Psychology. He has received best published paper awards in information systems and in social informatics, and the Paul Gray Award for the Most Thought Provoking Paper in Information Systems. He has been recognized with the IFIP Outstanding Service and Silver Core Awards, the Academy of Management Lifetime Service Award and the Research in Information Science Award from the Association for Information Science & Technology (ASIS&T), for "outstanding research contribution in the field of information science".

Carsten Østerlund received his Ph.D. in management from the Sloan School of Management at the Massachusetts Institute of Technology in 2003.

He is a professor at the School of Information Studies at Syracuse University and the author of three books and more than 70 articles. His research interests include the organization, creation, and use of documents and other sociomaterial practices in distributed work environments, with an emphasis on learning and knowledge dynamics in new forms of work. Empirically, he studies these issues through in-depth qualitative studies of everyday work practices in a range of settings including citizen science, crowdsourcing, distributed science teams, and healthcare. Recently, he has been particularly interested in how we can merge qualitative and quantitative methodologies to study trace data. He is an Associate Editor of the journals *Information System Research* and *Information & Organization*.

Prof. Østerlund is a member of the Association for Computing Machinery (ACM), Academy of Management, Association for Information Systems, IFIP Working Group 8.2 and Society for Social Studies of Science. He was a recipient of the Robert Benjamin Junior Faculty Award, Outstanding Research & Scholarship and the Jeffrey Katzer Professor of the Year, Outstanding Teaching and Advising Award from the School of Information Studies, Syracuse University and the Diana Forsythe Award from the American Medical Informatics Association in 2005.

Tae Kyoung Lee was born in South Korea. She received the B.A. degree in advertising and public relations from Ewha Womans University, Seoul, South Korea, the M.A. degree in communication from the University of California, Davis, CA, in 2010 and the Ph.D. degree in communication from Cornell University, Ithaca, NY, in 2016.

She was a Postdoctoral Researcher in Spring 2016 at the Syracuse University School of Information Studies. Since Fall 2016, she has been an Assistant Professor in the Department of Communication at the University of Utah, Salt Lake City, UT. Her research interests include message processing.

Prof. Lee is a member of International Communication Association and National Communication Association.

Corey Jackson received a B.A. in political science from the University of Illinois Urbana-Champaign (UIUC), Urbana, IL, US in 2010, an M.S. in library and information science from the Graduate School of Library and Information Science at UIUC in 2012, and a Ph.D. in information science and technology from the School of Information Studies at Syracuse University, Syracuse, NY, US.

From 2012 to 2019, he was a Research Assistant at the School of Information Studies at Syracuse University. Since 2019, he has been a Postdoctoral Scholar with the School of Information, University of California, Berkeley, Berkeley, CA, USA. His research interest includes phenomena such as contribution behaviors, learning and motivation of users to large-scale socio-technical systems.

Dr. Jackson is a member of the ACM.

Mahboobeh Harandi received a B.S. degree in software engineering from the University of Science and Culture, Tehran, Iran in 2005 and the M.S. degree in Information System Engineering from Norwegian University of Science and Technology (NTNU), Trondheim, Norway in 2015. She is currently pursuing a Ph.D. degree in information science and technology at Syracuse University, Syracuse, NY, USA.

From 2007 to 2012, she was a software developer, and from 2012 to 2015, she was a student researcher in computer science (smart media project) and psychology department (NuLab) at NTNU. From 2016 she has been involved as a research assistant in various research projects to understand human behaviors in socio-technical systems.

Ms. Harandi is a member of the ACM.

Sarah R. Allen was born in Phoenix, Arizona in 1986. She received a B.A in music from the University of Arizona in 2008.

