



This is a repository copy of *The unintended consequences of automated scripts in crowdwork platforms: a simulation study in MTurk.*

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/195506/>

Version: Accepted Version

---

**Article:**

Xie, H., Checco, A. and Zamani, E. [orcid.org/0000-0003-3110-7495](https://orcid.org/0000-0003-3110-7495) (2023) The unintended consequences of automated scripts in crowdwork platforms: a simulation study in MTurk. *Information Systems Frontiers*. ISSN 1387-3326

<https://doi.org/10.1007/s10796-023-10373-x>

---

This version of the article has been accepted for publication, after peer review (when applicable) and is subject to Springer Nature's AM terms of use, but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: <http://dx.doi.org/10.1007/s10796-023-10373-x>

**Reuse**

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

**Takedown**

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing [eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk) including the URL of the record and the reason for the withdrawal request.



[eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk)  
<https://eprints.whiterose.ac.uk/>

[Click here to view linked References](#)

# 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).

1 While such scripts are becoming indispensable, it is unclear what is their impact on  
2 the crowdworkers that do not use them. For example, there is evidence to suggest that  
3 the use of such scripts may pose an entry barrier for novice crowdworkers and result  
4 in fewer opportunities (Davis and Kanopka, 2020). It is quite possible that due to  
5 frustration and/or inability to secure quality tasks, these crowdworkers may exit the  
6 platform altogether, which can have negative implications for the health of later.  
7 Against this backdrop, in this study we examine the possible unintended  
8 consequences of the widespread use of microtask catching scripts by examining  
9 different scenarios using discrete event simulation with parameters modelled on real-  
10 world measurements. We find that, under certain conditions: a) crowdworkers who  
11 use such scripts trigger a tragedy of the commons phenomenon (Greco and Floridi,  
12 2004), whereby they reserve too many tasks, which they fail to complete on time. As  
13 such, many of these microtasks eventually expire, negatively impacting the job  
14 requester and the platform; b) those crowdworkers who don't use microtask catching  
15 scripts are unable to catch any of the better microtasks because these become reserved  
16 immediately by scripts; c) those who use scripts increase their reputation and gain a  
17 higher skilled status on the platform at the expense of those who don't use scripts,  
18 giving rise to the Matthew effect (Merton, 1988).  
19  
20  
21  
22  
23

24 The contributions of this study enrich the current discourse on crowdwork literature in  
25 several ways. First, while the crowdwork literature has acknowledged the importance  
26 of using tools for the purpose of increasing hourly income (Kaplan et al., 2018), the  
27 broader implications and the unintended consequences on crowdworkers' working  
28 conditions, e.g., opportunities, earnings per time unit, including task quality and task  
29 completion efficiency, remain relatively unknown. In our study, by simulating  
30 different scenarios, we quantify these in order to explicate what the impacts are for  
31 the workers and the platform, examining the effects on workers' income and the  
32 microtasks. Second, we provide a set of recommendations for script designers and  
33 platforms using simulations in an effort to reduce the impacts of the tragedy of the  
34 commons effect and to improve batch diversity. Previous studies have proposed  
35 ranking mechanisms, based on the premise that platforms will improve their operation  
36 standards to become more appealing to their clients (Fredman et al., 2020). We  
37 believe that such attempts overestimate employers' sensitivity regarding working  
38 conditions, especially because platforms have triggered a race to the bottom for  
39 cheaper labour (Altenried, 2020). Instead, we detail the ways in which the health of a  
40 platform can be in jeopardy due to the current design of microtask catching scripts,  
41 whereby crowdworkers become disillusioned and quit the platform entirely. This we  
42 believe is a greater incentive for implementing changes in the way scripts currently  
43 operate, mitigating their negative impacts, facilitating the healthy growth of the  
44 platform, and indirectly improving the working conditions of crowdworkers.  
45  
46  
47  
48  
49  
50  
51

52 We can summarise our research questions as follows:

53  
54 RQ1: To what extent the use of automated scripts affects worker diversity, hourly  
55 wages, and equity in the context of crowdwork?  
56

57  
58 RQ2: How can we theorise about the use of automated scripts by crowdworkers and  
59 their consequences?  
60  
61  
62  
63  
64  
65

1 The article is structured as follows. First, we present the theoretical background of our  
2 study and related work. Then, we described our research method and the data we used.  
3 In the fourth section, we report our results. Then, we offer a discussion based on our  
4 major findings in relation to the existing literature and draw the study's major  
5 contributions in relation to research and practice, and the article closes by discussing  
6 the limitations and avenues for further research.  
7

## 10 **2. Background**

### 13 **2.1 Crowdwork and microtask catching scripts**

14 Within a crowdsourcing platform, such as Amazon Mechanical Turk (MTurk), job  
15 requesters publish microtasks through an open call to an undefined number of workers  
16 (Blesik et al., 2021). These microtasks are called Human Intelligence Tasks (HITs),  
17 and typically are easier for humans to solve rather than for computers, but require the  
18 participation of the crowd due to their large volume. Typical microtasks or HITs  
19 include market research questionnaires for a particular industry, or requests for  
20 participants to transcribe text from an audio recording (Gadiraju et al., 2014). In the  
21 case of MTurk, for example, the crowdworkers involved in a HIT usually spend a few  
22 minutes to a few hours on HIT completion by submitting their survey responses or  
23 data outputs, as requested by the job requester who posted the HITs. The requester  
24 ultimately pays the crowdworker with a monetary reward, depending on whether the  
25 quality of the output is acceptable or not. Quite often, novice crowdworkers have to  
26 complete a large number of HITs in order to increase their number of completed HITs  
27 and improve their HIT approval rate (Hara et al., 2018). The HIT approval rate is a  
28 metric of an individual's quality of HIT completion over time. It is commonly used by  
29 job requesters as a filter when choosing their target crowdworkers, and therefore as a  
30 metric it often determines whether or not crowdworkers are eligible to receive high-  
31 rewarded quality tasks (Savage et al., 2020; Kaplan et al., 2018).  
32  
33

34 Besides the time spent on executing a HIT, a crowdworker spends time in searching  
35 and identifying relevant HITs, reading instructions in relation to their execution, and  
36 potentially learning how to complete them, including learning how to interact with the  
37 HIT user interface in many cases. This extra time is not paid for, and often it exceeds  
38 the time dedicated to executing the chosen HIT (Chilton et al., 2010). To address this  
39 issue, crowdworkers turn to the use of HIT automated scripts (henceforth referred to  
40 as catching scripts), such as Panda Crazy and MTurk Suite (Irani and Silberman, 2013;  
41 Ramirez, 2021; *TurkerView mTurk Forum*, 2021). These scripts allow crowdworkers  
42 to 'catch' HITs posted by specific job requesters (based on their ID number), with  
43 high ratings or based on the specificities of the HIT (Saito et al., 2019), whereby the  
44 requirements of the HIT fit their prior experience or personal interests (Dror et al.,  
45 2011; Geiger and Schader, 2014). In other words, these scripts support selective and  
46 automatic catching of HITs based on personal preferences.  
47  
48

49 In this study, scripts specifically refer to the tool, bots, or extensions that allow the  
50 workers to accept HITs automatically. These scripts allow them to customise the  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

