Predicting the Working Time of Microtasks Based on Workers' Perception of Prediction Errors

SUSUMU SAITO, WASEDA UNIVERSITY

CHUN-WEI CHIANG, WEST VIRGINIA UNIVERSITY

SAIPH SAVAGE, UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO (UNAM)

TEPPEI NAKANO, WASEDA UNIVERSITY

TETSUNORI KOBAYASHI, WASEDA UNIVERSITY

JEFFREY P. BIGHAM, CARNEGIE MELLON UNIVERSITY

ABSTRACT

Crowd workers struggle to earn adequate wages. Given the limited task-related information provided on crowd platforms, workers often fail to estimate how long it would take to complete certain microtasks. Although there exist a few third-party tools and online communities that provide estimates of working times, such information is limited to microtasks that have been previously completed by other workers, and such tasks are usually booked immediately by experienced workers. This paper presents a computational technique for predicting microtask working times (*i.e.*, how much time it takes to complete microtasks) based on past experiences of workers regarding similar tasks. The following two challenges were addressed during development of the proposed predictive model — *(i)* collection of sufficient training data labeled with accurate working times, and *(ii)* evaluation and optimization of the prediction model. The paper first describes how 7,303 microtask submission data records were collected using a web browser extension — installed by 83 Amazon Mechanical Turk (AMT) workers — created for characterization of the diversity of worker behavior to facilitate accurate recording of working times. Next, challenges encountered in defining evaluation and/or objective functions have been described based on the tolerance demonstrated by workers with regard to prediction errors. To this end, surveys were conducted in AMT asking workers how they felt regarding prediction errors in working times pertaining to microtasks simulated using an "imaginary" AI system. Based on 91,060 survey responses submitted by 875 workers, objective/evaluation functions were derived for use in the prediction model to reflect whether or not the calculated prediction errors would be tolerated by workers. Evaluation results based on worker perceptions of prediction errors revealed that the proposed model was capable of predicting worker-tolerable working times in 73.6% of all tested microtask cases. Further, the derived objective function contributed to realization of accurate predictions across microtasks with more diverse durations.

1.   INTRODUCTION

Crowd workers often struggle to earn appropriate wages (Irani and Silberman, 2013; McInnis et al., 2016; Ipeirotis, 2010). This is one of the largest problems affecting current crowd markets, considering that the main motivation of workers is to earn sufficient wages to make a living (Berg, 2015; Brewer et al., 2016; Kuek et al., 2015; Martin et al., 2014). The key challenge in the quest to earn more lies in selecting potentially lucrative microtasks by predicting if they are worth their suggested prices via estimation of their working times — (*i.e.,* time required to complete a microtask). Unfortunately, the said working-time estimation is difficult, because workers are only provided "raw" information about microtasks in the form of simple textual descriptions or previewed user interfaces, which do not directly indicate how long completion of a certain microtask would take. In this study, we propose a method for working-time prediction for a given microtask based on various types of work-relevant information provided on crowd-worker forums even before the microtask is started by a worker.

Existing techniques for working-time evaluation are available for only those microtasks with previous working histories for which have been collected from their users. There exist online platforms[1][2][3] and worker tools proposed by researchers (Callison-Burch, 2014; Hanrahan et al., 2015) that leverage working records collected from users to calculate working times, thereby suggesting which microtasks are likely the most lucrative. However, such working-time calculation is not always possible if microtask completion history is not previously provided by any worker. Considering that suggested microtasks are already popular and competitive in most cases, and that new microtasks are frequently posted every day, working-time estimation for previously unseen microtasks becomes all-the-more important.

In this study, we propose a method for predicting the working time for any microtask based on working histories of other microtasks completed by workers. This paper is an extension of our previous work, wherein we proposed TurkScanner system (Saito et al., 2019a), which outputs working times in seconds via use of machine learning-based regression and different types of work-relevant information, such as *microtask information* (microtask HTML elements and microtask metadata), *worker information* (e.g., basic worker profile, dashboard information), and *requester information* (requester reputation posted in Turkopticon (Irani and Silberman, 2013)), most of which can be obtained before workers actually start working on a microtask. Through use of the proposed approach, it is expected that microtasks that contain specific keywords, many input elements, or longer time limits would take longer to complete; workers with more experience would finish microtasks earlier; and requesters with more and better reviews are good at designing microtasks that run smoothly.

There are a few possible ideas for real-world applications. First, our working time prediction system could be developed as a worker tool. For instance, when a worker seeks microtasks in a crowd platform, the tool could estimate working time and hourly wage of every single microtask and

---

[1]Turkopticon, https://turkopticon.ucsd.edu/
[2]Turkopticon 2, https://turkopticon.info/
[3]TurkerView, https://www.turkerview.com

visualize it in the searched microtask list. With this, the worker can check the potential values of all available microtasks at a glance, which can assist workers' efficient decision making of selecting lucrative microtasks. Another example is an application as a requester tool. In general, workers encounter cheap microtasks because requesters created them. That is, if the requesters know a proper standard of microtask pricing and calculate their microtask prices accurately based on the standard, the risk that workers choose cheap microtasks could be avoided. To this end, a requester tool could suggest a proper price for a microtask that was created and uploaded to the tool by a requester.

The following two challenges were encountered during development of the proposed system.

i) **Data collection:** For successful implementation of a statistical approach for predicting working times, it is necessary to collect reliable ground-truth data comprising actual working times of previously executed microtasks. Automatic gauging of working times is difficult to realize. Additionally, there is no common means to calculate working times, because worker behavior is diverse (Kaplan et al., 2018) and involves actions, such as visiting external websites as part of the work or just for checking emails, leaving their work desk for breaks, and accepting multiple microtasks in a batch and completing them in succession (Hara et al., 2018). To this end, a browser extension was developed to be installed by crowd workers on their machines. The extension collected three different times (two recorded automatically and one manual recording triggered by workers at the click of a button), and workers were asked to select the one that was likely the most accurate. The selected working time was then added as a label to microtask data scraped in the background to be used as input features. Among all data collected after deploying the script to Amazon Mechanical Turk (AMT) workers, 7,303 valid records of microtask submissions were extracted from 83 unique workers for use in performance evaluation of the working-time prediction system.

ii) **Defining worker perception for model optimization & evaluation:** Because our research aims at helping workers formulate working-time expectations, the proposed model must be optimized and evaluated based on how meaningful the information provided to workers is, and this cannot be determined exclusively by using objective values (*i.e.,* seconds). Psychologists have determined that there often exist differences between objective values of a certain stimulus and subjective human perception of the same. We assumed that this was equally applicable to working times of crowdsourcing microtasks. For instance, a prediction error of "30-second difference" for $(predicted, actual) = (30s, 60s)$ would not likely be as acceptable as $(1030s, 1060s)$, whereas that of "100% difference" in the case of the former would not be as problematic as $(1500s, 3000s)$. It is therefore believed that any statistic model for time estimation must always be optimized and evaluated in accordance with human perception. Otherwise, the overall or per-(arbitrary-)category accuracy, at best, of worker tools that depend solely on objective measurements could be evaluated exclusively in either seconds or percentages, thereby resulting in lower satisfaction among users. To understand the human perception of microtask working times, we performed a subjective survey randomly and repeatedly among AMT workers, where they were suggested a pair of predicted and actual working times for an imaginary microtask. During the survey, workers were asked whether the shown prediction error was acceptable to them. In the survey, 91,060 data samples were collected from 875 unique workers to obtain CrowdSense, a set of evaluation results demonstrating the workers' perception of whether or not they accepted the prediction error between the predicted and actual working times for a hypothetical microtask. With reference to CrowdSense, it was concluded

that the proposed working-time prediction model was capable of accurately predicting ∼73% of all tested microtasks in the dataset in accordance with our empirical definition based on worker acceptance of prediction errors. CrowdSense was additionally leveraged for model optimization, and it was observed that the proposed optimization approach facilitated accurate prediction of working times over a more diverse range compared to baseline methods.

The remainder of this paper has been structured as follows. Section 2 presents a review of extant previous literature investigating the problem of low wages of crowd workers followed by a discussion pertaining to online communities and worker tools that provide estimated working times and other information regarding available microtasks. Relevant literature defining subjective perception measurement across different research domains has also found mention. Section 3 explains the data collection approach employed in this study using a browser script, and results obtained thereof have been discussed. Section 4 describes the design of the survey conducted in this study along with results pertaining to workers' perception of prediction errors as well as objective and evaluation functions. Section 5 explains experimental settings and corresponding results for evaluating performance of CrowdSense — both overall and across different time scales. Section 6 discusses key limitations encountered in this study as well as relevant future endeavors planned. Lastly, major conclusions drawn from this study are discussed in section 7.

2. RELATED WORK

This study aims to solve a problem of unfair treatment and underpaying of crowd workers reported by several extant studies. In this section, we present a brief review of previous studies that point out the unfairness prevalent in crowd markets, followed by an introduction to systems that calculate the working time of microtasks to assist workers in their quest to earn higher wages. Ultimately, we mention discuss related work defining human subjective perception, a subject that is expected to pertain to the development of the proposed working-time prediction system to optimize prediction performance and system evaluation subsequently discussed in this paper.

2.1. Unfair Pay in Crowd Markets

Crowd workers are generally underpaid (Horton, 2011; Katz, 2017; (ILO), 2016; Durward et al., 2016; Thies et al., 2011). Because the main motivation of crowd workers is to earn sufficient wages (Martin et al., 2014), ensuring adequate pay forms an important aspect of preserving the crowd-working environment. To address this concern, several researchers have proposed approaches to help workers increase their income (Chiang et al., 2018; Coetzee et al., 2015; Dontcheva et al., 2014).

Several studies report that the power imbalance between requesters and workers is one of the main causes of the low-wage problem (Salehi et al., 2015; Silberman et al., 2010; O'neill and Martin, 2013). Requesters are usually assigned broad capabilities when posting microtasks. They are allowed to create microtasks freely, and are not only provided with microtask templates for easy task creation but also allowed to build their own systems and navigate workers to their site for execution of complex and unique microtasks. Additionally, requesters can set any price they feel appropriate for execution of microtasks created by them. It is known that several microtasks are set a very low price by requesters not having sufficient experience or who try to save money without consideration of worker welfare. For execution of their microtasks, requesters can instantly hire the desired

number of workers whenever needed through use of features that facilitate screening (Mason and Suri, 2012; Litman et al., 2017), blocking (Karger et al., 2011), and rejecting (Bederson and Quinn, 2011; Wu and Quinn, 2017) workers they do not like.