From 2008 to 2011, she worked for the University of Arizona's College of Medicine, Department of Family and Community Medicine as an IT Support Center Specialist managing the server infrastructure, technology purchasing, and help desk. In 2011, she moved to Chicago, Illinois and joined Northwestern University, Feinberg School of Medicine as an Application Support Specialist. Beginning in 2013, she shifted her IT career into web development and joined the Zooniverse at the Adler Planetarium in 2014. Currently, she is leading the front-end development of the Zooniverse's main project builder platform as a Senior Web Developer.

Ms. Allen received a nomination for the Team Award for Excellence at the University of Arizona in 2013 as a member of the IT committee who assisted the university's transition to a new email hosting system starting in 2010.

Sara Bahaadini received her B.S. in computer science and engineering from Shiraz University, Iran in 2008 and her M.S. degree in Artificial Intelligence from Sharif University of Technology, Iran in 2011. She recently defended her Ph.D. in Computer Science Department at Northwestern University, Evanston, IL.

From 2013 to 2014, she was research assistant at Idiap Research Institute, affiliated with EPFL University in Switzerland. In summer 2018, she joined NVIDIA as a deep learning summer intern working on human pose estimation for robots. Her Ph.D. thesis focused on leveraging deep neural networks for learning discriminative feature representations for various types of data such as image and time series.

Dr. Bahaadini has more than 8 years of research experience on several projects involving applications of machine learning and deep learning. She has published in prestigious peer-reviewed venues such as IEEE proceeding, ICIP, ICASSP, and Interspeech. She is also the recipient of the Royal E. Cabell Fellowship and Walter P. Murphy Fellowship of Northwestern University.

Scott Coughlin received a B.A. in mathematics, economics and classics in 2014 from Northwestern University and a M.S. in gravitational wave physics from Cardiff University. He is currently pursuing a Ph.D. at Cardiff University.

He has been an undergraduate and is a graduate research assistant at the Center for Interdisciplinary Exploration and

Research in Astrophysics (CIERA) at Northwestern University, in Evanston, IL.

Mr. Coughlin received an honorable mention in the 2018 LIGO Laboratory Awards for Excellence in Detector Characterization and Calibration for his role in the creation of the Gravity Spy system.

Aggelos K. Katsaggelos (S'80–M'85–SM'92–F'98) received the Diploma degree in electrical and mechanical engineering from the Aristotelian University of Thessaloniki, Greece, in 1979, and the M.S. and Ph.D. degrees in electrical engineering from the Georgia Institute of Technology in 1981 and 1985, respectively.

In 1985, he joined the Department of Electrical Engineering and Computer Science, Northwestern University. He was the Ameritech Chair of information technology and the AT&T Chair. He is currently a Professor of the Joseph Cummings Chair, Northwestern University. He is also an Academic Staff Member with NorthShore University Health System and an Affiliated Faculty Member with the Department of Linguistics. He has an appointment with the Argonne National Laboratory. He has published extensively in the areas of multimedia signal processing and communications, computational imaging, and machine learning (over 250 journal papers, 600 conference papers, and 40 book chapters). He holds 25 international patents. He has co-authored Rate-Distortion Based Video Compression (Kluwer, 1997), Super-Resolution for Images and Video (Claypool, 2007), Joint Source-Channel Video Transmission (Claypool, 2007), and Machine Learning Refined (Cambridge University Press, 2016). He has supervised 56 Ph.D. dissertations. Among his many professional activities, he was the Editor-in-Chief of the IEEE Signal Processing Magazine from 1997 to 2002, a BOG Member of the IEEE Signal Processing Society from 1999 to 2001, a member of the Publication Board of the IEEE Proceedings from 2003 to 2007, and a member of the Award Board of the IEEE Signal Processing Society.

Prof. Katsaggelos became a fellow of the IEEE in 1998, SPIE in 2009, EURASIP in 2017, and OSA in 2018. He was a recipient of the IEEE Third Millennium Medal in 2000, the IEEE Signal Processing Society Meritorious Service Award in 2001, the IEEE Signal Processing Society Technical Achievement Award in 2010, the IEEE Signal Processing Society Best Paper Award in 2001, the IEEE ICME Paper Award in 2006, the IEEE ICIP Paper Award in 2007, the ISPA Paper Award in 2009, and the EUSIPCO Paper Award in 2013. He was a Distinguished Lecturer of the IEEE Signal Processing Society from 2007 to 2008.