1 frequency and the number of caught HITs and create batch IDs (group of HITs) so as  
2 to continuously receive HITs that fit predefined criteria (DonovanM, 2018; Hellman,  
3 2021). In short, such advanced HIT catching scripts provide crowdworkers with extra  
4 opportunities to locate and secure their preferred HITs. For some crowdworkers, who  
5 choose to participate in community forums, such as Turker Nation in Slack, these  
6 opportunities are further enhanced (Zyskowski and Milland, 2018). These community  
7 forums bring crowdworkers and job requesters together, who can exchange  
8 information on HITs outside the boundaries of the platform. On the one hand, such  
9 forums allow participating crowdworkers to access additional information by directly  
10 contacting the job requesters and to identify higher quality tasks before others. On the  
11 other hand, job requesters may access crowdworkers through these forums and  
12 directly allocate them HITs, bypassing the platform's competitive process.  
13  
14  
15  
16  
17

## 18 **2.2 Previous work on catching scripts**

19  
20

21 The use of catching scripts is essentially a technical simulation of human behaviour.  
22 Fundamentally, the script allows individuals to expedite the process of identifying and  
23 securing extremely limited items in an online platform, particularly when the demand  
24 for these items is high and exceeds the supply (Vancea and Nemirschi, 2020). These  
25 items may range from limited edition sneakers and other apparel to concert tickets,  
26 which are then resold by the resellers to secondary markets (e.g., Craigslist, eBay and  
27 social networking sites, such as Twitter) for a profit.  
28  
29  
30

31 While the impact of the use of HIT catching script has been extensively researched in  
32 the online retailing area, it is still in its infancy in the crowdwork domain. To date, the  
33 literature has explored the use of catching scripts as a means to overcome platforms'  
34 poor search functionality for the identification of HITs (El Maarry et al., 2018;  
35 Kaplan et al., 2018). In addition, scholars have developed classifications of these  
36 scripts to support further development of such tools (Williams et al., 2019; El Maarry  
37 et al., 2018). There are, however, studies that have focused on the impacts of these  
38 scripts on crowdworkers' workflow and daily life, whereby it has been found that the  
39 use of these scripts can cause disruptions, interfere with everyday life (Williams et al.,  
40 2019), and lead to tasks being abandoned as crowdworkers become overwhelmed  
41 with too many HITs being automatically accepted (Williams et al., 2019). However,  
42 the use of such scripts can have positive effects as well as they allow crowdworkers to  
43 take advantage of all available opportunities to take on highly rewarding HITs  
44 (Williams et al., 2019). More importantly, the use of HIT catching script reduces the  
45 amount of unpaid time crowdworkers spend on HIT search and increases access to  
46 quality HITs (Kaplan et al., 2018). To date, however, research on crowdwork has not  
47 quantified the possible negative implications of catching scripts on crowdwork.  
48  
49  
50  
51  
52  
53

54 Within the crowdwork domain, crowdworkers use these catching scripts to capture  
55 HITs that align with their own interests and skills. This enables them to maximise  
56 their income while reducing the time spent on familiarising themselves with HITs'  
57 guidelines and interfaces, which typically results in unpaid labour. Experienced  
58 crowdworkers may earn up to \$12 per hour with the help of such scripts (Newman,  
59  
60  
61  
62  
63  
64  
65

1 2019), because they have a better chance of securing higher paying HITs, and in turn  
2 increasing their overall ranking in the platform once they complete them. A study has  
3 shown that high-paying HITs with high reputation scores are reserved within seconds  
4 from being published (Hanrahan et al., 2018). Therefore, crowdworkers who use  
5 catching scripts have a clear advantage over those who don't, making it difficult for  
6 less experienced crowdworkers to identify HITs with even moderate rewards.  
7 Progressively, this has resulted in high frequency of crowdworkers being frustrated  
8 and leaving the platform altogether, which is detrimental for its healthy growth  
9 (Hanrahan et al., 2018). This can be concerning because reduced participation of new  
10 crowdworkers could have a negative impact on the diversity of HIT responses. In  
11 terms of short-term impacts, high-quality HITs tend to be captured by workers with  
12 more effective HIT catching tools, meaning that, when there are no limits as to how  
13 many HITs a worker may undertake, highly rewarding HITs in theory may be  
14 completed by a single worker. In other words, the diversity of responses decreases  
15 (Hanrahan et al., 2018). In addition, many high quality HITs may be captured by  
16 automated scripts and sit idle in a few crowdworkers' queues, until they are  
17 completed or until they expire because the crowdworker is unable to complete all  
18 those they have captured within the time allotted by the job requester. In the latter  
19 case, the completion time of the entire batch gets delayed.  
20  
21  
22  
23  
24

25 What the above indicates is that, while the original intent of using automated scripts  
26 for catching HITs was to increase crowdworkers' efficiency, their excessive use can  
27 have destabilising effects for the ecosystem, inhibiting the healthy growth of the  
28 platform, and negatively impacting the crowdworkers who do not use them. In what  
29 follows, we explore why platforms may remain open to the use of such automated  
30 scripts despite the potential negative impacts on their ecosystem and what measures  
31 they take in order to control against excessive use.  
32  
33  
34  
35  
36  
37

### 38 **2.3 Why do platforms allow the use of catching scripts?**

39  
40 Crowdwork platforms, like most digital platforms, are in essence marketplaces where  
41 buyers and sellers can meet and exchange services for an agreed fee (Giaglis et al.,  
42 2002). Similarly, to other digital platforms, in order to enhance the platform's  
43 functionality and therefore its overall competitiveness, platforms provide  
44 complementors, also known as third parties, with access to the platform, who create  
45 plug-ins, add-ons and other extensions (Wessel et al., 2017). This provides sufficient  
46 autonomy to third parties and encourages complementary innovations (Boudreau,  
47 2010; Hein et al., 2020).  
48  
49  
50

51 In platforms, the use of automated microtask catching scripts is a reflection of  
52 platforms' openness to such complementors, where third parties are able to develop  
53 their scripts in compliance with the platform's regulations. These platforms explicitly  
54 allow the use of automated scripts so that crowdworkers are better able to search for  
55 and preview microtasks (HITs). This enhanced functionality makes platforms more  
56 attractive to crowdworkers by allowing them to optimise the workflow of searching,  
57 identifying and reserving preferred and high-rewarding HITs in batches.  
58  
59  
60  
61  
62  
63  
64  
65

1 However, the challenges observed in other platforms (e.g., crowdfunding and open-  
2 source platforms) are also present. Wessel et al. (2017), for example, discuss that a  
3 major challenge is identifying the balance between platform openness towards third  
4 parties and maintaining control. Being too open can potentially destabilise the  
5 ecosystem. In the case of crowdwork, the stability of the ecosystem extends beyond  
6 retaining oversight of operations; openness needs to be balanced against the need for  
7 ensuring the fair treatment of crowdworkers and providing job requesters with high  
8 quality outputs, both of which feed into the healthy growth of the ecosystem. Amazon  
9 Mechanical Turk, for example, prohibits the use of scripts that send requests at an  
10 excessively high frequency and those that automatically accept HITs (*Amazon*  
11 *Mechanical Turk*, 2018). It is unclear, however, whether this ban is for the purpose of  
12 maintaining the operational stability of servers, for ensuring the fair treatment of all  
13 crowdworkers or for satisfying diversity in the data collected via HITs. In reality,  
14 while the imposed limitation on the frequency supports operational stability, it is  
15 unknown whether it can ensure fairness across those who use automated scripts and  
16 those who don't.  
17  
18  
19  
20  
21  
22  
23