In contrast, workers are usually provided very limited task-related information on crowdsourcing platforms (Irani and Silberman, 2013; Chilton et al., 2010; Alsayasneh et al., 2017). On major platforms, such as Amazon Mechanical Turk[4], Microworkers[5], and Prolific[6], workers are typically provided with the requester's name, number of remaining slots in the microtask batch, reward amount, allotted time, microtask title (along with a brief description), and an example of the microtask interface. To earn efficiently, workers need to immediately judge, based on available data, as to which microtask would provide the best benefit with regard to maximizing their wage. A failure to do so would result in workers taking up sub-optimal microtasks that require too much completion time for the suggested price, thereby making their work routines less efficient. Previous studies related to worker ethics have suggested that the United States' minimum wage be considered the lower limit for microtask hourly wage (Barowy et al., 2017; Hara and Bigham, 2017); however, in many cases, workers fail to estimate the hourly wage because they do not know how to evaluate the given information.

Thus, workers often miss opportunities to earn more because of the power imbalance between requesters and workers. This, in turn, results in many workers being paid less than their rightful minimum wage (Irani and Silberman, 2013; McInnis et al., 2016; Ipeirotis, 2010; Hitlin, 2016; Horton and Chilton, 2010; Irani and Silberman, 2016; Martin et al., 2014). The above discussion clearly emphasizes the need for development of better methods to help crowd workers earn higher wages.

2.2.   **Estimating Working Time of Microtasks**

An important information to determine benefits of a given microtask is to know its total working time (*i.e.,* how long it would take to finish the microtask) prior to beginning working on it. Several researchers and practitioners have explored means to accurately estimate microtask working times, thereby assisting workers to get better at microtask selection (McInnis et al., 2016; Chiang et al., 2018).

Crowd workers often leverage the availability of online communities and worker tools to achieve better work efficiency (Kaplan et al., 2018). Among the many existing options, Turkopticon and TurkerView are major online communities where fellow workers post reputations of requesters and microtasks. In addition, these sites offer worker tools that utilize posted information. The said tools not only collect five-grade evaluations of several reputation criteria and comments from workers but also ask workers to list corresponding working times to facilitate calculation of the average working time for the microtasks. The same can then be used to calculate their hourly wages. However, the said tools employ history-based methods that can estimate working times for only a limited number of microtasks, which have already been executed by other workers, who presumably provided their relevant working time to the system. Considering that experienced workers often use other tools,

---

[4]https://www.mturk.com/

[5]https://www.microworkers.com/

[6]https://www.prolific.co/

such as Panda Crazy[7], that "auto-accept" popular microtasks with high requester ratings, and that new microtasks are posted in platforms on a daily basis (Saito et al., 2019a), it is believed that the current scope for assistance is so small that several workers would remain unaided.

Researchers have also explored several means to estimate microtask working times. CrowdWorkers (Callison-Burch, 2014) is a browser extension for AMT workers that records user working times for each submitted microtask and calculates the estimated working time and hourly wage for other users. TurkBench (Hanrahan et al., 2015) is a web tool that recommends which microtask to take up next (based on its hourly wage) and auto-accepts the microtasks, thereby enabling workers to maximize their hourly wages. TurkBench records working times in the background and leverages them for accurate estimation of hourly wages. However, both above-mentioned tools are based on the history of workers executing specific microtasks. Consequently, these tools require several recent records pertaining to each microtask, thereby making it difficult to employ these techniques during selection of previously unseen microtasks.

We have previously proposed TurkScanner (Saito et al., 2019a), a machine learning-based system capable of accurately predicting working times for AMT microtasks despite them being posted for the first time on a platform. In this system, many features were extracted from microtask data (*i.e.,* metadata and HTML elements). Although the tool leverages Turkopticon ratings and worker profiles as auxiliary data, performance of its prediction operation was independent of their availability. Major challenges associated with the use of Turkscanner was caused by its data-driven approach. The first challenge involved collection of reliable datasets for training the TurkScanner model, since there exists no publicly available dataset containing both microtask data and working times. Additionally, there exists no easy method to accurately gauge microtask working times owing to diverse worker behaviors. The second challenge was to define suitable criteria for model optimization and evaluation based on worker perception. This forms the primary objective of this study, as described in the previous section. A discussion of relevant literature pertaining to human perception is included in the next subsection.

2.3. **Subjective Perception Measurement**

Several extant studies focus on investigating the relationship between subjective human perception and objective numerical scales in different domains. As a first in this regard, Weber, in the 19th century, demonstrated the notion of the "just noticeable difference (JND)," which corresponded to the threshold value of a stimulus that humans perceive as a difference. For derivation of his proposed Weber contrast (Fechner et al., 1966), let R denote the strength of a stimulus and ΔR denote the corresponding JND. The value of $\Delta R/R$ always remains constant, regardless of the R value. For example, for an increase in the value of a stimulus from 100 to 110, a corresponding increase from 200 to 220 in the value of the same stimulus is necessary for humans to perceive an identical change. Subsequently, Fechner derived the Weber–Fechner law (Fechner et al., 1966) by integrating the Weber contrast defined by the relationship $E = C \log R$, where E denotes the subjective perception; R denotes the strength of the stimulus; and C is a constant. The above law is applicable to several phenomena, such as weight, sound, and vision (brightness), and it indicates that humans perceive these phenomena approximately logarithmically. Other studies have also applied

---

[7]http://pandacrazy.allbyjohn.com/

the above law in other applications, such as quality of service of communication systems (Reichl et al., 2010) and marketing (Britt, 1975).

In our study, we define the relationship between the subjective perception of workers against microtask completion times and their objective numerical scales. The said relationship has not yet been investigated in extant studies performed in this regard. We believe that development of such a relationship would be a significant contribution to the literature on crowd work and estimation of task-completion times.

3. TRAINING DATA COLLECTION

This section first explains the method employed in the proposed system to calculate the completion time of microtasks from collected data. Subsequently, the design of the web browser extension developed for data collection is described, followed by an overview of the collected dataset.

3.1. Defining and Measuring Working Time

Accurate gauging of microtask completion times is difficult in practice owing to the observed diversity in worker behaviors during task execution (Bederson and Quinn, 2011). Workers are sometimes asked by requesters to temporarily leave their microtask page and browse external websites or use search engines as a part of the task execution. In addition, there exists a possibility that workers may encounter a lapse in concentration and browse other websites not relevant to their assigned task or leave their work desk to take a break. Some workers might even accept multiple microtasks and complete them in succession. Therefore, it is necessary to formalize such behavioral patterns and determine whether the entire time spent must be considered under working hours to make automated data labeling possible.

The proposed system is based on a heuristic labeling method that records working times in three different ways, each possesses its pros and cons. The system ultimately asks workers to select the most reasonable working time applicable. We expected that this method would facilitate accurate labeling of collected data under various operating conditions. The three methods for recording working times can be described as under.

– **TIME_ALL**. In this approach, the working time is automatically recorded by using a program that accounts for the entire duration between the beginning and completion of execution of a microtask assigned to a worker. *Pros*: This is the most reliable method for calculating the working time when a worker starts and completes a task without interruption. *Cons*: All time spent in irrelevant chores (e.g., checking emails, grabbing coffee) is accounted for.
– **TIME_FOCUS**. This approach is similar to TIME_ALL, but only accounts for the time for which the worker was operating on the microtask-page browser tab. *Pros*: This method excludes time spent on task-irrelevant events. *Cons*: Time spent on other tabs related to task task execusion (*e.g.,* survey web pages, Google searches) cannot be properly accounted for.
– **TIME_BTN**. In this approach, workers record the working time themselves, by toggling on-screen buttons at instances when they begin and finish working on the assigned task. *Pros*: This approach can cover all worker behavior patterns, including unexpected ones. *Cons*: The approach is fully dependent on worker operation, thereby it is vulnerable to human error (*e.g.,* careless or spam response).
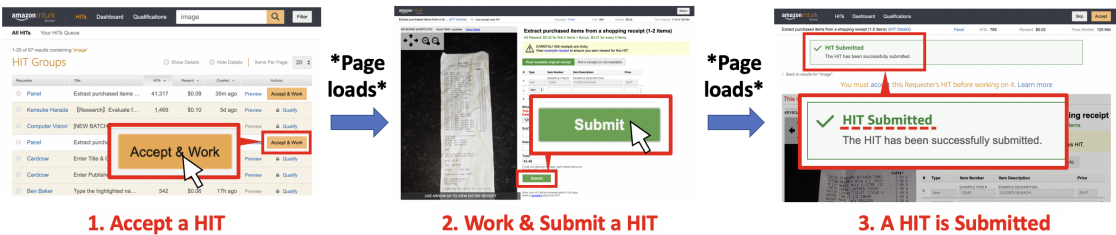
Figure 1. *A procedure of how a worker accepts and completes a HIT in Amazon Mechanical Turk.*

To arrive at a final decision, a fourth working time type, described below, was also introduced.

– **TIME_CUSTOM**. In case none of the three above-mentioned options seem to correctly represent the working time, working hours are to be manually input by workers. *Pros*: It provides workers with a last-resort option to label the correct working time when all other recording methods fail. *Cons*: Errors in worker response may still be present.

The following hypotheses regarding working-time calculation were considered in this study. First, TIME_BTN approach was considered the most dominant choice by workers for determining task completion times, followed by TIME_ALL. This is because workers frequently browse external websites for various reasons and may even leave their work desks halfway through. Thus, they would be able to record moderately accurate working times by simply toggling a button. Although the TIME_ALL approach is the most reliable when workers do not take breaks when executing an assigned task, some experienced workers often accept multiple microtasks in multiple browser tabs (Kaplan et al., 2018), and thus, it was surmised that the TIME_ALL approach would not be as frequently used as TIME_BTN.

3.2.   **Data Collection with a Web Browser Script**

To facilitate microtask data collection with working-time labels, AMT workers were recruited and asked to install our web browser extension to scrape microtask data and send it to our server. See Figure 2 for the whole procedure.

