The unintended consequences of automated scripts in crowdwork platforms: A Simulation study in MTurk

Haoyu Xie
Information School
The University of Sheffield
Sheffield, S10 2TN, United Kingdom
hxie5@sheffield.ac.uk

Efpraxia D. Zamani
Information School
The University of Sheffield
Sheffield, S10 2TN, United Kingdom
e.zamani@sheffield.ac.uk

Alessandro Checco
Computer Science Department
University of Rome La Sapienza
Piazzale Aldo Moro, 5, 00185 Roma RM, Italy
alessandro.checco@uniroma1.it

Abstract

Crowdworkers on platforms like Amazon Mechanical Turk face growing competition as a result of the global excess supply of digital labour. As a result, many crowdworkers are turning to automated scripts, which help them locate better tasks faster and to boost their earnings. However, to date, it is not clear whether and to what extent the use of such scripts influence the opportunities for those crowdworkers who do not use them. This an important aspect that warrants further exploration because it can have negative implications for the health of crowdwork platforms. In this study, we use Discrete Event Simulation to identify and quantify the unintended consequences of the excessive use of automated scripts. Our findings show that, while the use of scripts allows some crowdworkers to identify and accept far more tasks than others, in the long run, this behaviour results in their competence persistence and reputational persistence and progressively to detrimental impacts for those workers who do not use scripts, and who may ultimately be forced to exit the platform. As a result, automated scripts have negative consequences, whereby their excessive use leads to a tragedy of the commons for all platform stakeholders, including the crowdworkers, the job requesters and the platform itself.

Keywords: crowdwork; platforms, microtask, HIT catching script, unintended consequences, Matthew effect, discrete event simulation

1. Introduction

Over the recent years, the world of work has been experiencing significant changes, with a number of new work modalities emerging, such as gig work, platform work and crowdwork (Connelly et al., 2021). A common denominator across all these work modalities is that access to work opportunities is offered via an online platform, such as Uber and Amazon Mechanical Turk (MTurk), where workers are largely treated as freelancers. In many cases, these new work modalities have offered great benefits to their workers, because in effect they create and offer employment opportunities where they previously did not exist, such as among marginalised individuals and communities or hard-to-reach areas (Idowu and Elbanna, 2020; Malik et al., 2021). Yet, at the same time, because such platforms operate outside or at the margins of labour law, there is very little room for regulation or for ensuring that labour rights are protected (Fredman et al., 2020). Even more crucially, scholars have argued that such platforms seem to be unfair by design, shifting risk and insecurity from employers to workers, by restricting and controlling how power is distributed between them (Fieseler et al., 2019a).

As a result, the impact of the power imbalance in crowdwork often results in poor working conditions (Gegenhuber et al., 2020a). As the target platform for this study, Amazon Mechanical Turk (MTurk) allows requesters to post Human Intelligence Tasks (HITs) and recruit crowdworkers to complete them for a small monetary reward. HITs are also known as microtasks, which include identifying objects from images or videos, answering survey questions, etc. More often than not, crowdworkers earn below the minimum wage (Hara et al., 2018), which has been associated with ill-designed tasks and high rejection rates of completed tasks, among other factors (Gadiraju et al., 2017; McInnis et al., 2016). At the same time, crowdworkers find themselves in intense competition against each other: to identify high-paying microtasks before other crowdworkers to secure them, and then complete them as soon as possible and to the best of their abilities so that they can move on to the next batch of microtasks (Gerber, 2021), and ultimately increase their overall ratings and income.

Today, the combination of fierce competition among crowdworkers (D’Cruz and Noronha, 2016), and the excessive global digital labour supply has resulted in many crowdworkers turning to the use of scripts, such as PandaCrazy Max and TurkerView, for finding and filtering microtasks (Irani and Silberman, 2013). These scripts are essentially automated bots that usually run on the crowdworker's browser in the form of a browser plugin, extension or web script (Williams et al., 2019). They help crowdworkers identify those microtasks that offer higher rewards, and which align with their skills and interests. Therefore, the use of these scripts allows crowdworkers to spend less time investigating whether they can address the requirements of the task (the script has done it already for them) and to release some of the pressure of continuously searching for relevant microtasks. As such, identifying and securing HITs via such scripts is far more efficient than doing it manually (Hellman 2021; DonovanM 2018).

While such scripts are becoming indispensable, it is unclear what is their impact on the crowdworkers that do not use them. For example, there is evidence to suggest that the use of such scripts may pose an entry barrier for novice crowdworkers and result in fewer opportunities (Davis and Kanopka, 2020). It is quite possible that due to frustration and/or inability to secure quality tasks, these crowdworkers may exit the platform altogether, which can have negative implications for the health of later. Against this backdrop, in this study we examine the possible unintended consequences of the widespread use of microtask catching scripts by examining different scenarios using discrete event simulation with parameters modelled on real-world measurements. We find that, under certain conditions: a) crowdworkers who use such scripts trigger a tragedy of the commons phenomenon (Greco and Floridi, 2004), whereby they reserve too many tasks, which they fail to complete on time. As such, many of these microtasks eventually expire, negatively impacting the job requester and the platform; b) those crowdworkers who don't use microtask catching scripts are unable to catch any of the better microtasks because these become reserved immediately by scripts; c) those who use scripts increase their reputation and gain a higher skilled status on the platform at the expense of those who don't use scripts, giving rise to the Matthew effect (Merton, 1988).

The contributions of this study enrich the current discourse on crowdwork literature in several ways. First, while the crowdwork literature has acknowledged the importance of using tools for the purpose of increasing hourly income (Kaplan et al., 2018), the broader implications and the unintended consequences on crowdworkers' working conditions, e.g., opportunities, earnings per time unit, including task quality and task completion efficiency, remain relatively unknown. In our study, by simulating different scenarios, we quantify these in order to explicate what the impacts are for the workers and the platform, examining the effects on workers' income and the microtasks. Second, we provide a set of recommendations for script designers and platforms using simulations in an effort to reduce the impacts of the tragedy of the commons effect and to improve batch diversity. Previous studies have proposed ranking mechanisms, based on the premise that platforms will improve their operation standards to become more appealing to their clients (Fredman et al., 2020). We believe that such attempts overestimate employers' sensitivity regarding working conditions, especially because platforms have triggered a race to the bottom for cheaper labour (Altenried, 2020). Instead, we detail the ways in which the health of a platform can be in jeopardy due to the current design of microtask catching scripts, whereby crowdworkers become disillusioned and quit the platform entirely. This we believe is a greater incentive for implementing changes in the way scripts currently operate, mitigating their negative impacts, facilitating the healthy growth of the platform, and indirectly improving the working conditions of crowdworkers.

We can summarise our research questions as follows:

RQ1: To what extent the use of automated scripts affects worker diversity, hourly wages, and equity in the context of crowdwork?

RQ2: How can we theorise about the use of automated scripts by crowdworkers and their consequences?

The article is structured as follows. First, we present the theoretical background of our study and related work. Then, we described our research method and the data we used. In the fourth section, we report our results. Then, we offer a discussion based on our major findings in relation to the existing literature and draw the study's major contributions in relation to research and practice, and the article closes by discussing the limitations and avenues for further research.

2. Background

2.1 Crowdwork and microtask catching scripts

Within a crowdsourcing platform, such as Amazon Mechanical Turk (MTurk), job requesters publish microtasks through an open call to an undefined number of workers (Blesik et al., 2021). These microtasks are called Human Intelligence Tasks (HITs), and typically are easier for humans to solve rather than for computers, but require the participation of the crowd due to their large volume. Typical microtasks or HITs include market research questionnaires for a particular industry, or requests for participants to transcribe text from an audio recording (Gadiraju et al., 2014). In the case of MTurk, for example, the crowdworkers involved in a HIT usually spend a few minutes to a few hours on HIT completion by submitting their survey responses or data outputs, as requested by the job requester who posted the HITs. The requester ultimately pays the crowdworker with a monetary reward, depending on whether the quality of the output is acceptable or not. Quite often, novice crowdworkers have to complete a large number of HITs in order to increase their number of completed HITs and improve their HIT approval rate (Hara et al., 2018). The HIT approval rate is a metric of an individual's quality of HIT completion over time. It is commonly used by job requesters as a filter when choosing their target crowdworkers, and therefore as a metric it often determines whether or not crowdworkers are eligible to receive high-rewarded quality tasks (Savage et al., 2020; Kaplan et al., 2018).

Besides the time spent on executing a HIT, a crowdworker spends time in searching and identifying relevant HITs, reading instructions in relation to their execution, and potentially learning how to complete them, including learning how to interact with the HIT user interface in many cases. This extra time is not paid for, and often it exceeds the time dedicated to executing the chosen HIT (Chilton et al., 2010). To address this issue, crowdworkers turn to the use of HIT automated scripts (henceforth referred to as catching scripts), such as Panda Crazy and MTurk Suite (Irani and Silberman, 2013; Ramirez, 2021; *TurkerView mTurk Forum*, 2021). These scripts allow crowdworkers to 'catch' HITs posted by specific job requesters (based on their ID number), with high ratings or based on the specificities of the HIT (Saito et al., 2019), whereby the requirements of the HIT fit their prior experience or personal interests (Dror et al., 2011; Geiger and Schader, 2014). In other words, these scripts support selective and automatic catching of HITs based on personal preferences.

In this study, scripts specifically refer to the tool, bots, or extensions that allow the workers to accept HITs automatically. These scripts allow them to customise the frequency and the number of caught HITs and create batch IDs (group of HITs) so as to continuously receive HITs that fit predefined criteria (DonovanM, 2018; Hellman, 2021). In short, such advanced HIT catching scripts provide crowdworkers with extra opportunities to locate and secure their preferred HITs. For some crowdworkers, who choose to participate in community forums, such as Turker Nation in Slack, these opportunities are further enhanced (Zyskowski and Milland, 2018). These community forums bring crowdworkers and job requesters together, who can exchange information on HITs outside the boundaries of the platform. On the one hand, such forums allow participating crowdworkers to access additional information by directly contacting the job requesters and to identify higher quality tasks before others. On the other hand, job requesters may access crowdworkers through these forums and directly allocate them HITs, bypassing the platform's competitive process.

2.2 Previous work on catching scripts

The use of catching scripts is essentially a technical simulation of human behaviour. Fundamentally, the script allows individuals to expedite the process of identifying and securing extremely limited items in an online platform, particularly when the demand for these items is high and exceeds the supply (Vancea and Nemirschi, 2020). These items may range from limited edition sneakers and other apparel to concert tickets, which are then resold by the resellers to secondary markets (e.g., Craiglist, eBay and social networking sites, such as Twitter) for a profit.