### 24 **3. Theoretical Framing of the Study**

27 Within a crowdwork environment, crowdworkers are described by their approval  
28 rating, which indicates the number of successfully completed HITs. Once a  
29 crowdworker completes a HIT, the job requester will examine the output and if it is  
30 satisfactory, they will approve it and reward the worker, and the worker's approval  
31 rating will increase; if the output is not approved, the worker is not paid and their  
32 approval rating will decrease. In addition, many HITs require a minimum approval  
33 rating. This means that there are cases whereby crowdworkers with low approval  
34 ratings, even due to being newcomers to the platform, are prohibited by design from  
35 accepting the said HITs (Brawley and Pury, 2016).  
36  
37  
38

39 This approval system bears resemblance to the typical ranking and reward systems  
40 observed elsewhere (e.g., online marketplaces, h-index), whereby the ranking of  
41 individuals (e.g., merchants, researchers) is said to reflect expertise and mastery in  
42 particular types of HITs (Matherly, 2019). However, studies have shown that such  
43 systems are prone to bias, whereby rankings may push individuals to adopt strategic  
44 behaviours that do not necessarily support the flourishing of the ecosystem (Shen et  
45 al., 2015), and may lead to unfair treatment (Idowu and Elbanna, 2020)  
46  
47  
48

49 In this study, we are interested in exploring, but crucially, quantifying the impacts of  
50 automated catching scripts as materialised through the use of such ranking systems.  
51 We thus frame our empirical study drawing from Merton's Law of Unintended  
52 Consequences, which is also known as the Matthew effect (Merton, 1968). The  
53 Matthew Effect suggests that those who enjoy greater visibility receive greater  
54 rewards, and that those who are less visible, they receive disproportionately lower  
55 rewards and less recognition for the same performance. The Matthew effect is well  
56 documented and recognised in areas such as scientometrics and sociology for the  
57 examination of hierarchical systems (Fralich and Bitektine, 2020). However, much  
58  
59  
60  
61  
62  
63  
64  
65



1 less is known regarding the extent to which the Matthew effect promotes or restricts  
2 equal opportunities for participation and reward within an environment governed by  
3 the presence of automated scripts.  
4  
5

### 6 **3.1 The Matthew effect: competence and reputational** 7 **persistence** 8 9

10 When discussing the reward system for scientific contributions, Merton argued that  
11 society tends to honour those with greater reputation and visibility, irrespective of the  
12 degree of their contribution to a particular piece of work (Merton, 1968). Similarly,  
13 applications for research and development grants put forward by less known consortia  
14 and companies are often denied funding, because decisions tend to be influenced by  
15 the candidates' award history (Van Looy et al., 2004). In both cases, those who are  
16 less visible and newcomers are disadvantaged, without this indicating, however, that  
17 they are any less capable. Rather, such disadvantages and exclusions are the result of  
18 unfounded perceptions regarding one's competence and reputation (Antonelli and  
19 Crespi, 2013; Dannefer, 1987; O'Rand, 1996).  
20  
21  
22  
23  
24

25 Antonelli and Crespi (2013) have argued that competence persistence is an expression  
26 of virtuous Matthew effects, whereby the resources at one's disposal allow them to  
27 increase their overall competence, and therefore their outputs, such as income, are  
28 simply higher.  
29  
30

31 In the case of crowdwork, automated scripts allow crowdworkers to secure microtasks  
32 that meet certain criteria with a frequency and duration that far exceeds that of a  
33 manual workflow. Crowdworkers who leverage these scripts can gain a higher skilled  
34 status compared to their peers, gaining a competitive edge. This status is often  
35 achieved at the expense of crowdworkers with lower approval ratings or less skilled in  
36 using scripts (Reschke et al., 2018). More importantly, the use of automated scripts  
37 influences one's income. Super Turkers, those who use multiple scripts together to  
38 access high quality HIT referral channels, gain extremely high earnings in comparison  
39 (Savage et al., 2020). Because high rewarding HITs are finite and scarce, scripts  
40 enable Super Turkers to locate them and reserve them in a matter of seconds when  
41 they do become available.  
42  
43  
44  
45

46 This is an example of how competence persistence develops and endures. Super  
47 Turkers invest more time and effort in using automated scripts for HIT filtering  
48 (Kaplan et al., 2018), whereby HIT filtering allows them to identify HITs that match  
49 their skills and competencies, which they can tackle successfully, on time and receive  
50 the relevant reward. Those who do not or cannot use such scripts have fewer  
51 opportunities to identify enough HITs relevant to their skills, and complete fewer and  
52 fewer HITs overall. Worse still, because of fewer opportunities, crowdworkers often  
53 opt to complete lower quality HITs, posted by less (or non) reputable job requesters to  
54 increase their overall income. This is an additional risk for crowdworkers, because  
55 less reputable job requesters often reject without an explanation the submitted output,  
56 in which case the worker is not paid and the approval rating gets further reduced  
57  
58  
59  
60  
61  
62  
63  
64  
65

(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

1 potentially driving them out for crowdwork altogether. As such, reputational  
2 persistence gives way to further unintentional consequences.  
3  
4

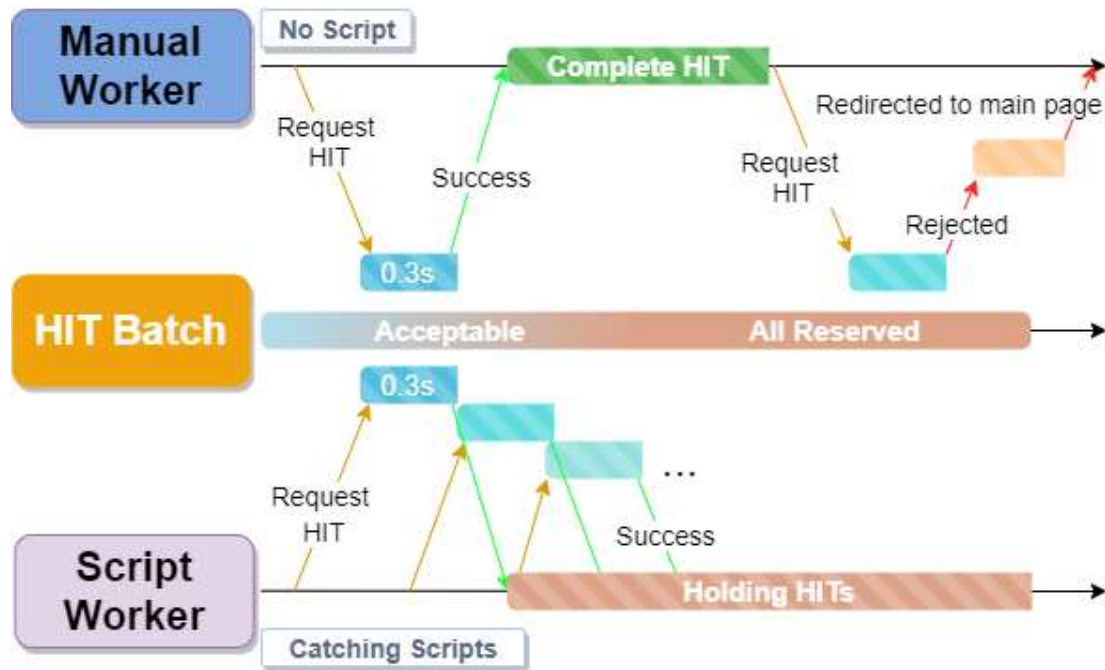
## 5 **4. Methodology**

### 6 **4.1 Rational of the simulation framework**

7  
8  
9  
10  
11 In this study, we are interested in exploring the unintended consequences stemming  
12 from the use of HITs catching scripts, and specifically quantifying the effects of  
13 technical competence. Our hypothesis is that the success rate of catching HITs for  
14 each crowdworker will be influenced by the persistence of their technical skills under  
15 the condition that everyone has the same access to these HITs.  
16  
17