Participant workers were recruited via an AMT survey microtask. When the participants accepted the microtask, they were first asked to provide details of their basic profile (*e.g.,* gender, age, household income, years of experience, and weekly working hours). Subsequently, they were navigated to a web page with a link to install the web extension script [8]. Upon completion of the installation, workers were advised that they may start the process, thereby contributing to data collection for up to 10 days, following which they were allowed to uninstall the software at any time. Based on their contribution to the said data collection, all participating workers were awarded a bonus.

After installation of the browser extension, workers were instructed to work on microtasks (or "HITs", as they are referred to in the AMT environment) per their regular workflows. A HIT is

---

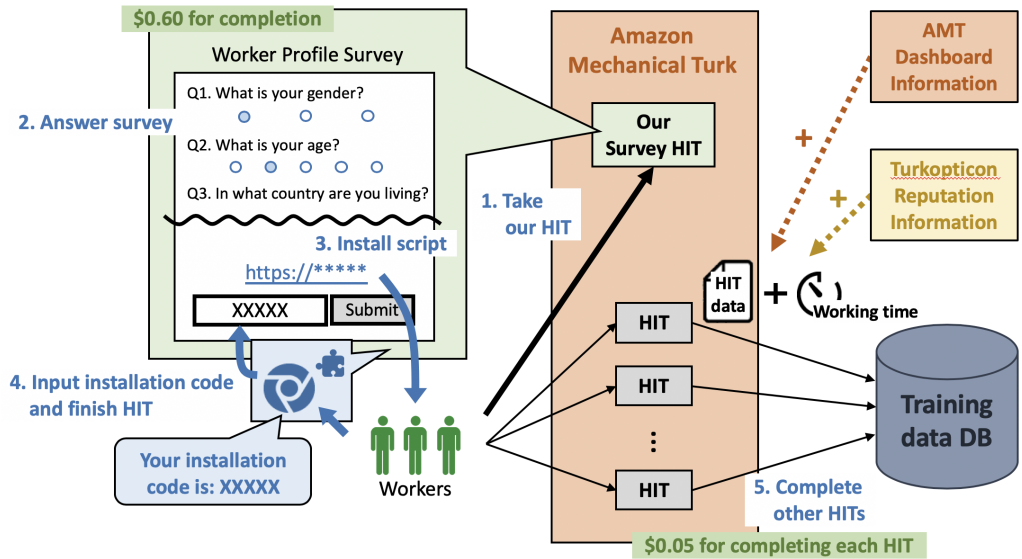[8]https://github.com/shuwakkumacs/hitscraper/

Figure 2. A data collection procedure. Workers first take our survey HITs to install our browser extension as well as to answer questions about their worker profiles for participating our data collection study. Once the browser extension is installed, it collects data of all HITs visited by the workers together with actual working times.

a single web page that appears when it is accepted until it is submitted by a worker (see Figure 1.) Each HIT completion record was collected in accordance with steps (a)–(c) below, and participating workers were awarded a bonus of 5 cents for completing steps (b) and (c).

(a) Background-data scraping: On each HIT page visited by a worker, the installed extension script extracted various microtask-relevant data as input features for the working-time estimation system. These features included information pertaining to the microtask, worker, and requester. Microtask information was extracted from HIT metadata and HTML elements available on every visited HIT page. Worker information contained survey responses regarding basic worker information collected prior to extension script installation, as well as worker profile obtained once per day via an asynchronous request on the worker dashboard page in AMT. We assumed that worker features would represent worker capabilities, such as their skill levels and learning-curve effects (Yelle, 1979), with regard to microtask execution (Rzeszotarski and Kittur, 2011). Requester information included their reputation on microtasks they created, as obtained from Turkopticon via provided APIs.

(b) Manual working time recording: Upon opening an accepted HIT page, workers were instructed to record their working time themselves until the assigned task was completed. The working states of workers (whether active or paused) was recorded by toggling a button rendered at the top of the HIT interface by the web-extension script (refer Figure 3.) The following two features were incorporated to prevent workers from forgetting to record their working state. First, the assigned
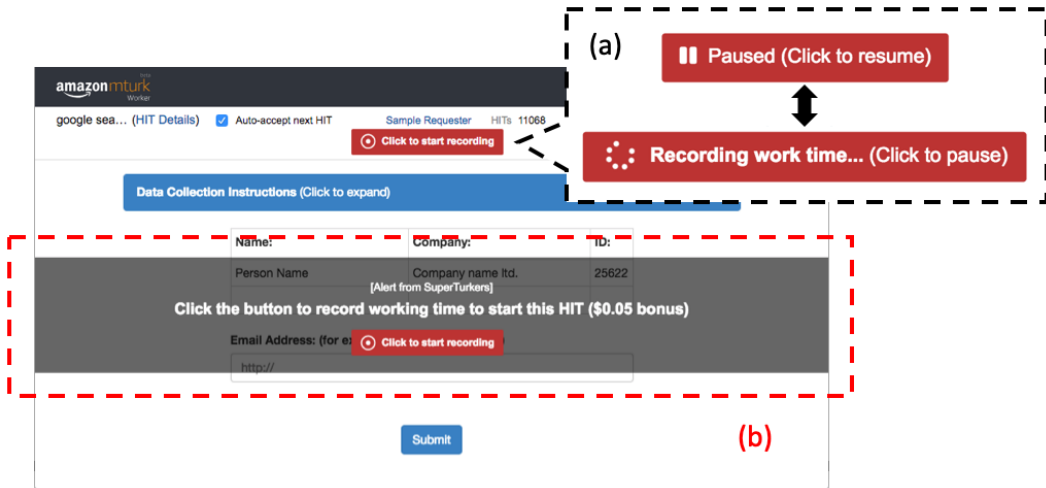
Figure 3. Interface to record TIME_BTN. (a) The button at the top of the HIT page can be toggled to pause/resume recording working time. (b) A black screen is rendered over the HIT at the beginning as a reminder workers to start the timer.

HIT page was overlaid with an alert screen, which workers were supposed to manually dismiss prior to beginning execution of relevant HIT microtasks. Secondly, a red border was rendered around the HIT page while the recording button was activated, so that workers would be easily notified of the status of the recording button. Not more than two buttons were activated simultaneously in multiple tabs, because it was assumed that multitasking on HITs is not practically possible.

(c) **Post-HIT survey:** Upon HIT completion, workers were asked to provide recorded working time as one out of multiple available choices for labeling the HIT submission record with a working time. Upon submission of each HIT, a window popped up asking workers to choose from amongst working times calculated using the TIME_ALL, TIME_FOCUS, TIME_BTN, and TIME_CUSTOM approaches as to what they thought was the most reasonable. In cases where workers felt that none of the first three choices were correct, they selected the last choice (*i.e.,* TIME_CUSTOM) and manually input their working time in the "X minutes and Y seconds" format in a separate text box provided for this purpose. After selection of an appropriate working time, workers could click the "Submit" button to send their answers and dismiss the window.

3.3. **Data Description**

The above described data collection exercise performed for 10 days in late October 2018 yielded 7,303 valid HIT records collected from 83 participating workers. The resulting HIT record dataset comprised 1,587 unique HIT groups (a batch of HIT instances that share the same microtask metadata and interface) created by 977 requesters. On average, participants contributed for 6.5 days (SD = 3.5; Median = 8.1) and worked on 109 HITs (Min = 1; Max = 1958; SD = 238.1; Median = 34).

Figure 4 depicts a histogram of working times corresponding to collected HIT records. The work-
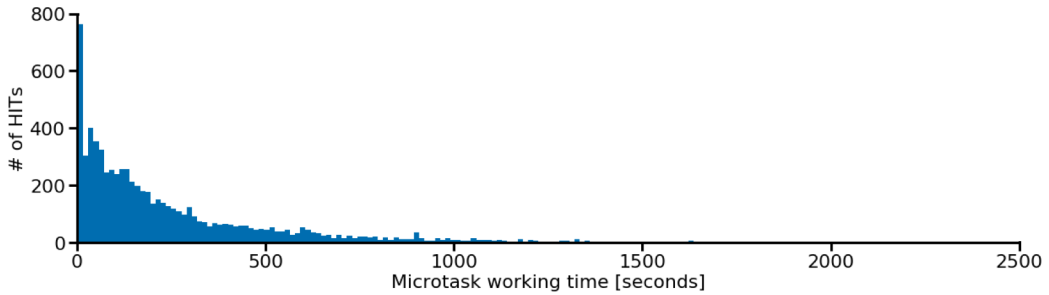
Figure 4. ***Working time distribution of microtasks in the dataset.***

ing time lengths demonstrated a long-tail distribution, wherein majority of submitted HITs were completed within durations less than 2.5 min and large variance ranging from 3 s to more than 1 h (SD = 380.2 s; Median = 148.3 s; Min = 3 s; Max = 4118 s, Mean = 277.9 s). We believe that a microtask working-time collection with such a long-tail distribution is natural considering that most workers in crowd markets are beginners (Hara et al., 2018) that are more likely to accept shorter tasks (Cheng et al., 2015).

As regard working time labels in the collected dataset, the TIME_ALL, TIME_BTN, TIME_FOCUS, and TIME_CUSTOM approaches were chosen for 3,600 (49.3%), 2,461 (33.7%), 745 (10.2%), and 497 (6.8%) HIT records, respectively. This result is in partial agreement with our initial hypotheses in that the TIME_FOCUS and TIME_CUSTOM approaches were chosen less frequently compared to other choices. However, the observed result was also contrary to our expectation, since the TIME_ALL approach was chosen more frequently compared to TIME_CUSTOM by 15.6 points.

The toggle button was clicked at least once in 6,681 (91.5%) HITs, and the timer was paused once, twice, and four in 60, 6, and 3 HITs, respectively, and the same was never stopped in the remaining 6,612 HITs. These results imply that in most cases, the alert screen seemed to be effective in prompting workers to activate the recording button, but the red border shown while the timer was activated did not necessarily work at all times. Among data labeled with TIME_BTN, the observed difference between working times calculated using the TIME_BTN and TIME_ALL approaches measured less than 5 s and 10 s in 75.4% and 86.9% of all recorded cases. This indicates that the TIME_ALL and TIME_BTN approaches recorded nearly the same working time in most cases labeled with TIME_BTN, thereby implying that the button was immediately clicked when the tasks were started and completed without taking a break. However, some workers still considered the TIME_BTN option to be more accurate without a strong reason. However, a rigid evaluation of whether the button was "correctly" used is not possible because the collected dataset did not contain tracking information of workers during HIT execution.