While the impact of the use of HIT catching script has been extensively researched in the online retailing area, it is still in its infancy in the crowdwork domain. To date, the literature has explored the use of catching scripts as a means to overcome platforms' poor search functionality for the identification of HITs (El Maarry et al., 2018; Kaplan et al., 2018). In addition, scholars have developed classifications of these scripts to support further development of such tools (Williams et al., 2019; El Maarry et al., 2018). There are, however, studies that have focused on the impacts of these scripts on crowdworkers' workflow and daily life, whereby it has been found that the use of these scripts can cause disruptions, interfere with everyday life (Williams et al., 2019), and lead to tasks being abandoned as crowdworkers become overwhelmed with too many HITs being automatically accepted (Williams et al., 2019). However, the use of such scripts can have positive effects as well as they allow crowdworkers to take advantage of all available opportunities to take on highly rewarding HITs (Williams et al., 2019). More importantly, the use of HIT catching script reduces the amount of unpaid time crowdworkers spend on HIT search and increases access to quality HITs (Kaplan et al., 2018). To date, however, research on crowdwork has not quantified the possible negative implications of catching scripts on crowdwork.

Within the crowdwork domain, crowdworkers use these catching scripts to capture HITs that align with their own interests and skills. This enables them to maximise their income while reducing the time spent on familiarising themselves with HITs' guidelines and interfaces, which typically results in unpaid labour. Experienced crowdworkers may earn up to $12 per hour with the help of such scripts (Newman, 2019), because they have a better chance of securing higher paying HITs, and in turn increasing their overall ranking in the platform once they complete them. A study has shown that high-paying HITs with high reputation scores are reserved within seconds from being published (Hanrahan et al., 2018). Therefore, crowdworkers who use catching scripts have a clear advantage over those who don't, making it difficult for less experienced crowdworkers to identify HITs with even moderate rewards. Progressively, this has resulted in high frequency of crowdworkers being frustrated and leaving the platform altogether, which is detrimental for its healthy growth (Hanrahan et al., 2018). This can be concerning because reduced participation of new crowdworkers could have a negative impact on the diversity of HIT responses. In terms of short-term impacts, high-quality HITs tend to be captured by workers with more effective HIT catching tools, meaning that, when there are no limits as to how many HITs a worker may undertake, highly rewarding HITs in theory may be completed by a single worker. In other words, the diversity of responses decreases (Hanrahan et al., 2018). In addition, many high quality HITs may be captured by automated scripts and sit idle in a few crowdworkers' queues, until they are completed or until they expire because the crowdworker is unable to complete all those they have captured within the time allotted by the job requester. In the latter case, the completion time of the entire batch gets delayed.

What the above indicates is that, while the original intent of using automated scripts for catching HITs was to increase crowdworkers' efficiency, their excessive use can have destabilising effects for the ecosystem, inhibiting the healthy growth of the platform, and negatively impacting the crowdworkers who do not use them. In what follows, we explore why platforms may remain open to the use of such automated scripts despite the potential negative impacts on their ecosystem and what measures they take in order to control against excessive use.

2.3 Why do platforms allow the use of catching scripts?

Crowdwork platforms, like most digital platforms, are in essence marketplaces where buyers and sellers can meet and exchange services for an agreed fee (Giaglis et al., 2002). Similarly, to other digital platforms, in order to enhance the platform's functionality and therefore its overall competitiveness, platforms provide complementors, also known as third parties, with access to the platform, who create plug-ins, add-ons and other extensions (Wessel et al., 2017). This provides sufficient autonomy to third parties and encourages complementary innovations (Boudreau, 2010; Hein et al., 2020).

In platforms, the use of automated microtask catching scripts is a reflection of platforms' openness to such complementors, where third parties are able to develop their scripts in compliance with the platform's regulations. These platforms explicitly allow the use of automated scripts so that crowdworkers are better able to search for and preview microtasks (HITs). This enhanced functionality makes platforms more attractive to crowdworkers by allowing them to optimise the workflow of searching, identifying and reserving preferred and high-rewarding HITs in batches.

However, the challenges observed in other platforms (e.g., crowdfunding and open-source platforms) are also present. Wessel et al. (2017), for example, discuss that a major challenge is identifying the balance between platform openness towards third parties and maintaining control. Being too open can potentially destabilise the ecosystem. In the case of crowdwork, the stability of the ecosystem extends beyond retaining oversight of operations; openness needs to be balanced against the need for ensuring the fair treatment of crowdworkers and providing job requesters with high quality outputs, both of which feed into the healthy growth of the ecosystem. Amazon Mechanical Turk, for example, prohibits the use of scripts that send requests at an excessively high frequency and those that automatically accept HITs (*Amazon Mechanical Turk*, 2018). It is unclear, however, whether this ban is for the purpose of maintaining the operational stability of servers, for ensuring the fair treatment of all crowdworkers or for satisfying diversity in the data collected via HITs. In reality, while the imposed limitation on the frequency supports operational stability, it is unknown whether it can ensure fairness across those who use automated scripts and those who don’t.

3. Theoretical Framing of the Study

Within a crowdwork environment, crowdworkers are described by their approval rating, which indicates the number of successfully completed HITs. Once a crowdworker completes a HIT, the job requester will examine the output and if it is satisfactory, they will approve it and reward the worker, and the worker’s approval rating will increase; if the output is not approved, the worker is not paid and their approval rating will decrease. In addition, many HITs require a minimum approval rating. This means that there are cases whereby crowdworkers with low approval ratings, even due to being newcomers to the platform, are prohibited by design from accepting the said HITs (Brawley and Pury, 2016).

This approval system bears resemblance to the typical ranking and reward systems observed elsewhere (e.g., online marketplaces, h-index), whereby the ranking of individuals (e.g., merchants, researchers) is said to reflect expertise and mastery in particular types of HITs (Matherly, 2019). However, studies have shown that such systems are prone to bias, whereby rankings may push individuals to adopt strategic behaviours that do not necessarily support the flourishing of the ecosystem (Shen et al., 2015), and may lead to unfair treatment (Idowu and Elbanna, 2020)

In this study, we are interested in exploring, but crucially, quantifying the impacts of automated catching scripts as materialised through the use of such ranking systems. We thus frame our empirical study drawing from Merton’s Law of Unintended Consequences, which is also known as the Matthew effect (Merton, 1968). The Matthew Effect suggests that those who enjoy greater visibility receive greater rewards, and that those who are less visible, they receive disproportionately lower rewards and less recognition for the same performance. The Matthew effect is well documented and recognised in areas such as scientometrics and sociology for the examination of hierarchical systems (Fralich and Bitektine, 2020). However, much less is known regarding the extent to which the Matthew effect promotes or restricts equal opportunities for participation and reward within an environment governed by the presence of automated scripts.

3.1 The Matthew effect: competence and reputational persistence

When discussing the reward system for scientific contributions, Merton argued that society tends to honour those with greater reputation and visibility, irrespective of the degree of their contribution to a particular piece of work (Merton, 1968). Similarly, applications for research and development grants put forward by less known consortia and companies are often denied funding, because decisions tend to be influenced by the candidates' award history (Van Looy et al., 2004). In both cases, those who are less visible and newcomers are disadvantaged, without this indicating, however, that they are any less capable. Rather, such disadvantages and exclusions are the result of unfounded perceptions regarding one's competence and reputation (Antonelli and Crespi, 2013; Dannefer, 1987; O'Rand, 1996).

Antonelli and Crespi (2013) have argued that competence persistence is an expression of virtuous Matthew effects, whereby the resources at one's disposal allow them to increase their overall competence, and therefore their outputs, such as income, are simply higher.

In the case of crowdwork, automated scripts allow crowdworkers to secure microtasks that meet certain criteria with a frequency and duration that far exceeds that of a manual workflow. Crowdworkers who leverage these scripts can gain a higher skilled status compared to their peers, gaining a competitive edge. This status is often achieved at the expense of crowdworkers with lower approval ratings or less skilled in using scripts (Reschke et al., 2018). More importantly, the use of automated scripts influences one's income. Super Turkers, those who use multiple scripts together to access high quality HIT referral channels, gain extremely high earnings in comparison (Savage et al., 2020). Because high rewarding HITs are finite and scarce, scripts enable Super Turkers to locate them and reserve them in a matter of seconds when they do become available.

This is an example of how competence persistence develops and endures. Super Turkers invest more time and effort in using automated scripts for HIT filtering (Kaplan et al., 2018), whereby HIT filtering allows them to identify HITs that match their skills and competencies, which they can tackle successfully, on time and receive the relevant reward. Those who do not or cannot use such scripts have fewer opportunities to identify enough HITs relevant to their skills, and complete fewer and fewer HITs overall. Worse still, because of fewer opportunities, crowdworkers often opt to complete lower quality HITs, posted by less (or non) reputable job requesters to increase their overall income. This is an additional risk for crowdworkers, because less reputable job requesters often reject without an explanation the submitted output, in which case the worker is not paid and the approval rating gets further reduced

(Deng et al., 2016). As such, non-script crowdworkers become trapped in a vicious cycle caused by the lack of technical competence, the latter being irrelevant to the actual HITs.

These scripts are free to access and use. Therefore, theoretically all crowdworkers should have the same technical advantage, i.e., the same competence persistence. However, when comparing the six widely used HIT catching scripts on the market, several differences in the technical implementation methods can be recognised (DonovanM, 2018; Hasan, 2018; Hellman, 2021; Ramirez, 2021; Schultz, 2020; Watwani, 2021). These include whether the HTTP request frequency of a single target HIT can be dynamically adjusted when catching multiple target HITs, whether the new HITs can be auto detected based on keywords and categories, etc. The difference of technical features such as migration management of quality HIT IDs influence whether users can detect and catch newly released quality HITs earlier than other workers. At the same time, the different difficulty of using different scripts and the few scripts containing advanced features require workers to pay a subscription to use them. In addition, not all crowdworkers have the skills to use such scripts successfully.

There is still little research on the use of scripts depending on crowdworkers' background and experience, especially in terms of examining worker behaviour at the micro level. Experienced crowdworkers are more likely to be aware of new scripts, to know how to use them and thus locate and secure high-reward tasks (Hanrahan et al., 2021; Irani & Silberman, 2013; ChrisTurk, 2022). Novice crowdworkers need to spend more time searching for tasks, which in turn creates a clear income gap between Super Turkers and average crowdworkers (Savage et al., 2020). This gap is, of course, due to a combination of factors such as task search time and completion time. In general, however, the above lead to competence persistence.