18  
19 We investigate the above using data collected through a series of simulations. Our  
20 simulation framework has been designed with the view to conduct controlled variable  
21 experiments and simulate different HIT acceptance behaviours of crowdworkers to  
22 achieve more in-depth insights. More specifically, this simulation framework allows  
23 us to quantify the advantage that scripts-using crowdworkers have over non-script-  
24 using crowdworkers when accepting HITs from the same batch and the impact of this  
25 on the completion of the whole batch.  
26  
27

28  
29 In the proposed simulations, scripts-using crowdworkers consistently fill their HIT  
30 queues without considering their ability to complete all accepted HITs in a timely  
31 manner. As such, the question we wish to answer is whether scripts-using  
32 crowdworkers cost non-script-using crowdworkers their rightful work opportunities  
33 (as depicted in Figure 1). In turn, we wish to explore the impact of this on the final  
34 number of completed HITs. A possible scenario is that when there is only one batch  
35 of HITs available on the platform and all HITs are captured by scripts, non-script-  
36 using crowdworkers will remain idle until a new HIT gets published (Hanrahan et al.,  
37 2018).  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65



**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

1 implications for non-script-using crowdworkers. In addition, the simulation allows us  
2 to perform multiple iterations and adjust the parameters of the experiment with  
3 extremely low cost, thus helping us to identify additional phenomena worth further  
4 exploration. Regarding the ethical issues, our experimental data derive from  
5 simulations through the MTurk Sandbox requester where we created multiple  
6 crowdworker accounts (owned by the researchers), as such there was no collection of  
7 live data and no crowdworkers were involved in our simulation.  
8  
9

### 10 11 **4.3 Data collection**

12 Before the development of the simulation framework, the durations between different  
13 key events throughout the life cycle of the HITs (e.g. time for a HIT to become visible,  
14 time for an expired HIT to be published again etc.) were estimated on the MTurk  
15 Developer Sandbox (*amazon mturk requester*, 2021).  
16  
17

18 The MTurk Developer Sandbox is a mirror of the production platform which allows  
19 requesters to test their microtasks before publishing on the production site (*Amazon*  
20 *Mechanical Turk Developer Guide*, 2021). Other than the monetary transfer being  
21 disabled, the MTurk Developer Sandbox has the same functionalities as the MTurk  
22 platform.  
23  
24  
25  
26  
27

### 28 29 **4.4 Simulation Design**

30 Apart from the aforementioned time parameters, in order to initialise the DES we set  
31 up (i) the percentage of crowdworkers using scripts, (ii) the number of crowdworkers  
32 per batch, (iii) the crowdworker expected HIT completion time, and (iv) the  
33 crowdworker acceptance strategy.  
34  
35  
36  
37

38 Regarding (i), no studies have yet been conducted to estimate the number of people  
39 using scripts on MTurk. Moreover, this study aims to construct a simulation  
40 framework to quantify the unintended consequences of using catching scripts at a  
41 relatively micro level. Therefore, the total number of crowdworkers (ii) is limited to a  
42 maximum of 50. For (iii), each HIT can be reserved by one crowdworker for up to 1  
43 minute and usually takes half a minute to complete. But the exact time that one  
44 crowdworker spends on completing one HIT is a random number within a truncated  
45 exponential distribution with the mean value of 30 seconds (Araman et al., 2019).  
46  
47

48 Regarding (iv), a non-script-using crowdworker would normally start working on a  
49 batch by accepting a HIT from it. Then, they would accept a new HIT after the  
50 submission of the previous one. Script-using crowdworkers, on the other hand, will  
51 use scripts to continuously accept as many HITs as possible while attempting to  
52 complete the HITs with enough time remaining from their HIT queue. We do not  
53 limit the number of HITs that can be completed by each crowdworker. Both strategies  
54 are further explained via the DES workflow diagrams in Appendix 8.1 and 8.2.  
55  
56  
57

58 Two groups of simulations were designed to investigate the variation in unintended  
59 consequences of script use on both types of crowdworkers and the whole batch at  
60  
61  
62  
63  
64  
65

1 different experimental sizes and percentages of script-using crowdworkers. The  
2 results of the DES depend on several random effects, such as whether crowdworkers  
3 have available HITs to perform all the time, so multiple runs need to be conducted to  
4 estimate the expected scores (Seneta, 2013). In this study, 10 runs for each simulation  
5 were sufficient since the results clearly reveal the trend of variation and the boxplots  
6 are not overlapping.  
7  
8  
9

#### 10 **4.4.1 Script impact in different experimental scales**

11 The first group of simulations obtains data by gradually increasing the size of the  
12 experiment, which is increasing the number of HITs included in a batch and the  
13 number of workers by the same proportion (10:1), while controlling for the proportion  
14 of workers of both types and the ratio of the number of workers to the number of  
15 HITs. The aim is to explore how the unintended consequences of the use of the script  
16 on the workers and the HITs themselves change as the size of the experiment  
17 increases, e.g., whether the gap between the job opportunities of both types of  
18 workers increases further or decreases.  
19  
20  
21  
22  
23  
24

#### 25 **4.4.2 Script impact in different percentages of script workers**

26 The use of scripts gives workers a technical advantage, but what would be the impact  
27 on batch completion if there are more workers using scripts? To answer this question,  
28 simulations were conducted by increasing the percentage of script workers from 0%  
29 to 90% by maintaining a total of 500 HITs and 50 workers.  
30  
31  
32  
33  
34

### 35 **4.5 Metrics**

36 Our analysis focuses on two sets of measurements related to the whole batch and the  
37 workers, which contain the diversity of data on the final completion of the batch, the  
38 total completion time, the number of HIT acceptance and submission for both types of  
39 workers in each simulation. In particular, the equation for the batch diversity is as  
40 follows:  
41  
42  
43  
44

$$45 \left[ \text{Diversity} = 1 - \frac{\sum_{i=0}^n (2i - n - 1)x_i}{n \sum_{i=0}^n x_i} \right] \quad (1)$$

46 Equation (1) is derived from the equation for Gini coefficient of inequality, which is  
47 widely used in the field of economics (Cowell, 2011).  $x_i$  stands for the array in  
48 ascending order that stores the numbers of finished HITs for each worker. 1-Gini  
49 Coefficient is applied to represent the overall equity of opportunities on doing HITs  
50 for each worker and the batch diversity. The larger the ratio, the more fairness of  
51 catching HITs for each worker, and the higher diversity of the batch results since they  
52 come from a wider range of workers.  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