Another kind of analysis that would be interesting is to discuss collected working times per HIT type (*e.g.*, image tagging, writing, etc.) However, we would like to leave it for future work, because there is no clear criteria to categorize HIT types, and such categorization would require a lot of hours, which is currently out of our focus in this paper.

4. TRAINING AND EVALUATION OF PROPOSED WORKING TIME PREDICTION MODEL

This section introduces CrowdSense — an approach to measure worker perception, i.e., whether or not they "accept" the difference (hereinafter referred to as "prediction error") between predicted and actual working times. We define workers' acceptance of a prediction error in the sense that they perceive the prediction error as not being problematic to disturb their workflow.

During model training, our initial concern was to quantify an acceptable accuracy level for working-time prediction. Of course, ideally, a predictive system must not return any error. However, this is not realistic, because there always exist several types of noise that result in prediction of inaccurate working times. Thus, a threshold value for prediction errors must be defined for them to be accepted by workers. For instance, a prediction of $(predicted, actual) = (200s, 250s)$ might be acceptable, whereas that of $(200s, 300s)$ might get rejected. However, when working time have smaller values, both $(20s, 25s)$ and $(20s, 30s)$ might not be problematic but $(20s, 40s)$ might get rejected. Thus, we believed that a minimum prediction error, which workers perceive as unacceptable or problematic must be defined. Such relationships between the objective value of any stimulus and human perception of the same have been investigated in previous literature (Reichl et al., 2010; Britt, 1975), and the difference threshold is often referred to as the "just notifiable difference (JND)" (Fechner et al., 1966; Sher et al., 2017).

During CrowdSense development, we expected following possibilities. First, CrowdSense would facilitate **evaluation** of system performance based on worker acceptability of prediction errors. For example, the overall prediction accuracy can be calculated by checking whether the pair of predicted and actual working times for each tested microtask was below or above JND (*i.e.,* whether the prediction error would be acceptable to or rejected by workers.) Without CrowdSense, we can only discuss the difference between predicted and actual working times in terms of seconds or the percentage error. This does not explain how meaningful a certain prediction would be to workers. Secondly, CrowdSense would contribute to **optimization** of the predictive model, thereby reducing prediction errors that might be considered problematic by workers. The JND defines, for any given working-time duration, the maximum prediction error acceptable to workers, thereby demonstrating worker sensitivity to the prediction error. Realizing this sensitivity would help one prioritize during model training as to which type of prediction error must be eliminated first, thereby facilitating elimination of problematic prediction errors a priori. However, model optimization without CrowdSense would only allow us to calculate training losses in terms of simple differences between actual and predicted working times. This would not always make the model optimum in terms of worker acceptability of prediction errors; for example, calculating simple differences in seconds would result in relatively larger training losses for long-duration microtasks.

4.1. CrowdSense: Survey-based Measurement of Workers' Perception of Working Time

4.1.1. Strategy For Estimating JNDs

For JND estimation, we leveraged the *method of constant stimuli* (Woodworth and Schlosberg, 1954; Kuroda and Hasuo, 2014), considering the human perception in weightlifting as an example. To this end, participants were asked to lift a standard weight followed by a comparison weight.

Subsequently, they were asked to judge whether they detected a difference between the two weights. The standard weight is usually fixed to a certain value (*e.g.,* 100 g), whereas the comparison weight is discretely altered within a certain range (*e.g.,* 105 g, 110 g, 115 g, ..., 150 g). Participants were asked to compare any one weight pair during each comparison trial. After gathering responses from a sufficient number of participants for each pair, a threshold value of the comparison weight was determined to be that for which more than half of all participants perceived a difference in weight. This threshold was, thus, considered JND with regard to the standard weight.

In our study, the method of constant stimuli was employed to determine JNDs for working times of microtasks. To this end, we designed a microtask that asked workers to compare a pair of suggested working times by considering the predicted and actual working times as the standard and comparison values, respectively. During execution of the survey microtask, workers were instructed to assume a situation wherein they utilized a system that erroneously predicts the working time of a given microtask. Being given a random set of predicted and actual working times for an imaginary microtask, workers were asked whether the difference between the times, or prediction error, was acceptable to them or not. After questioning multiple workers, JNDs were determined for each standard value of the predicted working time by calculating the prediction-error threshold which more than half of all workers perceived as acceptable.

Two different metrics were considered in this study depending on whether the residual (or working-time prediction error) calculated as [actual time]-[predicted time] possessed a positive or negative value. Implications of a positive residual (*i.e., predicted < actual*) are easy to understand; the larger the residual, the more annoyed workers would be, because the system overestimates the benefit, which directly reduces worker earnings. Because this problem occurs when workers actually work on a microtask, we asked workers to imagine that they actually accepted and completed a certain microtask, and then asked if they felt the prediction error was problematic or not.

In contrast, a negative residual value (*i.e., predicted > actual*) necessitates use of a slightly different setting in the survey, since the obtained result is not intuitive. When a worker completes a microtask, the predicted working time of which exceeds the actual working time, the worker can earn more than expected. For this reason, survey with negative residuals would end up collecting more "acceptable" responses for nearly all comparison pairs. However, a system that always predicts working times shorter compared to the actual is not considered a good system. The more the microtasks are undervalued, the more often workers miss opportunities to find lucrative microtasks. To make workers understand the negative impact of this situation on their potential earnings, the survey instruction was slightly altered; workers were directed to assume a microtask that they decided *not* to do, and that they subsequently knew another worker had completed it faster than predicted, owing to existence of prediction error in the system. Workers were then asked if the incurred error was acceptable to them. By setting the survey question in this way, we expected accurate JND determination in cases involving negative residuals.

4.1.2. Microtask Survey Design

Figure 5 depicts microtask interface designs considered in this study. In each pair, workers were shown values of "predicted" and "actual" working times on the left and right, respectively. Whether the calculated prediction error was acceptable to workers or not was recorded by asking them to click an appropriate response button. The next comparison pair queue was displayed to them as

Figure 5.** **Microtask survey interface to evaluate prediction error (of a positive residual) by comparing a predicted working time (left) and the actual working time (right). To evaluate a negative residual, we changed the sentence to "you decided NOT to work" for predicted time and to "someone else ended up spending" for actual time, and we made the actual time shorter than the predicted time.

soon as they finished answering the previous question. This sequence continued until the last pair. The microtask was submitted after workers responded to a simple survey that also recorded their comments and feedbacks, if any. Each participant was paid $1.50 USD (expected ∼$10/hr) for evaluating up to 100 different comparison pairs (or less when no more pairs were listed in the queue) yielding positive or negative residuals. Each participant was allowed to take surveys for both residual types.

In this study, worker judgments were intentionally recorded by providing with working times exclusively, and no other information was provided, such as the price or type of microtask assigned. It seems to be a natural argument that microtask prices and types must also be provided to survey takers. This is because workers always refer to this information when judging whether a microtask is worth doing, and knowledge of the same might consequently change their response. Although we believe this to be true, it must be noted that there exist both pros and cons in making this information available to workers. While access to this information would provide greater insight into how workers practically evaluate the working time in their daily routines, it is also true that workers often set their own hourly earning goals and microtask-type preferences. Thus, their responses may easily get biased, and results obtained would solely demonstrate demographic distributions of

Table 1. List of parameter values used for generating comparison pairs
$(predicted[s], actual[s]) = (p_i, a_{ij})$. **_For positive residuals, there exist_** $\sum \mathbb{N}_{pos} = 641$ **_pairs,_**
wherein $a_{ij} = p_i + jd_i (1 \leq j \leq n_i, d_i \in \mathbb{D}_{pos}, n_i \in \mathbb{N}_{pos})$. **_For negative residuals, there exist_**
$\sum \mathbb{N}_{neg} = 277$ **_pairs, wherein_** $a_{ij} = p_i - jd_i (1 \leq j \leq n_i, d_i \in \mathbb{D}_{neg}, n_i \in \mathbb{N}_{neg})$. **_Frequencies of_** p_i, d_i,
and n_i **_were determined based on arbitrary choices made by authors based on the policy of i)_**
successfully determining JND thresholds for each p_i, **_and ii) sampling adequate data whilst_**
consider as few plots as possible to determine JNDs.

i	$p_i \in \mathbb{P}$	$d_i \in \mathbb{D}_{pos}$	$n_i \in \mathbb{N}_{pos}$	$d_i \in \mathbb{D}_{neg}$	$n_i \in \mathbb{N}_{neg}$
0	5	5	29	–	–
1	10	10	22	5	1
2	30	10	22	5	5
3	45	10	22	5	8
4	60	10	22	10	5
5	120	15	22	15	7
6	180	10	45	15	11
7	240	10	45	20	11
8	300	10	45	20	14
9	450	20	22	30	12
10	600	20	45	30	15
11	900	20	45	30	15
12	1200	20	45	30	20
13	1500	30	45	30	25
14	1800	30	45	30	30
15	2250	60	30	60	18
16	2700	60	30	60	23
17	3150	60	30	60	27
18	3600	60	30	60	30

hourly-wage goals and microtask preferences of participants, which are not of interest in this study. It is understood that worker responses vary based on their experience and circumstances. The sole intent of this study was to address this variance by collecting multiple responses from different workers for each sample and averaging them.

In this study, we prepared 641 and 277 comparison pairs with positive and negative residuals, respectively. Subsequently, 19 different time lengths (in seconds) were considered for use as predicted working times varying in the range of 5 s to 1 h, each of which was denoted as $p_i \in \mathbb{P}$. For each p_i, corresponding actual working times were set, denoted by $a_{ij} = p_i + jd_i (1 \leq j \leq n_i, d_i \in \mathbb{D}_{pos}, n_i \in \mathbb{N}_{pos})$ and $a_{ij} = p_i - jd_i (1 \leq j \leq n_i, d_i \in \mathbb{D}_{neg}, n_i \in \mathbb{N}_{neg})$ for positive and negative residuals, respectively, where d_i denotes interval of the difference between predicted and actual working times for each pair. The value of d_i increases upon each iteration of j, and n_i denotes the number of sampled a_{ij} for each p_i. See Table 1 for the full list of p_i, d_i, and n_i values.