Reputation persistence refers to vicious Matthew effects, whereby it is posited that one's track record is testament to their skills and abilities, and is thus used as the evidence base for their selection for future employment, funding (Antonelli & Crespi, 2013), and for the purposes of our study, for HITs. For example, with regards to scientific contributions, it is not only the discovery itself that affects the popularity of an academic discovery, but also the status of the scientist who made it (Merton, 1968).

In platforms such as Mechanical Turk, this track record is embodied in the approval rating. Drawing from the economics of big data, that relate to search costs and information asymmetry (Yan et al., 2015), we posit that a job requester is more likely to assign a HIT to a crowdworker with higher approval ratings and a higher number of completed HITs, because these scores can help them filter out crowdworkers whose quality of work is not guaranteed, and because, consciously or not, HITs by workers with higher scores will be perceived as more trustworthy.

However, approved HITs contribute towards approval ratings, whereby the higher the approval rating the higher quality HITs the crowdworker can access in the future. As such, job requesters' filtering behaviour is critical for crowdworkers, because it may permit or prohibit access to future higher earnings. At the same time, it inevitably filters out newcomers who may compete equally quality-wise but less well quantity-wise; thereby prohibiting them from securing better rewarding HITs in the future, and potentially driving them out for crowdwork altogether. As such, reputational persistence gives way to further unintentional consequences.

4. Methodology

4.1 Rational of the simulation framework

In this study, we are interested in exploring the unintended consequences stemming from the use of HITs catching scripts, and specifically quantifying the effects of technical competence. Our hypothesis is that the success rate of catching HITs for each crowdworker will be influenced by the persistence of their technical skills under the condition that everyone has the same access to these HITs.

We investigate the above using data collected through a series of simulations. Our simulation framework has been designed with the view to conduct controlled variable experiments and simulate different HIT acceptance behaviours of crowdworkers to achieve more in-depth insights. More specifically, this simulation framework allows us to quantify the advantage that scripts-using crowdworkers have over non-script-using crowdworkers when accepting HITs from the same batch and the impact of this on the completion of the whole batch.

In the proposed simulations, scripts-using crowdworkers consistently fill their HIT queues without considering their ability to complete all accepted HITs in a timely manner. As such, the question we wish to answer is whether scripts-using crowdworkers cost non-script-using crowdworkers their rightful work opportunities (as depicted in Figure 1). In turn, we wish to explore the impact of this on the final number of completed HITs. A possible scenario is that when there is only one batch of HITs available on the platform and all HITs are captured by scripts, non-script-using crowdworkers will remain idle until a new HIT gets published (Hanrahan et al., 2018).

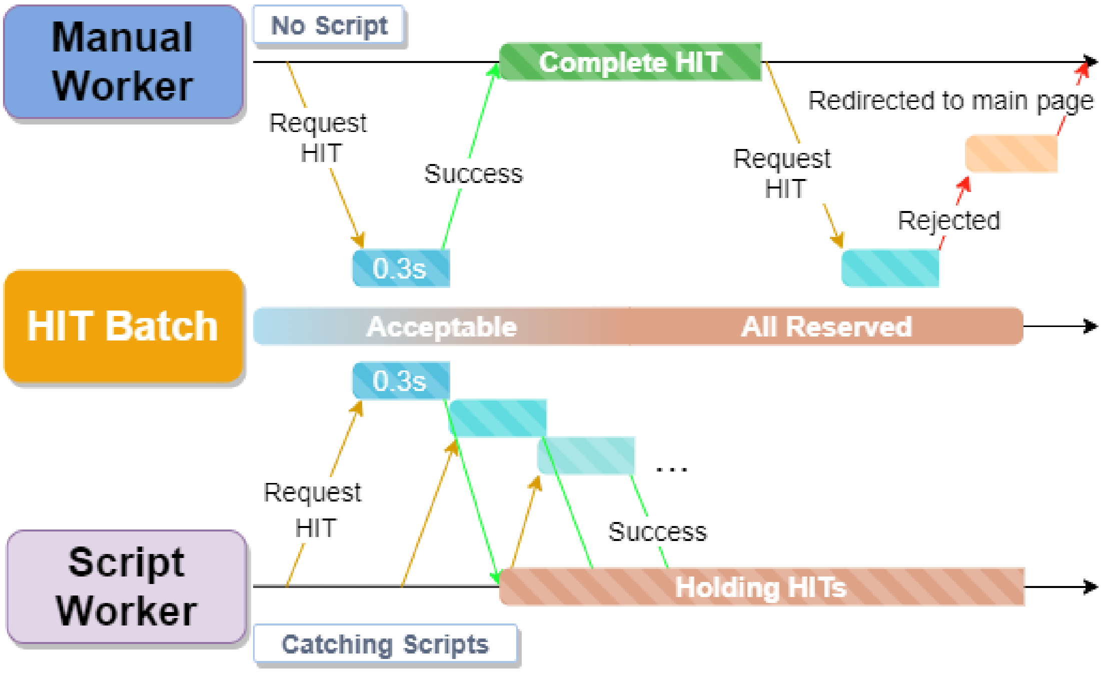


Figure 1 Workflow diagrams for both manual (non-script-using) and script workers (script-using)

4.2 Research approach

To quantify the consequences of HIT catching scripts, we modelled our scenario with SimPy, which is a discrete-event simulation (DES) framework in Python (Müller et al., 2021). DES allows us to observe phenomena that occur under complex rules. Since the work behaviours and the use of scripts consist of a complex series of independent events that interact with each other (e.g., a reserved HIT cannot be found by other crowdworkers before expiring in somebody else's HIT queue), it is appropriate to merge these complex decision logics through DES to model the entire process of a batch being received and performed by different crowdworkers. This is extremely difficult to achieve with other types of modelling.

When comparing simulations to experiments conducted directly on the MTurk platform, our observations would be subject to a number of confounding factors, such as the number of crowdworkers online at the same time, which varies greatly depending on the time of day and the time zone in which the crowdworkers are located. In addition, the popularity of the HIT, which is influenced by factors such as the type of HIT, the reward level and the reputation of the job requester, would also create uncertainty as to whether the crowdworkers will attempt to accept the HIT.

These confounding factors cannot be effectively controlled in a real-world environment. As a result, we choose to investigate the issue using the DES, which is initialised with real-world data for controlled environments. In addition, collecting experimental data from the platform requires interventions such as using the catching scripts ourselves. Such interventions could potentially bias our results due to negative implications for non-script-using crowdworkers. In addition, the simulation allows us to perform multiple iterations and adjust the parameters of the experiment with extremely low cost, thus helping us to identify additional phenomena worth further exploration. Regarding the ethical issues, our experimental data derive from simulations through the MTurk Sandbox requester where we created multiple crowdworker accounts (owned by the researchers), as such there was no collection of live data and no crowdworkers were involved in our simulation.

4.3 Data collection

Before the development of the simulation framework, the durations between different key events throughout the life cycle of the HITs (e.g. time for a HIT to become visible, time for an expired HIT to be published again etc.) were estimated on the MTurk Developer Sandbox (*amazon mturk requester*, 2021).

The MTurk Developer Sandbox is a mirror of the production platform which allows requesters to test their microtasks before publishing on the production site (*Amazon Mechanical Turk Developer Guide*, 2021). Other than the monetary transfer being disabled, the MTurk Developer Sandbox has the same functionalities as the MTurk platform.

4.4 Simulation Design

Apart from the aforementioned time parameters, in order to initialise the DES we set up (i) the percentage of crowdworkers using scripts, (ii) the number of crowdworkers per batch, (iii) the crowdworker expected HIT completion time, and (iv) the crowdworker acceptance strategy.

Regarding (i), no studies have yet been conducted to estimate the number of people using scripts on MTurk. Moreover, this study aims to construct a simulation framework to quantify the unintended consequences of using catching scripts at a relatively micro level. Therefore, the total number of crowdworkers (ii) is limited to a maximum of 50. For (iii), each HIT can be reserved by one crowdworker for up to 1 minute and usually takes half a minute to complete. But the exact time that one crowdworker spends on completing one HIT is a random number within a truncated exponential distribution with the mean value of 30 seconds (Araman et al., 2019).

Regarding (iv), a non-script-using crowdworker would normally start working on a batch by accepting a HIT from it. Then, they would accept a new HIT after the submission of the previous one. Script-using crowdworkers, on the other hand, will use scripts to continuously accept as many HITs as possible while attempting to complete the HITs with enough time remaining from their HIT queue. We do not limit the number of HITs that can be completed by each crowdworker. Both strategies are further explained via the DES workflow diagrams in Appendix 8.1 and 8.2.

Two groups of simulations were designed to investigate the variation in unintended consequences of script use on both types of crowdworkers and the whole batch at different experimental sizes and percentages of script-using crowdworkers. The results of the DES depend on several random effects, such as whether crowdworkers have available HITs to perform all the time, so multiple runs need to be conducted to estimate the expected scores (Seneta, 2013). In this study, 10 runs for each simulation were sufficient since the results clearly reveal the trend of variation and the boxplots are not overlapping.

4.4.1 Script impact in different experimental scales

The first group of simulations obtains data by gradually increasing the size of the experiment, which is increasing the number of HITs included in a batch and the number of workers by the same proportion (10:1), while controlling for the proportion of workers of both types and the ratio of the number of workers to the number of HITs. The aim is to explores how the unintended consequences of the use of the script on the workers and the HITs themselves change as the size of the experiment increases, e.g., whether the gap between the job opportunities of both types of workers increases further or decreases.

4.4.2 Script impact in different percentages of script workers

The use of scripts gives workers a technical advantage, but what would be the impact on batch completion if there are more workers using scripts? To answer this question, simulations were conducted by increasing the percentage of script workers from 0% to 90% by maintaining a total of 500 HITs and 50 workers.

4.5 Metrics

Our analysis focuses on two sets of measurements related to the whole batch and the workers, which contain the diversity of data on the final completion of the batch, the total completion time, the number of HIT acceptance and submission for both types of workers in each simulation. In particular, the equation for the batch diversity is as follows:

$$Diversity = 1 - \frac{\sum_{i=0}^{n}(2i - n - 1)x_i}{n\sum_{i=0}^{n} x_i} \tag{1}$$

Equation (1) is derived from the equation for Gini coefficient of inequality, which is widely used in the field of economics (Cowell, 2011). x_i stands for the array in ascending order that stores the numbers of finished HITs for each worker. 1-Gini Coefficient is applied to represent the overall equity of opportunities on doing HITs for each worker and the batch diversity. The larger the ratio, the more fairness of catching HITs for each worker, and the higher diversity of the batch results since they come from a wider range of workers.