## 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.

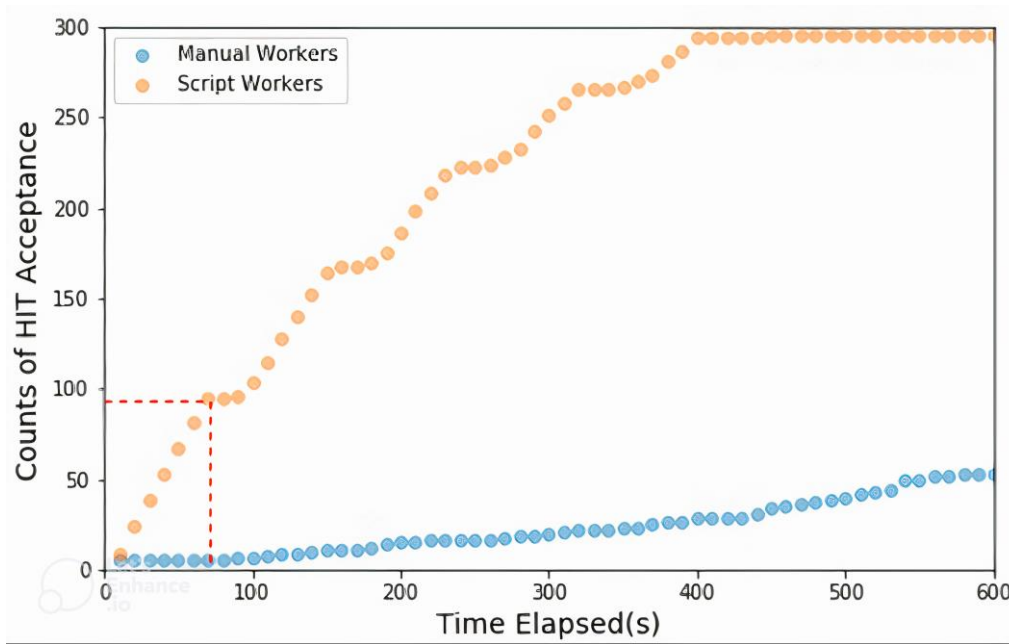
As can be seen in Table 1, the negative impact of scripting is gradually magnified as the size increases. Fairness of catching HITs also decreases from around 0.88 to 0.65. The average number of HITs completed per non-script-using worker also dropped from 4.8 to 2.1. In other words, as the batch size 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-using workers, 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.



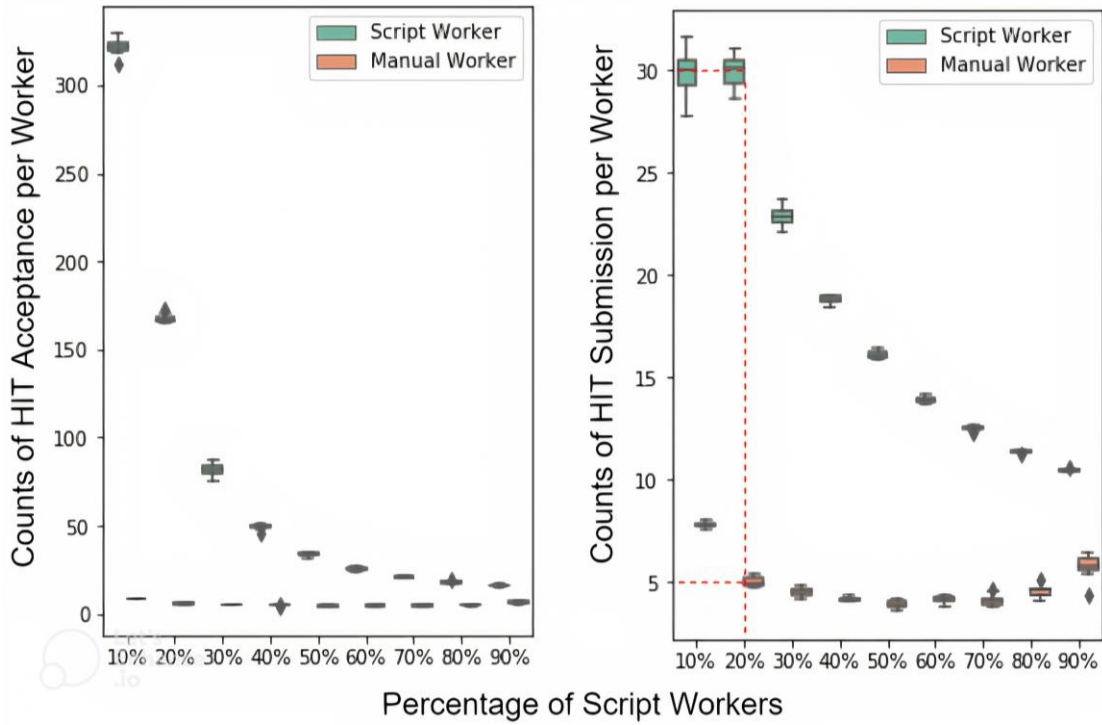
## 5.2 Script impact vs. percentage of script-using workers

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).

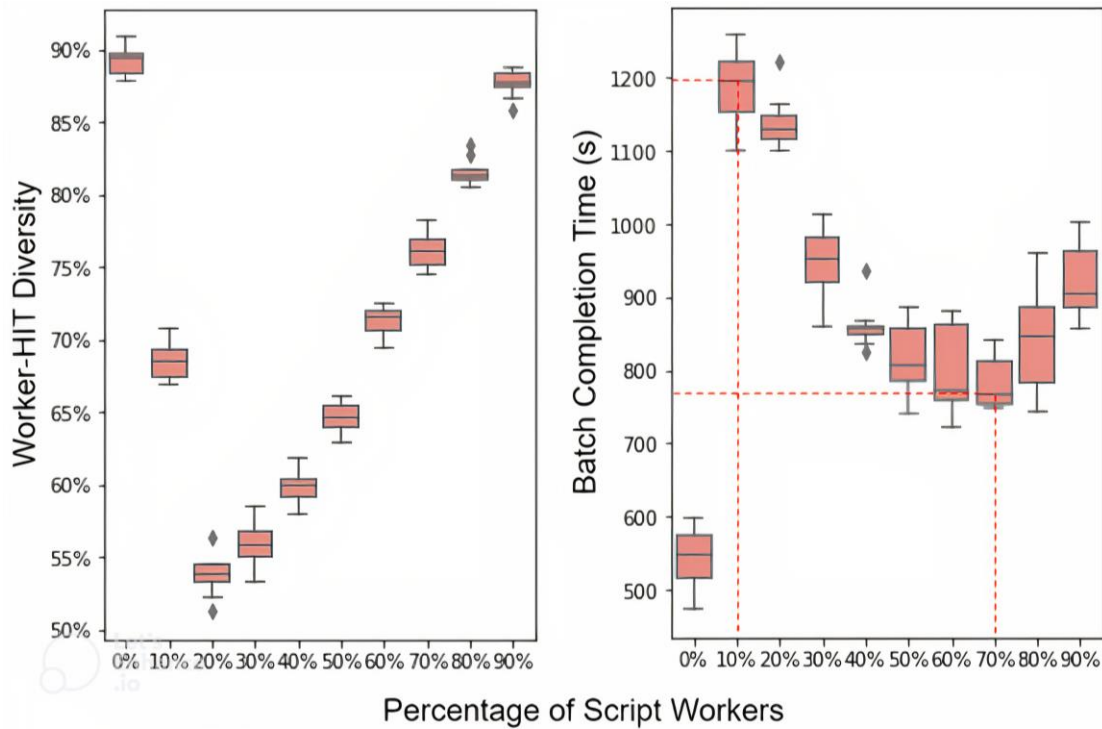
Interestingly, however, as more workers use scripts for automatic HIT acceptance, the diversity of data gradually returns. When 90% of all workers use scripts, batch diversity returns to its initial level (88%), the same as it would have been with all non-script-using workers (89%). The increase in diversity 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.