4.2. Defining An Evaluation Function Based on Collected Results

In this study, 91,060 worker responses were collected as comparison-pair data samples. The number of participating workers was 875, of which 131 responded to survey questions pertaining to both positive and negative residuals. With regard to positive residuals, evaluations were performed using 60,760 responses provided by 660 unique workers. On average, each comparison pair was evaluated by 95.4 unique workers (median = 97; SD = 8.9; minimum = 62; maximum = 118). For negative residuals, 30,300 comparison responses provided by 346 unique workers were collected. On average, each pair was evaluated by 109.4 unique workers (median = 110; SD = 5.2; minimum = 95; maximum = 123).

Similarly to what we discussed in Section 3.3, it would have been interesting to analyze the variance of workers' tolerance of prediction error across different microtask types, by specifying what type of microtasks the participants were evaluating. Although we agree that microtask types would make some difference to workers' tolerance, we were not able to conduct such analysis in this paper because *i)* there would be a large number of microtask type sub-categories (*e.g.*, image classification and bounding box drawing would vary workers' tolerance, while they belong to the same vision-related microtasks) and *ii)* the results would vary across workers by their preferences and expertise.

Once data collection was completed, an evaluation function was derived based on CrowdSense results. It was expected that the said CrowdSense-based evaluation function would enable us to quantitatively determine the possibility of predicting results with sufficient accuracy so as to be acceptable to workers. This was previously impossible with the exclusive use of objective values of the working time. To this end, we calculated a percentage of "acceptable" votes for each comparison pair to obtain the maximum acceptable prediction error for each p_i, denoted by $e_i \in E$. Using the constant stimuli approach, we defined acceptable prediction errors as those for which the response from 50% or more of all participating workers was recorded as "acceptable." Subsequently, a series of e_i values were curve-fitted to derive the function $e = f(p)$, where p denotes the predicted working time, and e denotes the residual corresponding to the maximum acceptable prediction error.

See Figure 6 for comparison-pair survey results. For the data samples obtained for positive and negative residuals, the least-squares method was employed to fit a function curve in accordance with the relation:

$$e = f(p) = \alpha \log(p + \beta) + \gamma \tag{1}$$

α, β, and γ are coefficients, calculated under the constraint that the curve function is fitted through all the calculated e_i (the maximal acceptable prediction error for each p_i, shown by black plots.) Using our survey results, values of the coefficients were determined to be $\alpha = 164.3$, $\beta = 173.1$, and $\gamma = -780.5$ for the positive-residual function ($f_{pos}(p)$); corresponding coefficient values for the negative-residual function ($f_{neg}(p)$) were $\alpha = -289.6$, $\beta = 358.5$, and $\gamma = 1703.1$. The resulting log-like curve demonstrated an interesting trend indicating that most participating workers were forgiving of large errors with regard to predicted working times for small microtasks, whereas they were likely to accept small relative errors (not exceeding ~500 s) for large microtasks up to an hour. In addition, participants were observed to be as tolerant to errors concerning negative-residual comparison pairs as they were to those pertaining to positive-residual ones. However, they demonstrated greater tolerance for prediction errors pertaining to longer predicted working times.

Based on Equation (1), we can then derive a function that calculates a system "accuracy", meaning
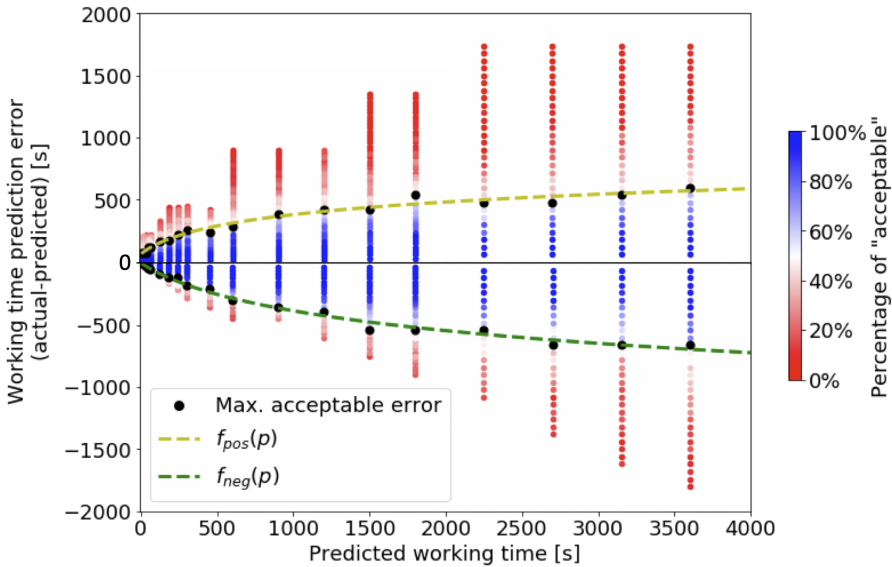
Figure 6. *Survey results for all a_{ij} (blue, white, or red plots), maximal acceptable prediction error e_i in each p_i (black plots), and a curve fitted to the series of $e_i(=\mathbb{E})$ (dashed curve), for the cases of positive and negative residuals respectively. $\mathbb{E}$ was fitted by a log curve (i.e., $f_{pos}(p), f_{neg}(p) = \alpha \log(p + \beta) + \gamma$, where α, β, and γ are constants).*

how frequent the system is able to predicting working times of microtasks within errors that workers can tolerate. Here we let a set of evaluation functions $f = \{f_{pos}(p), f_{neg}(p)\}$ and an actual working time of a microtask a. If $|a - p| \leq |f(p)|$, the prediction error of the microtask can be regarded as correct as workers would think is not problematic. Thus we define a function for calculating the system's prediction accuracy, that counts the number of microtasks with acceptable working time prediction error among all n tested microtasks ($T = \{t_0, t_1, ..., t_k, ..., t_n\}$) as follows:

$$\text{Prediction Accuracy [\%]} = \frac{\sum_{k=0}^{n}[|a_k - p_k| \leq |f(p_k)|]}{n} \times 100 \qquad (2)$$

4.3.   **Defining An Objective Function**

We designed objective functions for model optimization based on the two CrowdSense-based evaluation functions. The objective functions were derived with the intention to facilitate calculation of reasonable losses across different working-time ranges based on subjective worker perception. In contrast, as described in Section 1, prediction errors calculated with an objective working-time length would exaggerate losses for large microtasks with longer completion times.

The objective functions were designed by defining the "psychological amount" of working time, as depicted in Figure 7. This definition is based on that Weber–Fechner law (Fechner et al., 1966) was used for defining "psychological amount" of any stimuli derived from Weber's law incorporating
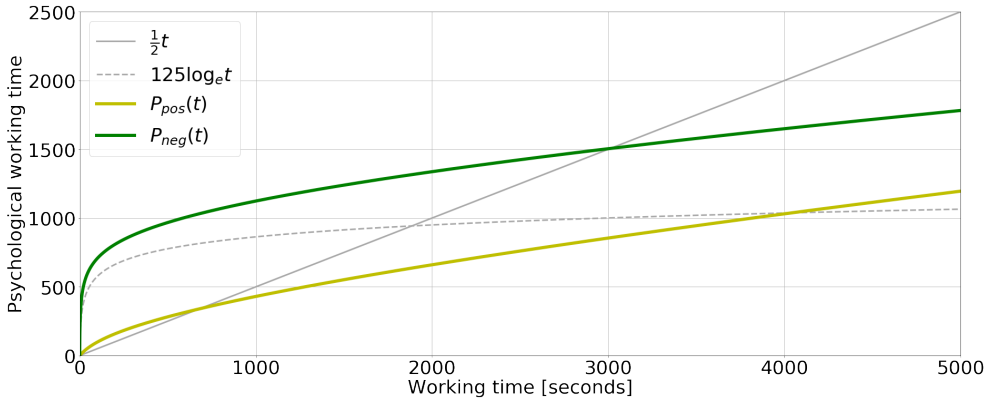
Figure 7. Psychological amount of working time. The linear function and the log function in gray color are visualized as baseline functions for reference. Offsets ensure that the graph of each function contains the point $(x,y) = (0,0)$***.***

JNDs. Reciprocals of evaluation functions f were considered to represent workers' sensitivity to the prediction error. Accordingly, it was expected that a function obtained by integrating the reciprocal of f would represent a new working-time scale that can also be referred to as the psychological working time (denoted by P):

$$
\begin{aligned}
P &= \{P_{pos}(t), P_{neg}(t)\} \\
&= \left\{ \int \frac{K}{f_{pos}(t)} dt, - \int \frac{K}{f_{neg}(t)} dt \right\} \\
&= \left\{ \frac{e^{-\frac{\gamma}{\alpha}} K \cdot \mathrm{Ei}(\log(t+\beta) + \frac{\gamma}{\alpha})}{\alpha}, - \frac{e^{-\frac{\gamma}{\alpha}} K \cdot \mathrm{Ei}(\log(t+\beta) + \frac{\gamma}{\alpha})}{\alpha} \right\}
\end{aligned}
\tag{3}
$$

where K is a constant and Ei is the exponential integral.

Appropriate prediction losses can simply be defined as the difference between psychological amounts of the predicted and actual working times estimated by using appropriate CrowdSense functions. That is, the psychological amount-based prediction loss can be expressed as

$$
\mathrm{Loss} = \begin{cases} P_{pos}(a) - P_{pos}(p) & (a \geq p) \\ P_{neg}(p) - P_{neg}(a) & (a < p) \end{cases}
\tag{4}
$$

where a denotes the actual working time, and p denotes the predicted working time, both measured in seconds.

For instance, let us consider calculation of losses on prediction errors in cases where $(predicted, actual) = (30s, 60s)$ and $(1030s, 1060s)$. Because residuals are positive in both cases, the resulting losses can be calculated as $P_{pos}(60) - P_{pos}(30) \approx 25.2$ and $P_{pos}(1060) - P_{pos}(1030) \approx$

7.8, respectively. This demonstrates that penalties can be appropriately calculated, as we mentioned previously that a loss would be smaller when its calculation is based on psychological working time than when it is based on seconds where the working time is longer.