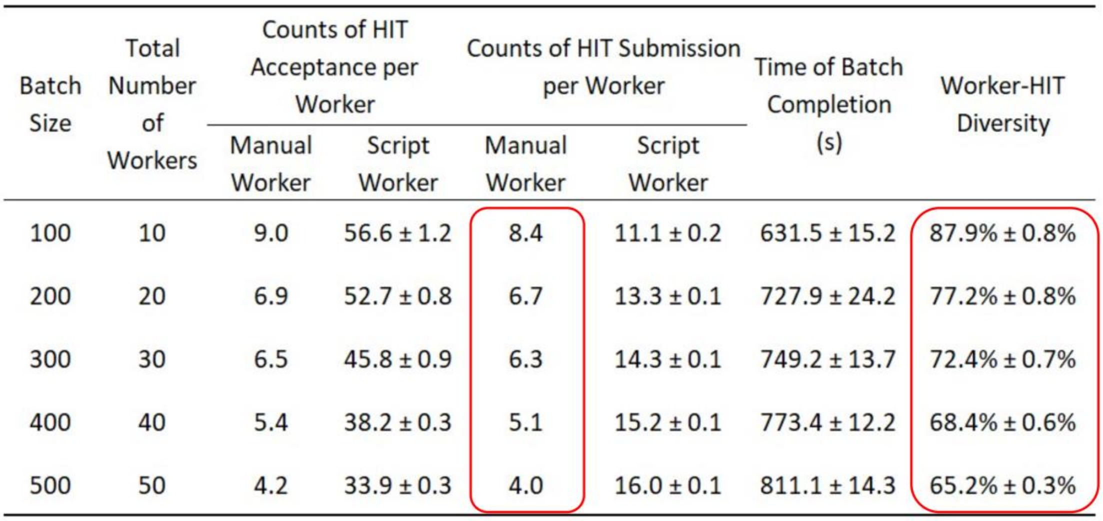
5. Analysis and Findings

5.1 Script impact vs. experimental scale

This section is dedicated to explaining the data from the first simulation and it states that: as the numbers of HITs and the number of workers grows in a constant ratio of 10:1, the unintended influences of using scripts on the completion time of the HIT, the diversity of the data and the job opportunities for non-script-using workers are further amplified.

As can be seen in Table 1, the negative impact of scripting is gradually magnified as the size increases. Fairness of catching HITs also decreases from around 0.88 to 0.65. The average number of HITs completed per non-script-using worker also dropped from 4.8 to 2.1. In other words, as the batch size increases, the script deprives the average worker of more and more job opportunities. Based on this trend, we can gain a more tangible and quantifiable understanding of the impact of catching scripts on the platform in real environments with thousands of workers involved at the same time. In addition, since the standard errors of the non-script-using workers' results are all less than 0.1, the variations of these results are not presented in the table.

Table 1 Summary of simulation statistics under 5 experimental scales



Batch Size	Total Number of Workers	Counts of HIT Acceptance per Worker		Counts of HIT Submission per Worker		Time of Batch Completion (s)	Worker-HIT Diversity
		Manual Worker	Script Worker	Manual Worker	Script Worker		
100	10	9.0	56.6 ± 1.2	8.4	11.1 ± 0.2	631.5 ± 15.2	87.9% ± 0.8%
200	20	6.9	52.7 ± 0.8	6.7	13.3 ± 0.1	727.9 ± 24.2	77.2% ± 0.8%
300	30	6.5	45.8 ± 0.9	6.3	14.3 ± 0.1	749.2 ± 13.7	72.4% ± 0.7%
400	40	5.4	38.2 ± 0.3	5.1	15.2 ± 0.1	773.4 ± 12.2	68.4% ± 0.6%
500	50	4.2	33.9 ± 0.3	4.0	16.0 ± 0.1	811.1 ± 14.3	65.2% ± 0.3%

Figure 2 indicates the cumulative number of HIT acceptance for both types of workers when batch size is 100. Due to the reason that the script-using workers caught HITs aggressively at the beginning of the simulation, many HITs expired from their queues before they even opened them, and they cannot re-accept them later anymore (unintentional HIT abandonment). What is worse, during the time they hold these HITs till expiration, the non-script-using workers had to wait due to the lack of available HITs. As can be noted from Figure 2, within less than 100 seconds of the start of the simulation, the script-using workers have reserved the majority of the 100

HITs. Such a phenomenon is the tragedy of the commons for both the workers and the requesters. Specifically, it takes away the HIT opportunities from non-script-usingworkers, extends the batch completion time, and wastes their own HIT catching opportunities due to unintentional HIT abandonment.

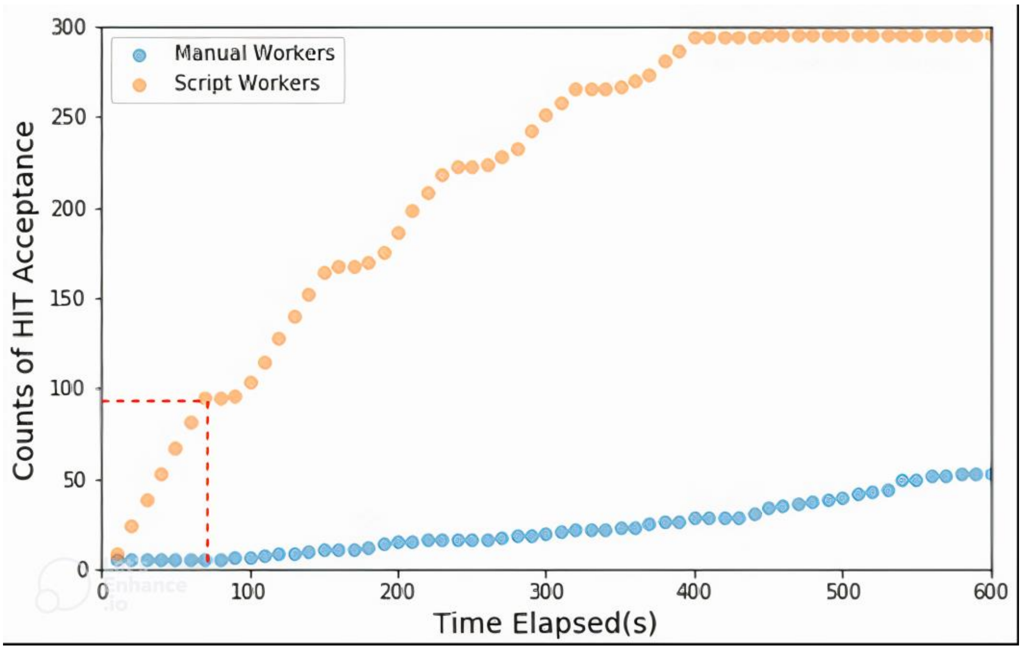


Figure 2 Counts of HIT acceptance over time for both types of workers under the batch size of 100

However, with the increase of the experimental size, the ratio of HIT submission over acceptance for the script workers increases (Table 1). As the number of total HITs increases, the penalty effect on workers for abandoning HITs becomes smaller. In other words, the impact of the tragedy of commons on script-using crowdworkers reduces. This leads to an increasing gain from high-frequency HIT catching, and thus their ratio of HIT submission over acceptance goes up. In real life, if one script-using worker can abandon many HITs without penalties, they tend to be aggressive in using catching scripts (katnapped, 2020).

From Table 1, it can be noticed that the average number of HITs completed by script-using workers gradually increases as the size of the experiment increases. This directly leads to a decrease in batch diversity due to the increasing proportion of HITs completed by script-using workers, who make up half of the total workforce. Dennis et al. (2019) collected "disturbingly low-quality responses" in their experiments on MTurk and expressed concerns about the reliability of MTurk data. This simulation validates their concerns in terms of the reduced batch diversity caused by the extensive use of scripts.

5.2 Script impact vs. percentage of script-using workers

Figure 3 shows that the technical advantage of catching scripts for users is most significant when the percentage of script-using crowdworkers reaches 20%, with an average of around 30 HITs submitted by each script worker, while it is only 5 for each non-script-using worker. At the same time, the diversity of data is at its lowest, which is around 54% (Figure 4). This is because more than half of the total HITs are completed by only 20% of total workers. Meanwhile, due to the large number of HITs being reserved by a very small number of script-using workers, the non-script-using workers cannot consistently catch the HITs, resulting in more than double the completion time (from 544.46s to 1136.96s) compared to if there were only non-script-using workers in the simulation (Figure 4).

Interestingly, however, as more workers use scripts for automatic HIT acceptance, the diversity of data gradually returns. When 90% of all workers use scripts, batch diversity returns to its initial level (88%), the same as it would have been with all non-script-using workers (89%). The increase in diversity was accompanied by a gradual decrease in total completion time, reaching the second shortest time after the simulation with all non-script-using workers at 70%. It indicates that the impact of catching scripts on the batch diversity decreases as it becomes more prevalent among all workers. However, it still has a great impact on total completion time due to consistently reserving too many HITs.

Regarding the batch completion time presented in Figure 4, when the percentage of script-using workers is 0%, there is no one reserving multiple HITs at the same time with scripts. Therefore, the batch completion time is the lowest compared with other percentages because almost no one gets delayed in their work by a lack of acceptable HITs. When there are 10% of all the workers using catching scripts, the script has the greatest positive impact on the users. With a small number of script competitors and adequate number of acceptable HITs, each script-using worker can accept as many HITs as possible without being affected by the platform's restrictions on accepting the same HITs repeatedly. But the drawback is that all the non-script-using workers, who make up 90% of the total workforce, are affected by the difficulty of catching HITs and have to slow down their work.

As more and more workers use scripts to catch HITs, there are less workers who are affected by the difficulty of catching HITs. Therefore, the batch completion time reduces after the percentage of script workers increases from 10%. However, when 70% of the total workforce are all using scripts, the use of scripts can have a far more negative impact on non-script-using workers than their positive impact on the script workers themselves, thus reducing the overall speed of batch completion.