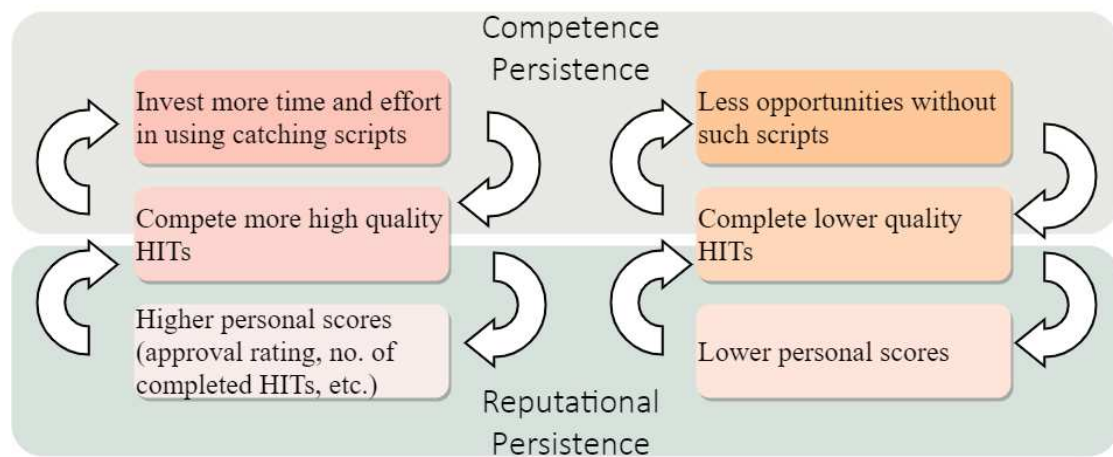
**Figure 3** Worker related statistics under different percentages of script workers (batch size = 500)



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

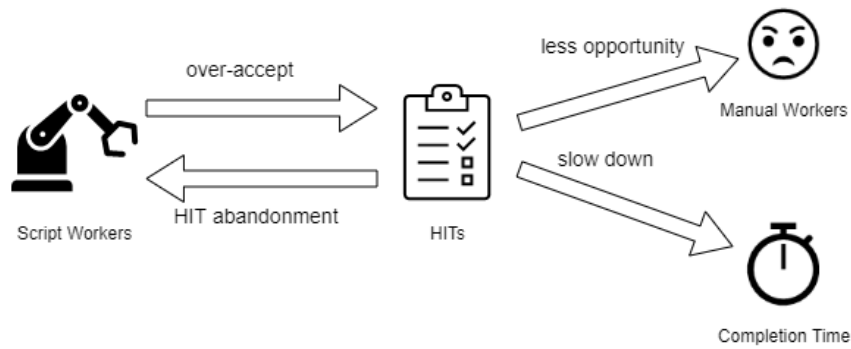
## 6. Discussion

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.



**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.

1 Second, our study enriches the growing body of literature on the impact of  
2 algorithmic control and working conditions. Gol et al. (2019) argue that the reputation  
3 system that feeds on HIT completion and approval is a good estimate for future  
4 performance, allows job requesters to verify crowdworkers' credentials and supports  
5 platforms to exercise the appropriate level of control for governance purposes.  
6 However, Wood et al. (2019) expressed concerns with regards to the consequences of  
7 such algorithmic control, where job requesters are able to identify 'quality' workers  
8 on the basis of the number of HITs completed. Indeed, job requesters tend to set very  
9 high acceptance criteria for the HITs they publish to filter out less experienced  
10 workers from the large labour pool as quickly as possible (Waldkirch et al., 2021).  
11 Our findings show that the reputation system that takes into account HIT completion  
12 and approval rates is susceptible to the vicious impacts of the Matthew effect, where  
13 the extensive use of HIT catching scripts have adverse impacts on the opportunities  
14 allowed for non-script-using workers and new workers in general. New workers in  
15 particular are required to complete a high number of HITs as quickly as possible to  
16 attain an acceptable score in order to be later considered for higher quality HITs. In  
17 other words, they may find themselves completing a large volume of low quality HITs,  
18 which are low rewarding and/or posted by less reputable job requesters (Savage et al.,  
19 2020), and thus risk completing HITs that may not be approved and thus not be  
20 rewarded (Kwek, 2020). Workers who are not proficient in scripting, or whose scripts  
21 are less functional, indeed have a more difficult time obtaining quality HITs  
22 (Williams et al., 2019). As such, our study suggests that beyond the impacts of  
23 platform design on the labour conditions of crowdworkers, the behaviour of peers, i.e.,  
24 other crowdworkers can have equally damaging effects.  
25  
26  
27  
28  
29  
30  
31

32 Our findings have further implications for research that delve into algorithmic control.  
33 We show that algorithmic control over time, has a negative impact on the platform, as  
34 well. Newcomers to the platform become discouraged due to the indirect obstacles  
35 imposed by design (Brawley and Pury, 2016), and those who do not use HIT catching  
36 scripts, are more likely to abandon the platform altogether. Previous studies have  
37 underlined that high churn rate is indeed a threat for platforms because job requesters  
38 may not be able to obtain high quality results for low enough costs (e.g., Deng et al.  
39 2016). In short, Matthew effects, not only ultimately force newcomers and non-script-  
40 using workers to leave the platform, but can potentially lead to the collapse of the  
41 platform itself.  
42  
43  
44  
45

46 As a result, we posit that our findings can have implications for the crowdwork  
47 literature. Studies that examine the labour conditions in the crowdwork domain and  
48 the implications of these on crowdworkers, tend to draw from qualitative data (e.g.,  
49 Gegenhuber et al., 2020b; Idowu and Elbanna, 2022; Panteli et al., 2020). In addition,  
50 while there is great attention to the crowdwork ecosystem, existing scholarship in this  
51 area seems more focused on identifying design-based sources of bias (e.g., Fieseler et  
52 al., 2019b) or crafting design principles for enhancing the efficiency and functionality  
53 of such platforms (e.g., Xie et al., 2021; Yuan and Hsieh, 2018). While the findings of  
54 such studies are significant in order to identify issues pertaining to crowdwork and to  
55 improve the underlying conditions, we note that in order to capture the full extent of  
56 the negative implications, researchers need to be able to quantify these in order to  
57  
58  
59  
60  
61  
62  
63  
64  
65

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.