5.   **EXPERIMENT**

In this section demonstrates how CrowdSense facilitates *i)* evaluation of the proposed system for predicting microtask working times based on workers' perception leading to acceptance or rejection of prediction errors, and *ii)* optimization of the proposed model for more accurate working-time prediction. Defining the "overall accuracy" of a system in terms of the psychological likelihood of workers to accept predicted working times, although with some errors, has not been considered in previous studies performed in this domain (Saito et al., 2019a). We hypothesized on the model optimization that CrowdSense would facilitate realization of an all-the-more accurate working-time prediction system capable of operating across different working-time scales — CrowdSense can define penalties on prediction errors based on workers' tolerance across different working-time lengths, thereby allowing the predictive model to optimize itself for minimizing the likelihood of "problematic" error prediction regardless of the microtask duration. This study demonstrates the above hypothesis to be true via comparison against baseline methods that define penalties based solely on working-time predictions in seconds or log-seconds.

5.1.   **Settings**

Evaluation Method and Criteria: Cross validation was performed with collected datasets described in Section 3. The training and test sets were split in a "task group-open" splitting manner; no microtask from the same group in either the training or test sets was included in the other set. In this study, the CrowdSense-based "system accuracy" was set as the evaluation criterion. For each compared optimization method, prediction accuracy was calculated using Equation (2).

Prediction Algorithm: We compared two algorithms to perform regression with Gradient Boosting Decision Tree (GBDT) (Friedman, 2001) and neural network (NN). Regardless of the method employed, a feature vector with 101 dimensions was considered as input, and a scalar value of the predicted working time in s was obtained as the output. In GBDT-based models, LightGBM (Ke et al., 2017) and hand-tuned hyper parameters were used for each compared method, such that its highest overall accuracy was realized upon convergence of its training loss. Values of the maximum tree depth lied in the range of 3âĂŞ4, and the number of trees (= iterations) ranged between 350 and 450. On the other hand, PyTorch (Ketkar, 2017) was used for building NN-based models that comprised 30- and 10-dimensional hidden layers. Additionally, Rectified Linear Unit (ReLU) (Nair and Hinton, 2010) was employed as the activation function for each hidden layer. Similar to GBDT, hyper parameters for each NN-based model were also hand-tuned. Batch training (batch size = 32) was performed for ∼1000 epochs to obtain the highest overall accuracy upon convergence of training losses.

Input Features: Using collected data, microtask-, requester-, and worker-relevant features were extracted. See Table 2 containing the comprehensive features list.

– *Microtask features* correspond to the most dominant representation. They include HIT metadata (basic profile of a HIT instance, e.g., time limit (in seconds), price, and HIT batch size) as well

Table 2. List of input features parsed from the collected data. The features consist of three categories and eight sub-categories. The parenthesized numbers in bold text represent the feature dimension sizes.

HIT (78) – HIT-relevant information		
META	**(5)**	**HIT metadata set by requesters:** time limit, reward, # of HITs in a batch, template type, and HTML text length.
URL	**(7)**	**URL counts in HTML source code:** anchor links, in-text links, image links, audio links, video links, Qualtrics[9] survey links, and all links.
INP	**(36)**	**Input-tag counts and percentage :** text, submit, radio, checkbox, select, hidden, text area, number, and other 8 input types.
KW	**(30)**	**Keyword occurrence in either HIT title, description, or page:** "summarize", "survey", "instructions", "opinion", "description", "describe", "read", "click", "audio", "video", and 20 other keywords.
WKR (20) – Worker-relevant information		
PRFL	**(8)**	**Worker-profile information collected in pre-surveys:** age, gender, education level, employment status, household income, years of work experience, weekly working hours, and estimated amount of earnings per hour.
EXT	**(8)**	**Installed AMT-relevant extension tools:** CrowdWorkers[10], Distill[11], Tampermonkey[12], OpenTurk[13], MTurk Suite[14], TurkOpticon, Page Monitor[15], and Auto Refresh[16].
HIST	**(4)**	**Worker dashboard information:** # of approved HITs, approval rate, total earnings, and # of HIT submissions in a HIT group
REQ (3) – Requester reputation information		
TO	**(3)**	**Turkopticon:** average of 5-point scale ratings relevant to requester evaluation, for their generosity/fairness and # of reviews.

as keyword occurrences, URL counts, and input-tag counts extracted from the HTML source code of the microtask. We expected that these features would contribute to the proposed model's learning of microtask contents that affect working time and the extent to which they do so.

– *Requester features* represent the reputation of requesters posted in Turkopticon (Irani and Silberman, 2013) by AMT workers, which can be obtained via API calls. This information is considered to indicate how appropriate working times might be assumed by the requester in consideration of microtask information provided to workers. In case the reputation a particular requester could not be obtained from Turkopticon, average values for each criterion were calculated and used as requester-reputation features.

– *Worker features* include worker-profile information provided by workers in the beginning of the data-collection exercise along with a list of AMT worker tools installed in their browser and worker dashboard information. By leveraging such features, the proposed model can possibly consider worker capabilities for crowd work, thereby fine-tuning the predicted working time of a

Table 3. System performance evaluation results based on worker error acceptance. a) Overall accuracy across all tested microtasks; b) Average accuracy for microtasks of which working time was shorter than 510 s (i.e., the first four working time categories) or longer (i.e., the last five working time categories.)

	Accuracy [%]		
	(a)	**(b)**	
	All	− 510 s	510 s −
GBDT_raw	65.3	69.9	**35.2**
GBDT_log	70.4	**79.8**	9.7
GBDT_CrowdSense	**73.6**	79.1	31.0
NN_raw	60.5	64.4	27.6
NN_log	63.0	**71.3**	12.0
NN_CrowdSense	**63.5**	65.3	**39.0**

microtask.

Objective Function: For both GBDT and NN models, we compared results obtained using three objective functions that define training losses by calculating the sum of mean squared errors (MSE) of prediction errors corresponding to *raw-*, *log-*, and *CrowdSense*-scaled working times. The prediction error calculated by using the *raw*-scale working time represents a simple difference in seconds between predicted and actual working times. For example, prediction errors for cases $(predicted, actual) = (60s, 30s)$ and $(300s, 450s)$ equal 30 and 150, respectively. The *Log*-scaled working time calculates prediction errors based on the difference between logged predicted and actual working times. In other words, prediction errors for the two above-mentioned cases are calculated as $|\log 60 - \log 30| \approx 0.69$ and $|\log 300 - \log 450| \approx 0.41$, respectively. Prediction errors corresponding to *CrowdSense*-scaled working times were calculated by using Equation (4). Values of prediction errors for the two above-mentioned cases were calculated to be $P_{neg}(60) - P_{neg}(30) \approx 90.02$ and $P_{pos}(450) - P_{pos}(300) \approx 58.95$, respectively.

5.2.   Results

Prediction accuracy by methods: Evaluation results obtained in this study reveal that use of the CrowdSense approach contributes to the determination of both the overall accuracy based on subjective worker perception and best prediction score. Table 3a lists overall accuracy values obtained for all compared methods, calculated by using Equation (2). As can be realized, in most cases, GBDT-based models demonstrate higher scores compared to NN-based models. The accuracy value of 73.6% was obtained for the GBDT_CrowdSense model. This implies that prediction results for 73.6% of all tested microtasks were sufficiently accurate to be considered acceptable by workers. Scores obtained for other GBDT-based methods were 69.9% (GBDT_log) and 65.3% (GBDT_raw). Results obtained for all NN-based models lied in the range of 60–63%, which are obviously lower compared to those obtained for GBDT-based models.

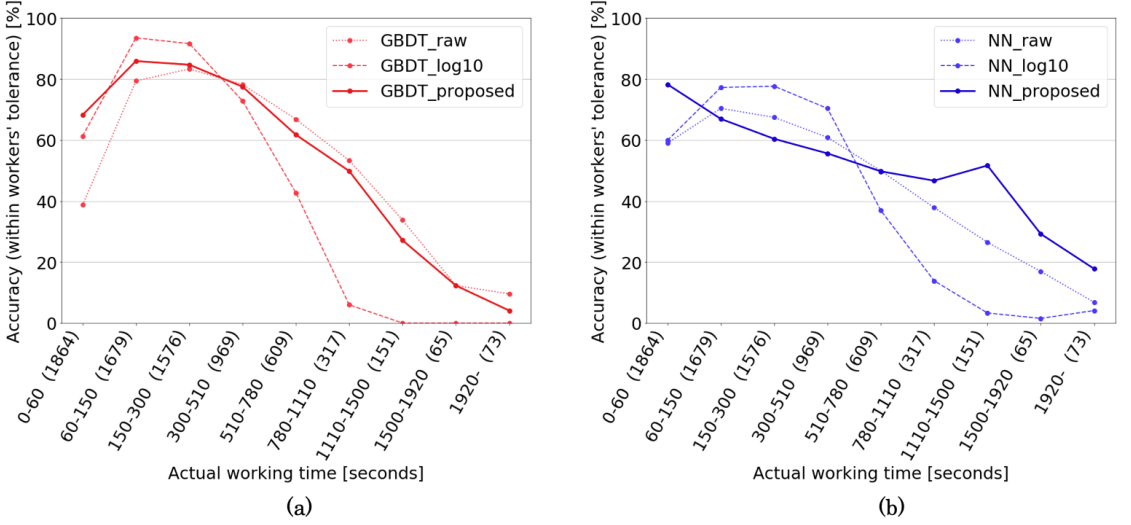It should be noted that our intention is not to emphasize that CrowdSense-based optimization scored

Figure 8. *System performance comparison by working time categories, a) for GBDT and b) for a neural network. In the parenthesis after each actual working time category, the number of tested microtask data whose actual working time is within the range is shown. The working time categories are split based on the evaluation function for positive residual errors, but the accuracy includes both positive and negative errors.*

the best accuracy, but rather to discuss the difference of the accuracy between the CrowdSense-based and objective value-based optimization methods. Since the objective functions and the evaluation functions are both based on the same criteria, it is not very surprising that CrowdSense-based optimization contributed to the best accuracy. However, the prediction accuracy was relatively low when the model is optimized by seconds or log-seconds, which indicates that there is a gap between the workers' perception and objective values of working time. This implies that, without Crowd-Sense, working time prediction would more likely result in undesirable consequences to workers such as making a worker tool that predicts working times of microtasks intuitively less useful.