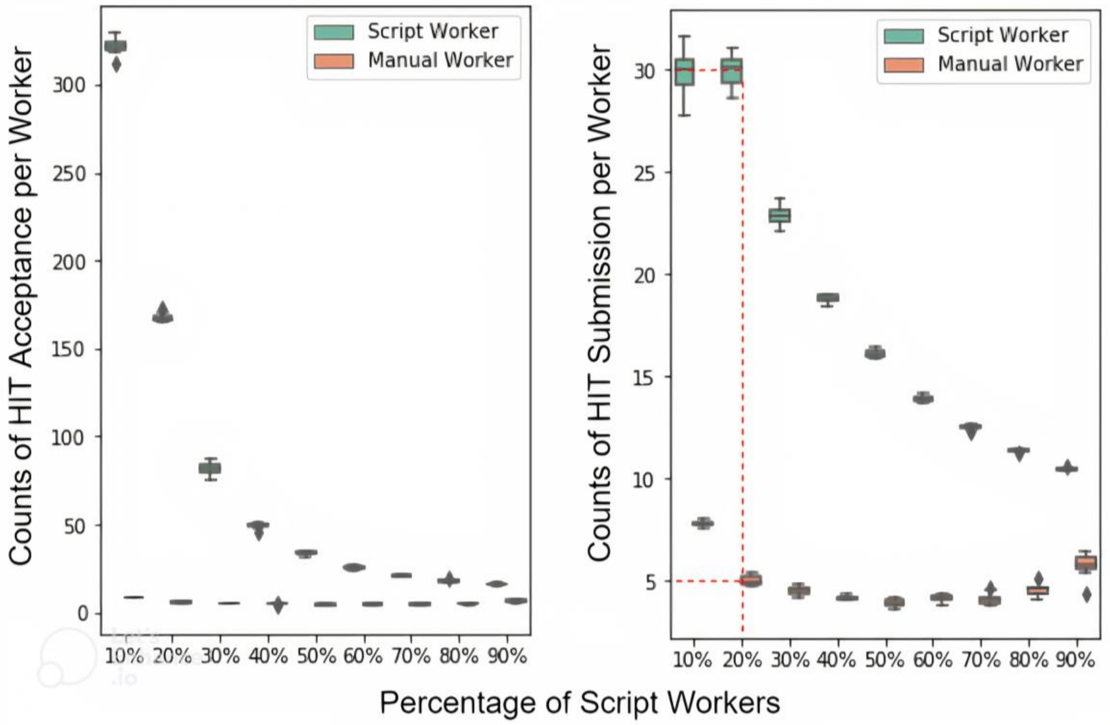


Figure 3 Worker related statistics under different percentages of script workers (batch size = 500)

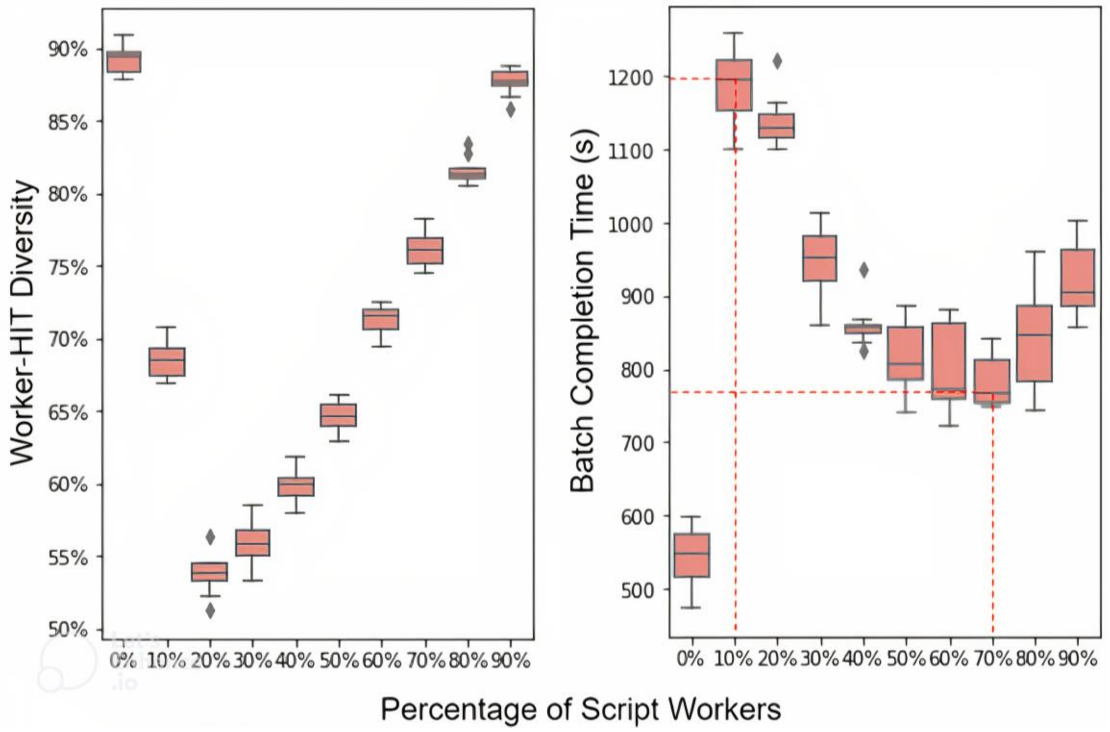


Figure 4 Batch related statistics under different percentages of script workers (batch size = 500)

6. Discussion

The aim of this study was to investigate the unintended consequences of the use of automated HIT acceptance scripts on their users, other crowdworkers, and job requesters, and thus indirectly on platforms, too. Our analysis reveals that script-using crowdworkers gain far more work opportunities within a batch than non-script-using workers, thanks to the technical advantages that scripts provide them with. Compared with the non-script-using workers, the additional HIT submissions enhance script-using workers' competence persistence, while the increase of this important metric on their worker profile might contribute in the long run to increases in reputational persistence. In other words, the technical advantage that scripts offer further increases script-using crowdworkers' job opportunities in the long-term due to the Matthew effects (Figure 5). Moreover, consistent with our hypothesis, the additional gains for workers with high rankings come at the expense of workers with lower ranking. Script-using workers were able to access additional HIT opportunities, while depriving non-script-using workers of equitable opportunities to access HITs.

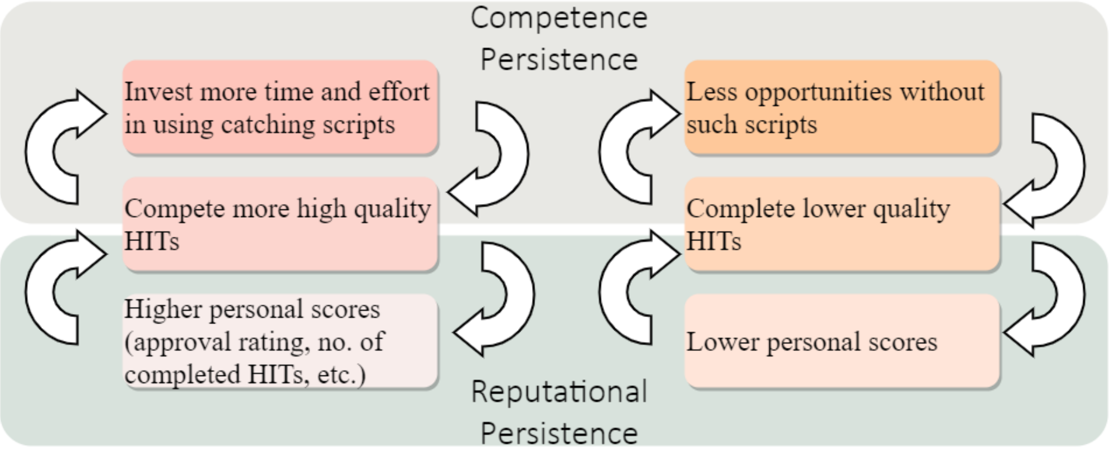

Competence Persistence
Invest more time and effort in using catching scripts
Compete more high quality HITs
Higher personal scores (approval rating, no. of completed HITs, etc.)
Less opportunities without such scripts
Complete lower quality HITs
Lower personal scores
Reputational Persistence

Figure 5 The virtuous (on the left) and vicious (on the right) Matthew Effect for crowdworkers.

Our study also reveals the existence of the tragedy of the commons. The over-acceptance of HITs by script-using workers deprives non-script-using workers of work opportunities, which in turn slows down the overall completion of the batch, leads to inevitable HIT abandonment, and results in script-using workers damaging their own future work and opportunities (Figure 6). The negative impacts of script use multiply as the scale of the batch increases, while the impact of the tragedy of the commons is magnified where the number of HITs waiting to be completed increases. However, we also found that a high prevalence of script use among workers can mitigate the impacts on batch diversity, because of the added competition between them.

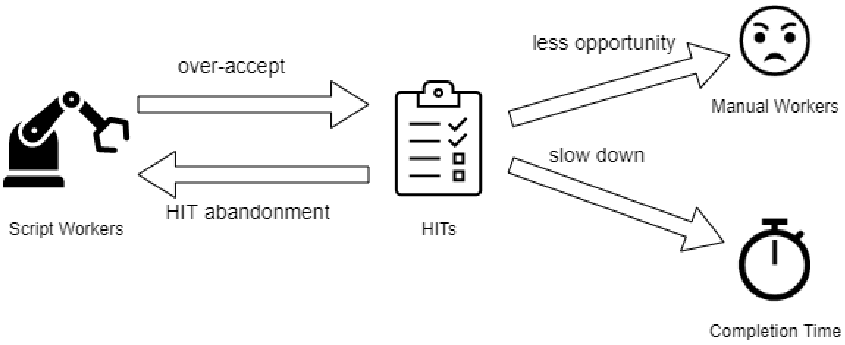


Figure 6 The tragedy of the commons caused by task over-acceptance of scripts

6.1 Theoretical Contributions and Implications for Research

Our study makes some important contributions to the crowdwork literature and platform studies, which have implications for research.

First, previous crowdwork studies with an interest on the working conditions of crowdworkers, tend to explore the phenomenon from a regulatory perspective and explore how the lack of a clear regulatory framework leaves crowdworkers exposed to low wages, job insecurity and lack of opportunities for collective action or organising (e.g., Gerber, 2021; Altenried, 2020). Other scholars focus on platform design and explore how platform features influence the power distribution between workers and job requesters (e.g., Fieseler et al., 2019; Irani and Silberman, 2013). To date, however, little attention has been paid towards analysing how the openness of the platforms to third party applications may influence the platform's success (Engert et al., 2022). Wessel et al. (2017) have indicated that platform openness may be a source of innovation and may make the platform more attractive, but at the same time, it can be a source of risks, whereby, insufficient control over third parties may destabilise the platform. Our findings extend our current knowledge with regards to how such openness may operate within the crowdwork context whereby the openness to the use of automated HIT catching scripts, provided by third parties, negatively influences the working conditions. We further quantify the impacts on the platform's participants in the short term and specifically show how the use of automated catching scripts impact the HIT acceptance strategies in the short term. Namely, we show that more than half of the total HITs may be completed by only one fifth of total workers, impacting batch diversity and significantly restricting the job opportunities of non-script-using workers. In the longer term, script workers who have completed more HITs, they will have improved their ability to use scripts and they will have enhanced their personal ranking. As such, competence and reputation persistence will lead to a continuously widening gap between script workers to and non-script-using workers, benefitting script workers at the expense of non-script-using workers. We posit that these two features, competence and reputation, can have explanatory value for many of the implications already identified by earlier studies, and therefore we consider that our work can enrich the arsenal of theories typically applied in this domain and extend our thinking.