- 1 Cobham A, Schlögl L and Sumner A (2016) Inequality and the tails: the Palma  
2 proposition and ratio. *Global Policy* 7(1). Wiley Online Library: 25–36.
- 3 Connelly CE, Fieseler C, Černe M, et al. (2021) Working in the digitized economy:  
4 HRM theory & practice. *Human Resource Management Review* 31(1): 100762.  
5 DOI: 10.1016/j.hrmr.2020.100762.
- 6  
7
- 8 Cowell F (2011) *Measuring Inequality*. 3rd ed. Oxford University Press.
- 9
- 10 Dannefer D (1987) Aging as intracohort differentiation: Accentuation, the Matthew  
11 effect, and the life course. In: *Sociological forum*, 1987, pp. 211–236. Springer.
- 12
- 13 Davis G and Kanopka K (2020) Complexity Theory: Amazon’s Mechanical Turk is a  
14 disaster for crowdworkers. Available at:  
15 [https://stanforddaily.com/2020/06/21/complexity-theory-amazons-mechanical-](https://stanforddaily.com/2020/06/21/complexity-theory-amazons-mechanical-turk-is-a-disaster-for-crowdworkers/)  
16 [turk-is-a-disaster-for-crowdworkers/](https://stanforddaily.com/2020/06/21/complexity-theory-amazons-mechanical-turk-is-a-disaster-for-crowdworkers/) (accessed 3 October 2022).
- 17  
18
- 19 D’Cruz P and Noronha E (2016) Positives outweighing negatives: the experiences of  
20 Indian crowdsourced workers. *Work Organisation, Labour & Globalisation*  
21 10(1). Pluto Journals: 44–63. DOI: 10.13169/workorgalaboglob.10.1.0044.
- 22  
23
- 24 Deng X, Joshi KD and Galliers RD (2016) The duality of empowerment and  
25 marginalization in microtask crowdsourcing: Giving voice to the less powerful  
26 through value sensitive design. *Mis Quarterly* 40(2). Society for Information  
27 Management and The Management Information Systems ...: 279–302.
- 28  
29
- 30 Dennis SA, Goodson BM and Pearson C (2019) *Online Worker Fraud and Evolving*  
31 *Threats to the Integrity of MTurk Data: A Discussion of Virtual Private*  
32 *Servers and the Limitations of IP-Based Screening Procedures*. ID 3233954,  
33 SSRN Scholarly Paper, 14 March. Rochester, NY: Social Science Research  
34 Network. DOI: 10.2139/ssrn.3233954.
- 35  
36
- 37 DonovanM (2018) *Turkmaster (Mturk)*. Available at:  
38 <https://greasyfork.org/scripts/4771-turkmaster-mturk> (accessed 9 March 2021).
- 39  
40
- 41 Dror G, Koren Y, Maarek Y, et al. (2011) I want to answer; who has a question?  
42 Yahoo! answers recommender system. In: *Proceedings of the 17th ACM*  
43 *SIGKDD international conference on Knowledge discovery and data mining*,  
44 San Diego, California, USA, 21 August 2011, pp. 1109–1117. KDD ’11.  
45 Association for Computing Machinery. DOI: 10.1145/2020408.2020582.
- 46  
47
- 48 Engert M, Evers J, Hein A, et al. (2022) The Engagement of Complementors and the  
49 Role of Platform Boundary Resources in e-Commerce Platform Ecosystems.  
50 *Information Systems Frontiers*. DOI: 10.1007/s10796-021-10236-3.
- 51  
52
- 53 Fieseler C, Bucher E and Hoffmann CP (2019a) Unfairness by Design? The Perceived  
54 Fairness of Digital Labor on Crowdfunding Platforms. *Journal of Business*  
55 *Ethics* 156(4): 987–1005. DOI: 10.1007/s10551-017-3607-2.
- 56  
57
- 58 Fieseler C, Bucher E and Hoffmann CP (2019b) Unfairness by Design? The  
59 Perceived Fairness of Digital Labor on Crowdfunding Platforms. *Journal of*  
60 *Business Ethics* 156(4): 987–1005. DOI: 10.1007/s10551-017-3607-2.
- 61  
62  
63  
64  
65

- 1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65
- 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.