Shane L. Larson received his B.S. in Physics from Oregon State University, Corvallis OR in 1991, his M.S. in Physics from Montana State University, Bozeman MT in 1994, and his Ph.D. from Montana State University, Bozeman MT in 1999.

He is currently the Associate Director of CIERA and was formerly an Associate Professor of Physics at Utah State University. His research interests are in gravitational wave astrophysics, particularly with regard to astrophysical sources for the LIGO and LISA gravitational wave observatories.

Prof. Larson is a Fellow of the American Physical Society.

Neda Rohani was born in Bandarabbas, Hormozgan, Iran in 1986. She received the B.S. and M.S. degrees in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2004 and 2008 and the Ph.D. degree in computer science from Northwestern University, Evanston, IL, in 2018.

From 2014 to 2018, she was a Research Assistant with Image and Video Processing Laboratory. Since 2019, she has been an Applied Scientist with Microsoft Search, Assistant and Intelligence (MSAI), Microsoft Corporation, Bellevue, WA. Her research interests include machine learning, deep learning, natural language processing and computer vision.

Dr. Rohani was a recipient of the Integrated Data-Driven Discovery (IDEAS) Fellowship in 2015 and Walter P. Murphy Fellowship in 2014.

Joshua R. Smith was born in Indian Lake, NY and attended Syracuse University, graduating with a B.S. in 2002. He earned his doctorate in 2006 from the University of Hannover's Max Planck Institute for Gravitational Physics / Albert Einstein Institute for his work on the GEO600 gravitational-wave detector.

He currently directs the Gravitational Wave Physics and Astronomy Center and is a professor of physics at California State University, Fullerton. Prior to joining Fullerton in 2010, he was a postdoctoral research associate at Syracuse University. His research is focused on optics experimentation and observing gravitational waves from astronomical sources using the laser interferometric gravitational wave observatory (LIGO).

Prof. Smith is a member of the Optical Society of America, the Society for Advancement of Chicanos and Native Americans in Science, the American Astronomical Society and the American Physical Society.

Laura Trouille was born in Evanston, IL, USA in 1981. She received the B.A. degree in physics from Dartmouth College, Hanover, NH in 2003 and the Ph.D. degree in astronomy from the University of Wisconsin, Madison in 2010.

From 2010 to 2012, she was a CIERA Postdoctoral Fellow at Northwestern University in Evanston, IL. From 2012–2015 she was a jointly appointed astronomer at Northwestern University and The Adler Planetarium in Chicago, IL. Since 2015 she was first Senior Director and now Vice President of Citizen Science at the Adler Planetarium. She is an author on more than 30 peer-reviewed publications. Her research interests include supermassive black holes and galaxy evolution, computational thinking in STEM, and citizen science. She is co-PI for the Zooniverse online citizen science platform and leads the Adler's Teen Programs efforts.

Dr. Trouille is a member of the American Astronomical Society and the American Geophysical Union.

Michael Zevin was born in Burr Ridge, IL in 1990. He received his B.S. in astronomy, physics, and music from the University of Illinois Urbana-Champaign in 2012 and his M.S. in physics and astronomy from Northwestern University in 2016. He is current a PhD candidate at Northwestern University.

From 2012–2014, he worked as a science educator at the Adler Planetarium in Chicago and a science teacher at Kids Science Labs, a science learning center based in Chicago. As a PhD student at Northwestern University, he has led multiple papers and been a co-author on over 100 publications, both as short-author works and as part of the LIGO Scientific Collaboration. His research interests include gravitational waves, compact object astrophysics, and stellar evolution.

Mr. Zevin is a member of the American Astronomical Society and the American Physical Society.