Second, our study enriches the growing body of literature on the impact of algorithmic control and working conditions. Gol et al. (2019) argue that the reputation system that feeds on HIT completion and approval is a good estimate for future performance, allows job requesters to verify crowdworkers' credentials and supports platforms to exercise the appropriate level of control for governance purposes. However, Wood et al. (2019) expressed concerns with regards to the consequences of such algorithmic control, where job requesters are able to identify 'quality' workers on the basis of the number of HITs completed. Indeed, job requesters tend to set very high acceptance criteria for the HITs they publish to filter out less experienced workers from the large labour pool as quickly as possible (Waldkirch et al., 2021). Our findings show that the reputation system that takes into account HIT completion and approval rates is susceptible to the vicious impacts of the Matthew effect, where the extensive use of HIT catching scripts have adverse impacts on the opportunities allowed for non-script-using workers and new workers in general. New workers in particular are required to complete a high number of HITs as quickly as possible to attain an acceptable score in order to be later considered for higher quality HITs. In other words, they may find themselves completing a large volume of low quality HITs, which are low rewarding and/or posted by less reputable job requesters (Savage et al., 2020), and thus risk completing HITs that may not be approved and thus not be rewarded (Kwek, 2020). Workers who are not proficient in scripting, or whose scripts are less functional, indeed have a more difficult time obtaining quality HITs (Williams et al., 2019). As such, our study suggests that beyond the impacts of platform design on the labour conditions of crowdworkers, the behaviour of peers, i.e., other crowdworkers can have equally damaging effects.

Our findings have further implications for research that delve into algorithmic control. We show that algorithmic control over time, has a negative impact on the platform, as well. Newcomers to the platform become discouraged due to the indirect obstacles imposed by design (Brawley and Pury, 2016), and those who do not use HIT catching scripts, are more likely to abandon the platform altogether. Previous studies have underlined that high churn rate is indeed a threat for platforms because job requesters may not be able to obtain high quality results for low enough costs (e.g., Deng et al. 2016). In short, Matthew effects, not only ultimately force newcomers and non-script-using workers to leave the platform, but can potentially lead to the collapse of the platform itself.

As a result, we posit that our findings can have implications for the crowdwork literature. Studies that examine the labour conditions in the crowdwork domain and the implications of these on crowdworkers, tend to draw from qualitative data (e.g., Gegenhuber et al., 2020b; Idowu and Elbanna, 2022; Panteli et al., 2020). In addition, while there is great attention to the crowdwork ecosystem, existing scholarship in this area seems more focused on identifying design-based sources of bias (e.g., Fieseler et al., 2019b) or crafting design principles for enhancing the efficiency and functionality of such platforms (e.g., Xie et al., 2021; Yuan and Hsieh, 2018). While the findings of such studies are significant in order to identify issues pertaining to crowdwork and to improve the underlying conditions, we note that in order to capture the full extent of the negative implications, researchers need to be able to quantify these in order to adequately appreciate the true impacts. Further, our study shows that crowdwork research needs to consider the crowdwork ecosystem in its totality, i.e., different types of crowdworkers and of different abilities, as well as the platform provider, whereby this will allow for a more nuanced discussion with regards to the extent of unintended consequences and implications for the ecosystem.

6.2 Implications for Practice

Besides the theoretical implications, our study makes some important contributions to practice, as our findings can be used for considering platform openness and more crucially, informing platform design and automated HIT catching scripts.

In addition to the simulations mentioned above, we also analysed potential countermeasures including the catching frequency and HIT queue length. It was found from these simulation results that: (i) the limitation on the script catching frequency could help improve the diversity of data and reduce the time of batch completion. (ii) As the batch size grows, the number of HITs that one worker can abandon increases, and the platform's rule against accepting the same HIT repeatedly has a less penalising effect.

As Hanrahan et al. (2018) explain, to avoid the HITs being over-accepted by the catching scripts, job requesters can limit the time allotted of each HIT. This would amplify the punishing effect of the platform's prohibition on workers accepting the same HIT repeatedly, thus prompting them to reduce the catching frequency used by their scripts and deterring them from securing an excessive number of HITs, which ultimately expire in their HIT queues. Of course, limiting the maximum number of completions per worker is the most direct and effective way to improve batch diversity.

Our study further shows that it is imperative for the sustainability and healthy growth of platforms to identify ways on how third-party contributors can be encouraged to help improve the functionality of the platform while avoiding the unintended consequences on the platform’s stakeholders and the diversity of the data. Besides applying upper limits on HIT completion per worker, platforms can draw inspiration and lessons learned from the strategies typically employed by online retailers in e.g., the sneaker and ticket industries, and potentially adjust the ways HITs are assigned. For example, the platform could first receive sign-ups from all workers interested in the batch over a period, and then use a lottery or equal distribution to assign the HITs. While such an approach would impact the overall completion time of the batch, the resulting delays would not exceed those currently observed due to non-completions, and the HITs could be assigned more fairly, thus ensuring better opportunities for the majority of the crowdworkers.

7. Conclusions

In this study, we show that the use of automated HIT catching scripts can be beneficial for those crowdworkers that use them, but only in the very short term. Their excessive use leads to unintended consequences for all crowdworkers, the job requesters and the platform itself. Crowdworkers who do not use automated scripts are left with few opportunities to increase their income and are likely to exit the platform. Progressively, job requesters may become disillusioned with the low diversity in completed HITs and the high churn rates. In turn, these will impact the sustainability of the platform, as job requesters will be less likely to prefer it for posting HITs because supply will not be able to meet demand requirements. Our findings inform script designers and platforms and specifically on the ways they can mitigate these unintended consequences, with the view not only to ensure the sustainability of the platform but to also ensure that crowdworkers enjoy better working conditions and equal opportunities.

While our study provides new insights and important contributions with regards to crowdwork, it comes with important limitations, primarily due to its exploratory nature. First, we designed our study around a simulation framework, where the data we used derived through the tests on the MTurk Developer Sandbox. This means that, while we are confident with regards to our measurements, our data are experimental. Therefore, future studies should explore the possibility of obtaining the key parameters from the platform, such as average rest time for each worker, and apply them to the simulation framework, in order to generate results that representative of real-life scenarios. We consider that this work can lead to useful insights with regards to developing design principles for such platforms (Xie et al., 2021), whereby crowdworkers, experienced or not can be empowered through the design of the platform itself (Yuan and Hsieh, 2018).

Second, in platforms such as MTurk, the workers involved in HIT acceptance always change over time. Both types of workers, those using scripts and those who don’t use them, have multiple strategies in identifying and completing HITs, and their behaviour may evolve over time as they become familiar with the platform and/or the use of scripts. Therefore, future studies can explore in more depth and longitudinally crowdworker behaviours and strategies, with the view to assess changes in the competence and reputational persistence among both types of workers. This will allow to improve the current simulation framework but most importantly, it will allow studying the long-term Matthew effects.

Finally, in order to measure for inequalities, we used the Gini coefficient of inequality. The Gini coefficient is widely used in the field of economics (Cowell, 2011) for estimating inequalities in income distribution or, better put, in the wealth distribution among a given population. However, it has been criticised as providing potentially misleading results because it is not as sensitive “at the extremes”, i.e., between the very rich and the very poor (Cobham et al., 2016), who, in our study would be those with the highest reputation scores and income and those with the lowest. Therefore, future studies interested in examining inequality and working conditions in crowdwork and platforms can combine the Gini coefficient with the Palma ratio, which we were unable to do due to the size of our simulation.

Declarations

The authors have no conflicts of interest to declare that are relevant to the content of this article.

References

Altenried M (2020) The platform as factory: Crowdwork and the hidden labour behind artificial intelligence. *Capital & Class* 44(2). SAGE Publications Sage UK: London, England: 145–158.

Amazon Mechanical Turk (2018) Acceptable Use Policy. Available at: https://www.mturk.com/worker/acceptable-use-policy (accessed 21 May 2021).

Amazon Mechanical Turk Developer Guide (2021) Using the sandbox. Available at: https://docs.aws.amazon.com/AWSMechTurk/latest/AWSMechanicalTurkRequester/mturk-use-sandbox.html (accessed 9 May 2021).

amazon mturk requester (2021) Developer Sandbox. Available at: https://requester.mturk.com/developer/sandbox (accessed 9 May 2021).

Antonelli C and Crespi F (2013) The 'Matthew effect' in R&D public subsidies: The Italian evidence. *Technological Forecasting and Social Change* 80(8): 1523–1534. DOI: 10.1016/j.techfore.2013.03.008.

Araman VF, Calmon A and Fridgeirsdottir K (2019) Pricing and Job Allocation in Online Labor Platforms. INSEAD Working Paper.

Blesik T, Bick M and Kummer T-F (2021) A Conceptualisation of Crowd Knowledge. *Information Systems Frontiers*. DOI: 10.1007/s10796-021-10176-y.

Boudreau K (2010) Open platform strategies and innovation: Granting access vs. devolving control. *Management science* 56(10). INFORMS: 1849–1872.

Brawley AM and Pury CLS (2016) Work experiences on MTurk: Job satisfaction, turnover, and information sharing. *Computers in Human Behavior* 54: 531–546. DOI: 10.1016/j.chb.2015.08.031.

Chilton LB, Horton JJ, Miller RC, et al. (2010) Task search in a human computation market. In: *Proceedings of the ACM SIGKDD Workshop on Human Computation - HCOMP '10*, Washington DC, 2010, p. 1. ACM Press. DOI: 10.1145/1837885.1837889.

Cobham A, Schlögl L and Sumner A (2016) Inequality and the tails: the Palma proposition and ratio. *Global Policy* 7(1). Wiley Online Library: 25–36.

Connelly CE, Fieseler C, Černe M, et al. (2021) Working in the digitized economy: HRM theory & practice. *Human Resource Management Review* 31(1): 100762. DOI: 10.1016/j.hrmr.2020.100762.

Cowell F (2011) *Measuring Inequality*. 3rd ed. Oxford University Press.

Dannefer D (1987) Aging as intracohort differentiation: Accentuation, the Matthew effect, and the life course. In: *Sociological forum*, 1987, pp. 211–236. Springer.