Prediction accuracy per working time length: Figure 8 illustrates proposed system performance observed for each working-time category. We first created working time categories such that each of them represents a time range that workers would accept as the prediction error (Saito et al., 2019b). The regions were defined as $\mathbb{R} = \{r^{(0)}, r^{(1)}, ..., r^{(N)}\} = \{[r_{min}^{(0)}, r_{max}^{(0)}), [r_{min}^{(1)}, r_{max}^{(1)}), ..., [r_{min}^{(N)}, r_{max}^{(N)}]\}$ where $r_{min}^{(k)} = r_{max}^{(k-1)}$, $r_{max}^{(k)} = r_{min}^{(k)} + f(r_{min}^{(k)}) - ((r_{min}^{(k)} + f(r_{min}^{(k)})) \mod R_{floor})$, $r_{min}^{(0)} = 1$, $r_{max}^{(N)} = \infty$, and $R_{floor} \geq 1$. Figure 9 depicts a schematic illustrating the above region calculation. In this study, $R_{floor} = 30[s]$ and $N = 9$.

For all comparison methods considered in this study, there was a similar trend that accuracy was higher where working time was shorter. The highest accuracy of $\sim$80% was observed for tested microtasks with under 510 s of completion time. For work time exceeding this value, the accuracy

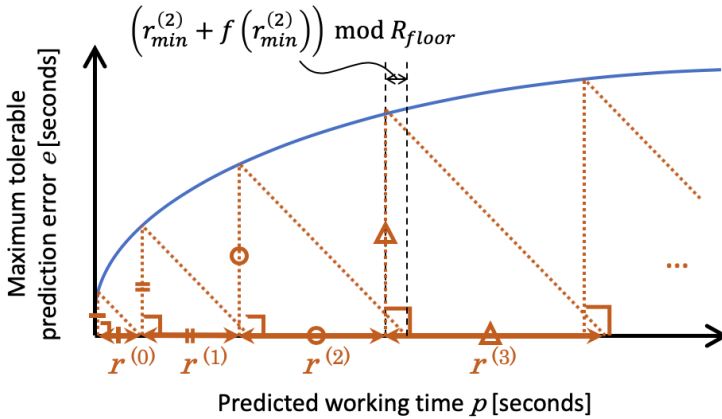$$\left(r_{min}^{(2)} + f\left(r_{min}^{(2)}\right)\right) \bmod R_{floor}$$

Figure 9. Strategy to define ranges based on the evaluation derived from CrowdSense.

was observed to deteriorate.

Differences were also observed between trends pertaining to the two baseline optimization methods — log10 and raw working-time-scale-based — when employing both GBDT- and NN-based models. Model optimization employing the log10 working-time scale demonstrated realization of the highest peaks between 60 s and 300 s, and its accuracy became extremely low with increase in working time beyond 510 s. In contrast, the raw-scale-based optimization demonstrated lower scores corresponding to shorter working times. However, it was found to predict longer working times more accurately. These differences support our hypothesis that the log-scale-based optimization would exaggerate error penalties for shorter microtasks whilst ignore those for longer tasks, and that the raw-scale-based would operate vice-versa. This makes overfitting of the model possible towards the exaggerated side.

However, contrary to the above discussion, the CrowdSense-based optimization was observed to be rather successful in leveraging attributes of both baseline methods. For a simple analysis of the difference between working-time lengths, the working time was divided into two groups comprising four and five regions, respectively, by means of a threshold value of 510 s. Average accuracies for these groups are separately listed in Table 3b. In both groups, the use of CrowdSense-based optimization demonstrated the highest accuracies. When employing the GBDT-based regression model, scores obtained using the CrowdSense-based approach demonstrated a difference of -0.7 points for ≤ 510 s and -4.2 points for > 510 s. Corresponding differences observed when employing NN-based regression equaled -9.9 and -27.9 points, respectively. When employing the NN-based model, CrowdSense demonstrated better accuracy compared to the raw working-time-based optimization for the former category, and the best performance for the latter catergory. This clearly demonstrates that use of the CrowdSense-based approach contributes towards reducing biases across the entire working-time range considered for baseline-optimization methods.

Feature importance: The important features evenly belonged to all the feature categories of microtask-, worker-, and requester-relevant features. See Figure 10 for the ranking list of feature

importance. First, the top microtask features included HIT meta data such as reward (1st), time limit duration (3rd), and HTML text length (7th). This is not very surprising since these features are considered as features that would directly affect HIT working time. On the other hand, the top worker-relevant features mainly represented worker experiences such as weekly working hours (4th), the number of approved HITs (5th), the number of total submissions in the same HIT group (6th), and worker total earnings (8th). These features are thought to be effective to adjust (although still roughly) estimated working time based on how much workers are good at working on micro-tasks. Also, the top requester-relevant features (i.e., Turkopticon ratings) were generosity (2nd), the number of reviews (9th), and fairness (12th). While the aforementioned HIT meta data are solely parameters that requesters can change arbitrarily, the requester-relevant features would enable it to control working time estimation by considering their reliability. Other subsequent top features were keywords, URLs, and input tags contained in HITs: for instance, "minute" / "click" / "describe" as keywords, the total number of URLs in the page / URLs that navigate to other pages as URLs, and textarea / button as input tags.
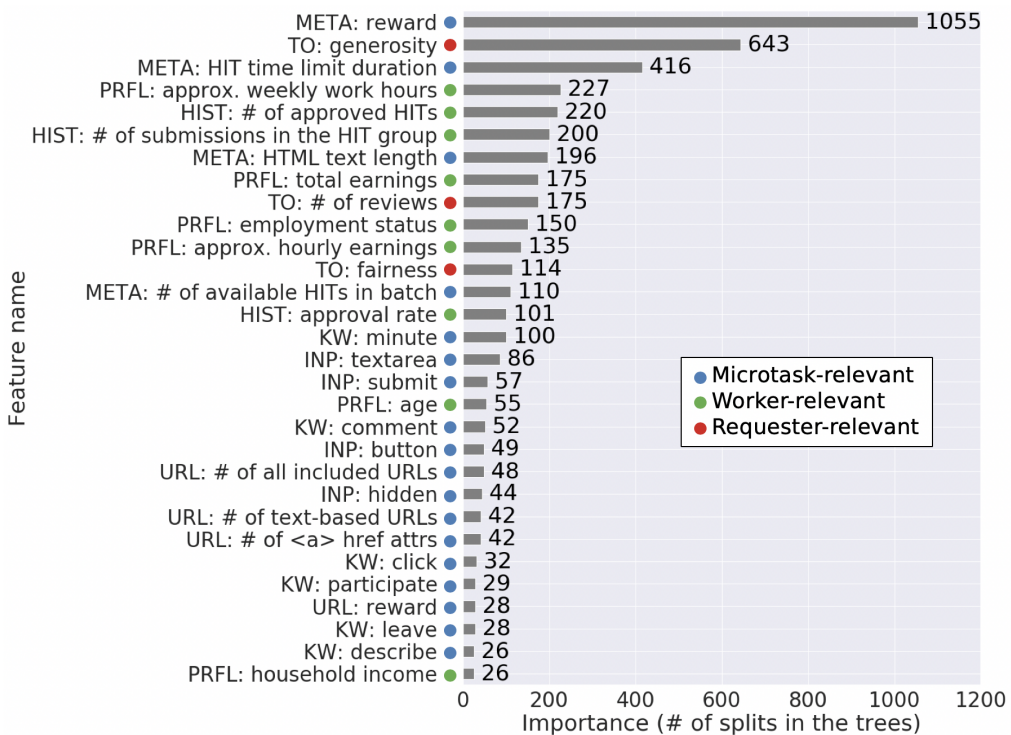

Figure 10.** **The top 30 important features for working time prediction. The importance values were calculated from GBDT_CrowdSense model with a split-based measure (by counting numbers of times the feature was used in the model.)

6.   **LIMITATIONS AND FUTURE WORK**

This study explored development of a system for accurately predicting the working time of micro-tasks by leveraging the subjective perception of workers with regard to the working time. This also resulted in reasonable optimization and evaluation of the underlying prediction model. Limitations of the proposed system and possible future directions are discussed below.

Respecting objective value-based evaluation: As we showed in the experimental results, objective value-based model optimization give biases for penalties of prediction errors across different time scales. By seconds-based optimization, the system was capable of more accurate time prediction for long microtasks but not for short microtasks, and vice-versa by log-seconds-based optimization. Our experimental results demonstrated that our workers' tolerance-based method would be very useful to build a new optimization criteria that takes only the good points of the both objective value-based criteria. In this sense, we believe the workers' tolerance-based model optimization is totally fair for workers; we had no intention of exploiting workers' feelings, but rather our results just showed that it was capable of predicting working times more accurately by solving the problem that the objective value-based optimization had.

However, we should also keep in mind objective meanings of time. Our evaluation function regards each working time prediction as "correct" if workers could tolerate its error regardless of its objective value. Therefore it should be further evaluated among the correct predictions on how much objective error they actually contained. In this paper, supporting workers' decision making — by considering lost time caused by the actual amount of time difference — is the problem of risk management strategy design for workers and task scheduling, which could be another research topic and is out of our scope.

Applicability of workers' tolerance to the real-world usage: In the proposed approach for defining CrowdSense, we posted microtasks that asked workers to analyze their daily crowd-work environment and express how they felt about prediction errors pertaining to simulated microtasks. Although our assumption regarding the simulated environment was sufficient for determining worker perception, it cannot be guaranteed that results obtained by using this approach would exactly match with worker opinions in real-world scenarios. For example, values of parameters, such as α, β, γ, and K, in Equation (1) can be different, or worker perception can possibly be more or less diverse compared to results obtained in this study. Thus, further investigation needs to be performed to determine these differences. However, collection of real-world data is very difficult owing to following reasons: *i)* there is no control on the experimenter side with regard to which pairs of predicted/actual working times must be used for questioning workers; this makes it difficult to sample enough data for each pair; *ii)* workers need to use to a certain kind of working-time prediction system to check predicted working times and complete a microtask to record the actual working time; this is simply too much work to be done for collection of a single sample; *iii)* there are other factors, such as requester preferences or microtask content, that add noise to or bias data, thereby making CrowdSense less generalized. Because of these reasons, a more detailed study design is necessary to collect real-world data for CrowdSense execution.