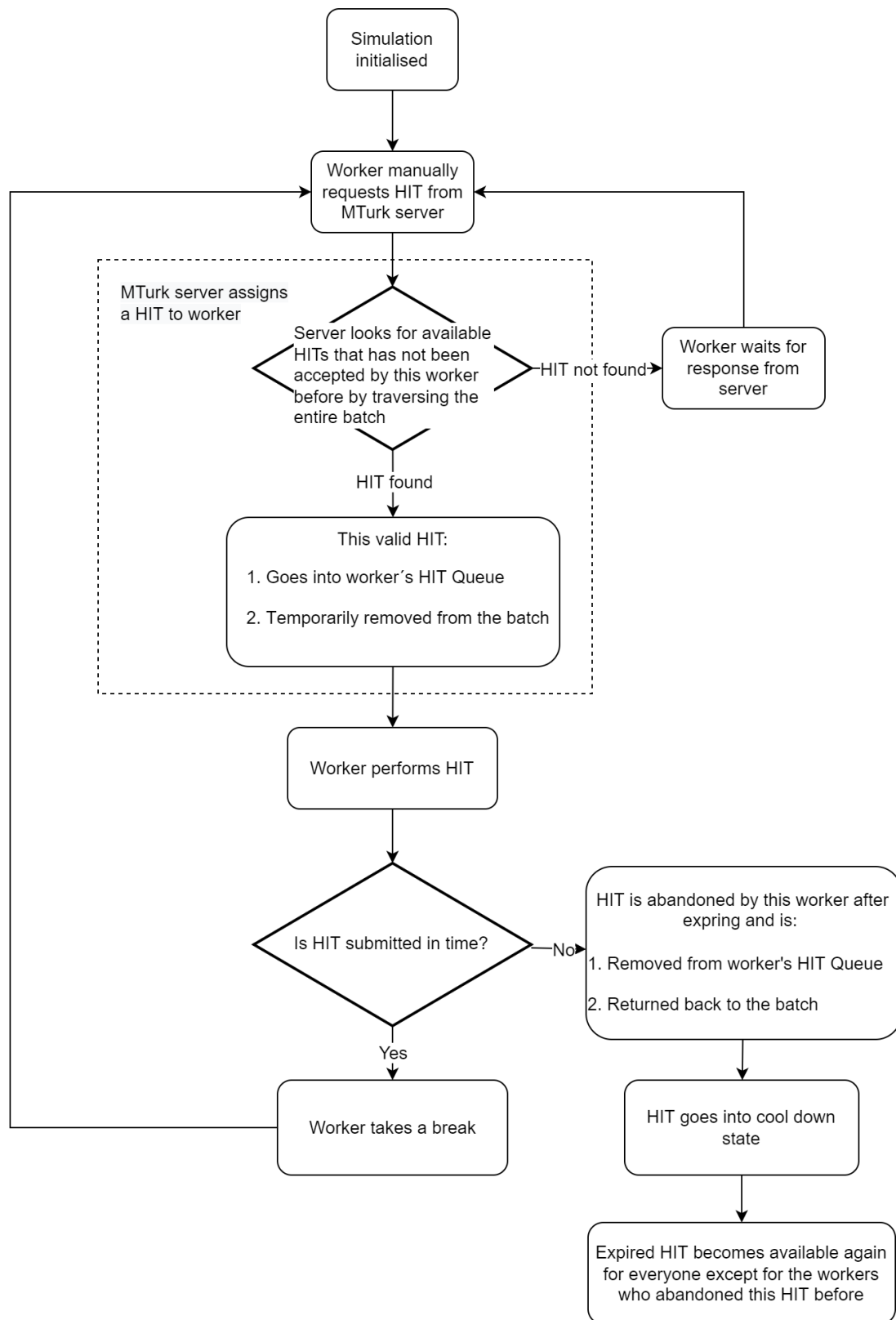
- 1 Hellman B (2021) *MTurk Suite*. JavaScript. Available at:  
2 <https://github.com/Kadauchi/mturk-suite> (accessed 21 May 2021).
- 3 Idowu A and Elbanna A (2020) Digital Platforms of Work and the Crafting of Career  
4 Path: The Crowdworkers' Perspective. *Information Systems Frontiers*. DOI:  
5 10.1007/s10796-020-10036-1.
- 6  
7  
8 Idowu A and Elbanna A (2022) Digital Platforms of Work and the Crafting of Career  
9 Path: The Crowdworkers' Perspective. *Information Systems Frontiers* 24(2):  
10 441–457. DOI: 10.1007/s10796-020-10036-1.
- 11  
12  
13 Irani LC and Silberman MS (2013) Turkopticon: interrupting worker invisibility in  
14 amazon mechanical turk. In: *Proceedings of the SIGCHI Conference on*  
15 *Human Factors in Computing Systems - CHI '13*, Paris, France, 2013, p. 611.  
16 ACM Press. DOI: 10.1145/2470654.2470742.
- 17  
18  
19 Kaplan T, Saito S, Hara K, et al. (2018) Striving to Earn More: A Survey of Work  
20 Strategies and Tool Use Among Crowd Workers.: 9.
- 21  
22 katnapped (2020) How to catch hits more quickly on hit finder? In: *r/mturk*. Reddit  
23 Post. Available at:  
24 [www.reddit.com/r/mturk/comments/g9tt14/how\\_to\\_catch\\_hits\\_more\\_quickly\\_](http://www.reddit.com/r/mturk/comments/g9tt14/how_to_catch_hits_more_quickly_on_hit_finder/)  
25 [on\\_hit\\_finder/](http://www.reddit.com/r/mturk/comments/g9tt14/how_to_catch_hits_more_quickly_on_hit_finder/) (accessed 12 March 2021).
- 26  
27  
28 Kwek A (2020) Crowdsourced Research: Vulnerability, Autonomy, and Exploitation.  
29 *Ethics & human research* 42(1). Wiley Online Library: 22–35.
- 30  
31  
32 Malik F, Heeks R, Masiero S, et al. (2021) Digital labour platforms in Pakistan:  
33 institutional voids and solidarity networks. *Information Technology & People*.  
34 Emerald Publishing Limited. DOI: 10.1108/ITP-04-2020-0218.
- 35  
36  
37 Matherly T (2019) A panel for lemons? Positivity bias, reputation systems and data  
38 quality on MTurk. *European Journal of Marketing*. Emerald Publishing  
39 Limited.
- 40  
41  
42 McInnis B, Cosley D, Nam C, et al. (2016) Taking a HIT: Designing around  
43 Rejection, Mistrust, Risk, and Workers' Experiences in Amazon Mechanical  
44 Turk. In: *Proceedings of the 2016 CHI Conference on Human Factors in*  
45 *Computing Systems - CHI '16*, Santa Clara, California, USA, 2016, pp. 2271–  
46 2282. ACM Press. DOI: 10.1145/2858036.2858539.
- 47  
48  
49 Merton RK (1968) The Matthew effect in science: The reward and communication  
50 systems of science are considered. *Science* 159(3810). American Association  
51 for the Advancement of Science: 56–63.
- 52  
53  
54 Merton RK (1988) The Matthew effect in science, II: Cumulative advantage and the  
55 symbolism of intellectual property. *isis* 79(4). Department of History and  
56 Science, University of Pennsylvania: 606–623.
- 57  
58  
59 Müller K, Vignaux T, Lünsdorf O, et al. (2021) *Simpy: Event Discrete, Process Based*  
60 *Simulation for Python*. Python. Available at: <https://simpy.readthedocs.io>  
61 (accessed 17 March 2021).
- 62  
63  
64  
65

- 1 Newman A (2019) I found work on an Amazon website. I made 97 cents an hour. *The*  
2 *New York Times*.
- 3 O’Rand AM (1996) The precious and the precocious: Understanding cumulative  
4 disadvantage and cumulative advantage over the life course. *The*  
5 *Gerontologist* 36(2). The Gerontological Society of America: 230–238.
- 6  
7  
8 Panteli N, Rapti A and Scholarios D (2020) ‘If He Just Knew Who We Were’:  
9 Microworkers’ Emerging Bonds of Attachment in a Fragmented Employment  
10 Relationship. *Work, Employment and Society* 34(3). SAGE Publications Ltd:  
11 476–494. DOI: 10.1177/0950017019897872.
- 12  
13  
14 Ramirez J (2021) *PandaCrazy Max*. Available at:  
15 <https://github.com/JohnnyRS/PandaCrazy-Max> (accessed 21 May 2021).
- 16  
17  
18 Reschke BP, Azoulay P and Stuart TE (2018) Status Spillovers: The Effect of Status-  
19 conferring Prizes on the Allocation of Attention. *Administrative Science*  
20 *Quarterly* 63(4). SAGE Publications Inc: 819–847. DOI:  
21 10.1177/0001839217731997.
- 22  
23  
24 Saito S, Chiang C-W, Savage S, et al. (2019) TurkScanner: Predicting the Hourly  
25 Wage of Microtasks. In: *The World Wide Web Conference*, San Francisco, CA,  
26 USA, 13 May 2019, pp. 3187–3193. WWW ’19. Association for Computing  
27 Machinery. DOI: 10.1145/3308558.3313716.
- 28  
29  
30 Savage S, Chiang C, Saito S, et al. (2020) Becoming the Super Turker: Increasing  
31 Wages via a Strategy from High Earning Workers.: 12.
- 32  
33  
34 Schultz R (2020) *Stax*. Available at: <https://gitlab.com/StructByLightning/stax>  
35 (accessed 27 June 2021).
- 36  
37  
38 Seneta E (2013) A tricentenary history of the law of large numbers. *Bernoulli* 19(4).  
39 Bernoulli Society for Mathematical Statistics and Probability: 1088–1121.
- 40  
41  
42 Shen W, Hu YJ and Ulmer JR (2015) Competing for Attention: An empirical study of  
43 online reviewers’ strategic behavior. *Mis Quarterly* 39(3). JSTOR: 683–696.
- 44  
45  
46 *TurkerView mTurk Forum* (2021) *TurkerView’s mTurk forum*. Available at:  
47 <https://forum.turkerview.com/> (accessed 15 August 2020).
- 48  
49  
50 Van Looy B, Ranga M, Callaert J, et al. (2004) Combining entrepreneurial and  
51 scientific performance in academia: towards a compounded and reciprocal  
52 Matthew-effect? *Research Policy* 33(3). Elsevier: 425–441.
- 53  
54  
55 Vancea DCP and Nemirschi A (2020) Challenges in Using AI in Online Commerce.  
56 (2): 5.
- 57  
58  
59 Waldkirch M, Bucher E, Schou PK, et al. (2021) Controlled by the algorithm,  
60 coached by the crowd—how HRM activities take shape on digital work  
61 platforms in the gig economy. *The International Journal of Human Resource*  
62 *Management*. Taylor & Francis: 1–36.
- 63  
64  
65

- 1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65
- 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



16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65

## DES Workflow Diagram for Script Workers