Davis G and Kanopka K (2020) Complexity Theory: Amazon's Mechanical Turk is a disaster for crowdworkers. Available at: https://stanforddaily.com/2020/06/21/complexity-theory-amazons-mechanical-turk-is-a-disaster-for-crowdworkers/ (accessed 3 October 2022).

D'Cruz P and Noronha E (2016) Positives outweighing negatives: the experiences of Indian crowdsourced workers. *Work Organisation, Labour & Globalisation* 10(1). Pluto Journals: 44–63. DOI: 10.13169/workorgalaboglob.10.1.0044.

Deng X, Joshi KD and Galliers RD (2016) The duality of empowerment and marginalization in microtask crowdsourcing: Giving voice to the less powerful through value sensitive design. *Mis Quarterly* 40(2). Society for Information Management and The Management Information Systems …: 279–302.

Dennis SA, Goodson BM and Pearson C (2019) *Online Worker Fraud and Evolving Threats to the Integrity of MTurk Data: A Discussion of Virtual Private Servers and the Limitations of IP-Based Screening Procedures*. ID 3233954, SSRN Scholarly Paper, 14 March. Rochester, NY: Social Science Research Network. DOI: 10.2139/ssrn.3233954.

DonovanM (2018) *Turkmaster (Mturk)*. Available at: https://greasyfork.org/scripts/4771-turkmaster-mturk (accessed 9 March 2021).

Dror G, Koren Y, Maarek Y, et al. (2011) I want to answer; who has a question? Yahoo! answers recommender system. In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, San Diego, California, USA, 21 August 2011, pp. 1109–1117. KDD '11. Association for Computing Machinery. DOI: 10.1145/2020408.2020582.

Engert M, Evers J, Hein A, et al. (2022) The Engagement of Complementors and the Role of Platform Boundary Resources in e-Commerce Platform Ecosystems. *Information Systems Frontiers*. DOI: 10.1007/s10796-021-10236-3.

Fieseler C, Bucher E and Hoffmann CP (2019a) Unfairness by Design? The Perceived Fairness of Digital Labor on Crowdworking Platforms. *Journal of Business Ethics* 156(4): 987–1005. DOI: 10.1007/s10551-017-3607-2.

Fieseler C, Bucher E and Hoffmann CP (2019b) Unfairness by Design? The Perceived Fairness of Digital Labor on Crowdworking Platforms. *Journal of Business Ethics* 156(4): 987–1005. DOI: 10.1007/s10551-017-3607-2.

Fralich R and Bitektine A (2020) “Invincibles” and “invisibles”: CEO status and the ‘Matthew effect’in strategic decision-making. *Long Range Planning* 53(3). Elsevier: 101887.

Fredman S, Toit D du, Graham M, et al. (2020) Thinking Out of the Box: Fair Work for Platform Workers. *King’s Law Journal* 31(2). Routledge: 236–249. DOI: 10.1080/09615768.2020.1794196.

Gadiraju U, Checco A, Gupta N, et al. (2017) Modus Operandi of Crowd Workers: The Invisible Role of Microtask Work Environments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1(3): 1–29. DOI: 10.1145/3130914.

Gegenhuber T, Ellmer M and Schüßler E (2020a) Microphones, not megaphones: Functional crowdworker voice regimes on digital work platforms. *human relations*. SAGE Publications Sage UK: London, England.

Gegenhuber T, Ellmer M and Schüßler E (2020b) Microphones, not megaphones: Functional crowdworker voice regimes on digital work platforms. *Human Relations*. SAGE Publications Ltd: 0018726720915761. DOI: 10.1177/0018726720915761.

Geiger D and Schader M (2014) Personalized task recommendation in crowdsourcing information systems — Current state of the art. *Decision Support Systems* 65: 3–16. DOI: 10.1016/j.dss.2014.05.007.

Gerber C (2021) Community building on crowdwork platforms: Autonomy and control of online workers? *Competition & Change* 25(2). SAGE Publications Sage UK: London, England: 190–211.

Gol ES, Stein M-K and Avital M (2019) Crowdwork platform governance toward organizational value creation. *The Journal of Strategic Information Systems* 28(2). Elsevier: 175–195.

Greco GM and Floridi L (2004) The tragedy of the digital commons. *Ethics and Information Technology* 6(2). Springer: 73–81.

Hanrahan BV, Martin D, Willamowski J, et al. (2018) Investigating the Amazon Mechanical Turk Market Through Tool Design. *Computer Supported Cooperative Work (CSCW)* 27(3–6): 1255–1274. DOI: 10.1007/s10606-018-9312-6.

Hara K, Adams A, Milland K, et al. (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 - CHI ’18,* Montreal QC, Canada, 2018, pp. 1–14. ACM Press. DOI: 10.1145/3173574.3174023.

Hasan S (2018) *Mturk-Engine*. TypeScript. Available at: https://github.com/Anveio/mturk-engine (accessed 3 July 2021).

Hein A, Schreieck M, Riasanow T, et al. (2020) Digital platform ecosystems. *Electronic Markets* 30(1): 87–98. DOI: 10.1007/s12525-019-00377-4.

Hellman B (2021) *MTurk Suite*. JavaScript. Available at: https://github.com/Kadauchi/mturk-suite (accessed 21 May 2021).

Idowu A and Elbanna A (2020) Digital Platforms of Work and the Crafting of Career Path: The Crowdworkers’ Perspective. *Information Systems Frontiers*. DOI: 10.1007/s10796-020-10036-1.

Idowu A and Elbanna A (2022) Digital Platforms of Work and the Crafting of Career Path: The Crowdworkers’ Perspective. *Information Systems Frontiers* 24(2): 441–457. DOI: 10.1007/s10796-020-10036-1.

Irani LC and Silberman MS (2013) Turkopticon: interrupting worker invisibility in amazon mechanical turk. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI ’13*, Paris, France, 2013, p. 611. ACM Press. DOI: 10.1145/2470654.2470742.

Kaplan T, Saito S, Hara K, et al. (2018) Striving to Earn More: A Survey of Work Strategies and Tool Use Among Crowd Workers.: 9.

katnapped (2020) How to catch hits more quickly on hit finder? In: *r/mturk*. Reddit Post. Available at: www.reddit.com/r/mturk/comments/g9tt14/how_to_catch_hits_more_quickly_on_hit_finder/ (accessed 12 March 2021).

Kwek A (2020) Crowdsourced Research: Vulnerability, Autonomy, and Exploitation. *Ethics & human research* 42(1). Wiley Online Library: 22–35.

Malik F, Heeks R, Masiero S, et al. (2021) Digital labour platforms in Pakistan: institutional voids and solidarity networks. *Information Technology & People*. Emerald Publishing Limited. DOI: 10.1108/ITP-04-2020-0218.

Matherly T (2019) A panel for lemons? Positivity bias, reputation systems and data quality on MTurk. *European Journal of Marketing*. Emerald Publishing Limited.

McInnis B, Cosley D, Nam C, et al. (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 - CHI ’16*, Santa Clara, California, USA, 2016, pp. 2271–2282. ACM Press. DOI: 10.1145/2858036.2858539.

Merton RK (1968) The Matthew effect in science: The reward and communication systems of science are considered. *Science* 159(3810). American Association for the Advancement of Science: 56–63.

Merton RK (1988) The Matthew effect in science, II: Cumulative advantage and the symbolism of intellectual property. *isis* 79(4). Department of History and Science, University of Pennsylvania: 606–623.

Müller K, Vignaux T, Lünsdorf O, et al. (2021) *Simpy: Event Discrete, Process Based Simulation for Python.* Python. Available at: https://simpy.readthedocs.io (accessed 17 March 2021).

Newman A (2019) I found work on an Amazon website. I made 97 cents an hour. *The New York Times*.

O'Rand AM (1996) The precious and the precocious: Understanding cumulative disadvantage and cumulative advantage over the life course. *The Gerontologist* 36(2). The Gerontological Society of America: 230–238.

Panteli N, Rapti A and Scholarios D (2020) 'If He Just Knew Who We Were': Microworkers' Emerging Bonds of Attachment in a Fragmented Employment Relationship. *Work, Employment and Society* 34(3). SAGE Publications Ltd: 476–494. DOI: 10.1177/0950017019897872.

Ramirez J (2021) *PandaCrazy Max*. Available at: https://github.com/JohnnyRS/PandaCrazy-Max (accessed 21 May 2021).

Reschke BP, Azoulay P and Stuart TE (2018) Status Spillovers: The Effect of Status-conferring Prizes on the Allocation of Attention. *Administrative Science Quarterly* 63(4). SAGE Publications Inc: 819–847. DOI: 10.1177/0001839217731997.

Saito S, Chiang C-W, Savage S, et al. (2019) TurkScanner: Predicting the Hourly Wage of Microtasks. In: *The World Wide Web Conference*, San Francisco, CA, USA, 13 May 2019, pp. 3187–3193. WWW '19. Association for Computing Machinery. DOI: 10.1145/3308558.3313716.

Savage S, Chiang C, Saito S, et al. (2020) Becoming the Super Turker:Increasing Wages via a Strategy from High Earning Workers.: 12.

Schultz R (2020) *Stax*. Available at: https://gitlab.com/StructByLightning/stax (accessed 27 June 2021).

Seneta E (2013) A tricentenary history of the law of large numbers. *Bernoulli* 19(4). Bernoulli Society for Mathematical Statistics and Probability: 1088–1121.

Shen W, Hu YJ and Ulmer JR (2015) Competing for Attention: An empirical study of online reviewers' strategic behavior. *Mis Quarterly* 39(3). JSTOR: 683–696.

TurkerView mTurk Forum (2021) TurkerView's mTurk forum. Available at: https://forum.turkerview.com/ (accessed 15 August 2020).

Van Looy B, Ranga M, Callaert J, et al. (2004) Combining entrepreneurial and scientific performance in academia: towards a compounded and reciprocal Matthew-effect? *Research Policy* 33(3). Elsevier: 425–441.

Vancea DCP and Nemirschi A (2020) Challenges in Using AI in Online Commerce. (2): 5.

Waldkirch M, Bucher E, Schou PK, et al. (2021) Controlled by the algorithm, coached by the crowd–how HRM activities take shape on digital work platforms in the gig economy. *The International Journal of Human Resource Management*. Taylor & Francis: 1–36.

Watwani K (2021) *Turk Guru*. Available at: https://www.turk-guru.com (accessed 27 June 2021).