Pursuing a better prediction accuracy: The highest prediction accuracy observed in this study equaled ~73%. We do not conclude this as the "real" upper limit of performance of our proposed working-time prediction approach, and suggest the following means to enhance the said perfor-

mance. *i)* further feature engineering would contribute to attainment of higher accuracies. In addition to input features used in this study, more meaningful data, such as media content and dynamic elements rendered using JavaScript, can be extracted from microtasks. Other techniques, such as microtask category classification, based on natural language processing can also be employed; *ii)* As mentioned in Section 3.3, collected data demonstrates a long-tail distribution of working-time labels wherein most microtasks demonstrated short working times. Such bias in a dataset causes the trained prediction model to be over-optimized for microtasks with shorter working times. This issue can be addressed via collection of a larger dataset, which specifically aims at obtaining microtasks with longer working times by setting a higher bonus for workers accepting longer microtasks.

7. CONCLUSIONS

This paper presented an approach to predict microtask working times based on CrowdSense — a technique to measure worker perception towards working-time prediction errors for model optimization and evaluation. The motivation behind this study was to help crowd workers estimate how long it would take them to complete a given microtask, thereby facilitating their search for more lucrative microtasks.

The proposed method first addressed the difficulty encountered in defining and gauging the "working time" and collecting associated microtask submission data. Next, the paper presents the CrowdSense approach to quantify the subjective perception of workers towards errors in working-time prediction. This facilitates both optimization and evaluation of the model in terms of how workers perceive prediction results, which was previously not possible owing to exclusive use of objective values of the working time in seconds. Experimental results obtained in this study demonstrate that the proposed working-time prediction system was capable of predicting microtask working times with due worker acceptance in $\sim$73% of all tested cases. The results also revealed that use of the CrowdSense-based model optimization enhances the prediction accuracy across a broad range of working times, which was not possible by using extant baseline approaches.

8. ACKNOWLEDGEMENTS

9. REFERENCES

Alsayasneh, M, Amer-Yahia, S, Gaussier, E, Leroy, V, Pilourdault, J, Borromeo, R. M, Toyama, M, and Renders, J.-M. (2017). Personalized and diverse task composition in crowdsourcing. *IEEE Transactions on Knowledge and Data Engineering* 30, 1 (2017), 128–141.

Barowy, D. W, Berger, E. D, Goldstein, D. G, and Suri, S. (2017). Voxpl: Programming with the wisdom of the crowd. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 2347–2358.

Bederson, B. B and Quinn, A. J. (2011). Web workers unite! addressing challenges of online laborers. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*. ACM, 97–106.

Berg, J. (2015). Income security in the on-demand economy: Findings and policy lessons from a survey of crowdworkers. *Comp. Lab. L. & Pol'y J.* 37 (2015), 543.

Brewer, R, Morris, M. R, and Piper, A. M. (2016). Why would anybody do this?: Understanding older adults' motivations and challenges in crowd work. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2246–2257.

Britt, S. H. (1975). How Weber's law can be applied to marketing. *Business Horizons* 18, 1 (1975), 21–29.

Callison-Burch, C. (2014). Crowd-workers: Aggregating information across turkers to help them find higher paying work. In *Second AAAI Conference on Human Computation and Crowdsourcing*.

Cheng, J, Teevan, J, Iqbal, S. T, and Bernstein, M. S. (2015). Break it down: A comparison of macro-and microtasks. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 4061–4064.

Chiang, C.-W, Kasunic, A, and Savage, S. (2018). Crowd Coach: Peer Coaching for Crowd Workers' Skill Growth. *Proceedings of the ACM on Human-Computer Interaction* 2, CSCW (2018), 37.

Chilton, L. B, Horton, J. J, Miller, R. C, and Azenkot, S. (2010). Task search in a human computation market. In *Proceedings of the ACM SIGKDD workshop on human computation*. ACM, 1–9.

Coetzee, D, Lim, S, Fox, A, Hartmann, B, and Hearst, M. A. (2015). Structuring interactions for large-scale synchronous peer learning. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM, 1139–1152.

Dontcheva, M, Morris, R. R, Brandt, J. R, and Gerber, E. M. (2014). Combining crowdsourcing and learning to improve engagement and performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3379–3388.

Durward, D, Blohm, I, and Leimeister, J. M. (2016). Crowd work. *Business & Information Systems Engineering* 58, 4 (2016), 281–286.

Fechner, G. T, Howes, D. H, and Boring, E. G. (1966). *Elements of psychophysics*. Vol. 1. Holt, Rinehart and Winston New York.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

Hanrahan, B. V, Willamowski, J. K, Swaminathan, S, and Martin, D. B. (2015). TurkBench: Rendering the market for Turkers. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 1613–1616.

Hara, K, Adams, A, Milland, K, Savage, S, Callison-Burch, C, and Bigham, J. P. (2018). A data-driven analysis of workers' earnings on amazon mechanical turk. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 449.

Hara, K and Bigham, J. P. (2017). Introducing people with ASD to crowd work. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, 42–51.

Hitlin, P. (2016). *Research in the Crowdsourcing Age, a Case Study: How Scholars, Companies and Workers are Using Mechanical Turk, a" gig Economy" Platform, for Tasks Computers Can't Handle*. Pew Research Center.

Horton, J. J. (2011). The condition of the Turking class: Are online employers fair and honest? *Economics Letters* 111, 1 (2011), 10–12.

Horton, J. J and Chilton, L. B. (2010). The labor economics of paid crowdsourcing. In *Proceedings of the 11th ACM conference on Electronic commerce*. ACM, 209–218.

(ILO), I. L. O. (2016). Non-standard employment around the world: Understanding challenges, shaping prospects. (2016).

Ipeirotis, P. G. (2010). Analyzing the amazon mechanical turk marketplace. *XRDS: Crossroads, The ACM Magazine for Students, Forthcoming* (2010).

Irani, L. C and Silberman, M. (2013). Turkopticon: Interrupting worker invisibility in amazon mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 611–620.

Irani, L. C and Silberman, M. (2016). Stories we tell about labor: Turkopticon and the trouble with design. In *Proceedings of the 2016 CHI conference on human factors in computing systems*. ACM, 4573–4586.

Kaplan, T, Saito, S, Hara, K, and Bigham, J. P. (2018). Striving to earn more: a survey of work strategies and tool use among crowd workers. In *Sixth AAAI Conference on Human Computation and Crowdsourcing*.

Karger, D. R, Oh, S, and Shah, D. (2011). Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems*. 1953–1961.

Katz, M. (2017). Amazon Mechanical Turk Workers Have Had Enough. (2017). https://www.wired.com/story/amazons-turker-crowd-has-had-enough/

Ke, G, Meng, Q, Finley, T, Wang, T, Chen, W, Ma, W, Ye, Q, and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*. 3146–3154.

Ketkar, N. (2017). Introduction to pytorch. In *Deep learning with python*. Springer, 195–208.

Kuek, S. C, Paradi-Guilford, C, Fayomi, T, Imaizumi, S, Ipeirotis, P, Pina, P, and Singh, M. (2015). The global opportunity in online outsourcing. (2015).

Kuroda, T and Hasuo, E. (2014). The very first step to start psychophysical experiments. *Acoustical Science and Technology* 35, 1 (2014), 1–9.

Litman, L, Robinson, J, and Abberbock, T. (2017). TurkPrime. com: A versatile crowdsourcing data acquisition platform for the behavioral sciences. *Behavior research methods* 49, 2 (2017), 433–442.

Martin, D, Hanrahan, B. V, O'Neill, J, and Gupta, N. (2014). Being a turker. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*. ACM, 224–235.

Mason, W and Suri, S. (2012). Conducting behavioral research on AmazonâĂŹs Mechanical Turk. *Behavior research methods* 44, 1 (2012), 1–23.

McInnis, B, Cosley, D, Nam, C, and Leshed, G. (2016). Taking a HIT: Designing around rejection, mistrust, risk, and workers' experiences in Amazon Mechanical Turk. In *Proceedings of the 2016 CHI conference on human factors in computing systems*. ACM, 2271–2282.

Nair, V and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. 807–814.

O'neill, J and Martin, D. (2013). Relationship-based Business Process Crowdsourcing?. In *IFIP Conference on Human-Computer Interaction*. Springer, 429–446.

Reichl, P, Egger, S, Schatz, R, and D'Alconzo, A. (2010). The logarithmic nature of QoE and the role of the Weber-Fechner law in QoE assessment. In *2010 IEEE International Conference on Communications*. IEEE, 1–5.

Rzeszotarski, J. M and Kittur, A. (2011). Instrumenting the crowd: using implicit behavioral measures to predict task performance. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, 13–22.

Saito, S, Chiang, C.-W, Savage, S, Nakano, T, Kobayashi, T, and Bigham, J. P. (2019)a. TurkScanner: Predicting the Hourly Wage of Microtasks. In *The World Wide Web Conference*. ACM, 3187–3193.

Saito, S, Nakano, T, Kobayashi, T, and Bigham, J. P. (2019)b. MicroLapse: Measuring Workers' Leniency To Prediction Errors of Microtasks' Working Times. In *Submitted to the 22nd ACM Conference on Computer Supported Cooperative Work and Social Computing Companion (Submitted)*. ACM.

Salehi, N, Irani, L. C, Bernstein, M. S, Alkhatib, A, Ogbe, E, Milland, K, and others, . (2015). We are dynamo: Overcoming stalling and friction in collective action for crowd workers. In *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. ACM, 1621–1630.

Sher, V, Bemis, K. G, Liccardi, I, and Chen, M. (2017). An empirical study on the reliability of perceiving correlation indices using scatterplots. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 61–72.

Silberman, M, Ross, J, Irani, L, and Tomlinson, B. (2010). Sellers' problems in human computation markets. In *Proceedings of the acm sigkdd workshop on human computation*. ACM, 18–21.

Thies, W, Ratan, A, and Davis, J. (2011). Paid crowdsourcing as a vehicle for global development. In *CHI Workshop on Crowdsourcing and Human Computation*.

Woodworth, R. S and Schlosberg, H. (1954). *Experimental psychology*. Oxford and IBH Publishing.

Wu, M.-H and Quinn, A. J. (2017). Confusing the crowd: Task instruction quality on amazon mechanical turk. In *Fifth AAAI Conference on Human Computation and Crowdsourcing*.

Yelle, L. E. (1979). The learning curve: Historical review and comprehensive survey. *Decision sciences* 10, 2 (1979), 302–328.