Wessel M, Thies F and Benlian A (2017) Opening the Floodgates: The Implications of Increasing Platform Openness in Crowdfunding. *Journal of Information Technology* 32(4): 344–360. DOI: 10.1057/s41265-017-0040-z.

Williams AC, Mark G, Milland K, et al. (2019) The Perpetual Work Life of Crowdworkers: How Tooling Practices Increase Fragmentation in Crowdwork. *Proceedings of the ACM on Human-Computer Interaction* 3(CSCW): 1–28. DOI: 10.1145/3359126.

Wood AJ, Graham M, Lehdonvirta V, et al. (2019) Good Gig, Bad Gig: Autonomy and Algorithmic Control in the Global Gig Economy. *Work, Employment and Society* 33(1): 56–75. DOI: 10.1177/0950017018785616.

Xie H, Checco A and Zamani ED (2021) Design Principles and a Conceptual Framework for Crowd Teamwork Systems. In: *25th Pacific Asia Conference on Information Systems (PACIS 2021)*, Dubai, UEA, 12 July 2021. Available at: https://aisel.aisnet.org/pacis2021/214.

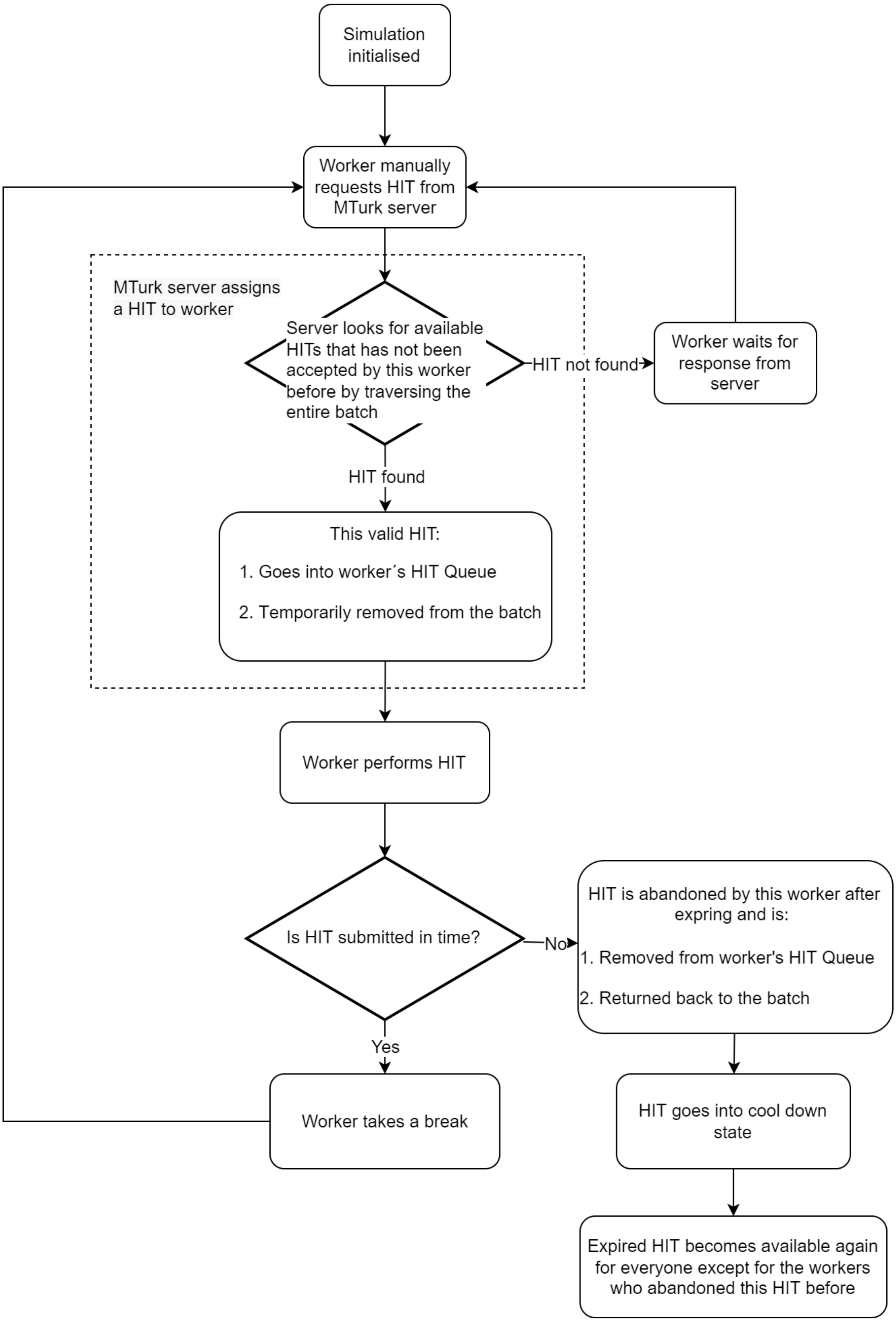
Yan J, Yu W and Zhao JL (2015) How signaling and search costs affect information asymmetry in P2P lending: the economics of big data. *Financial Innovation* 1(1): 19. DOI: 10.1186/s40854-015-0018-1.

Yuan S-TD and Hsieh C-F (2018) An impactful crowdsourcing intermediary design - a case of a service imagery crowdsourcing system. *Information Systems Frontiers* 20(4): 841–862. DOI: 10.1007/s10796-016-9700-8.

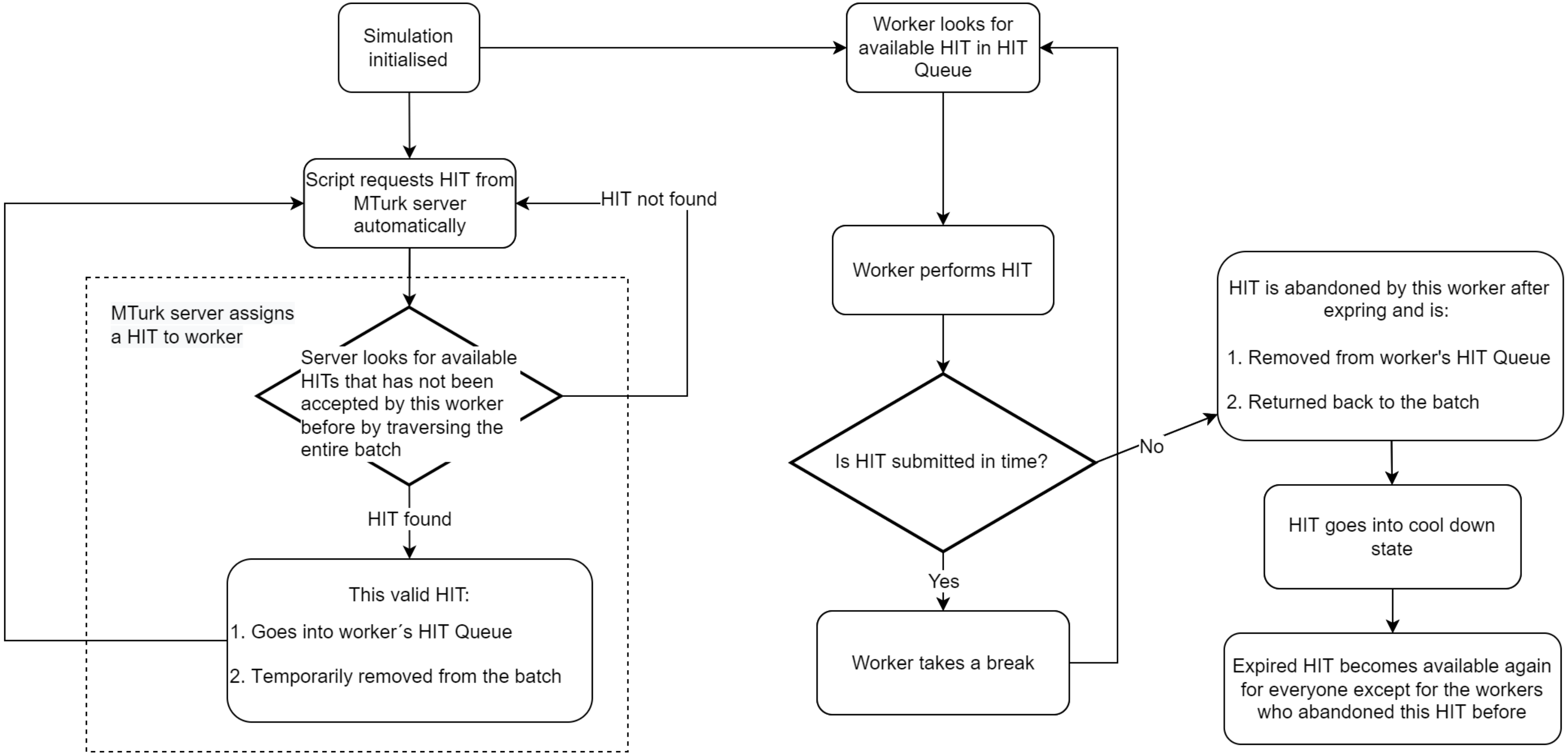
Zyskowski K and Milland K (2018) A Crowded Future: Working against Abstraction on Turker Nation. *Catalyst: Feminism, Theory, Technoscience* 4(2): 1–30. DOI: 10.28968/cftt.v4i2.29581.

Appendix

DES Workflow Diagram for Manual Workers


Simulation initialised
Worker manually requests HIT from MTurk server
MTurk server assigns a HIT to worker
Server looks for available HITs that has not been accepted by this worker before by traversing the entire batch
HIT not found
Worker waits for response from server
HIT found
This valid HIT:
1. Goes into worker´s HIT Queue
2. Temporarily removed from the batch
Worker performs HIT
Is HIT submitted in time?
No
HIT is abandoned by this worker after expring and is:
1. Removed from worker's HIT Queue
2. Returned back to the batch
Yes
Worker takes a break
HIT goes into cool down state
Expired HIT becomes available again for everyone except for the workers who abandoned this HIT before

DES Workflow Diagram for Script Workers


Simulation initialised
Worker looks for available HIT in HIT Queue
Script requests HIT from MTurk server automatically
HIT not found
MTurk server assigns a HIT to worker
Server looks for available HITs that has not been accepted by this worker before by traversing the entire batch
HIT found
This valid HIT:
1. Goes into worker´s HIT Queue
2. Temporarily removed from the batch
Worker performs HIT
Is HIT submitted in time?
No
Yes
Worker takes a break
HIT is abandoned by this worker after expring and is:
1. Removed from worker's HIT Queue
2. Returned back to the batch
HIT goes into cool down state
Expired HIT becomes available again for everyone except for the workers who abandoned this HIT before