# Impact of Task Recommendation Systems in Crowdsourcing Platforms

Kathrin Borchert, Matthias Hirth
University of Würzburg, Institute of Computer Science
Würzburg, Germany
[kathrin.borchert|matthias.hirth]@informatik.
uni-wuerzburg.de

Steffen Schnitzer, Christoph Rensing
University of Darmstadt, Multimedia Communications Lab
Darmstadt, Germany
[steffen.schnitzer|christoph.rensing]@kom.tu-darmstadt.
de

## ABSTRACT

Commercial crowdsourcing platforms accumulate hundreds of thousand of tasks with a wide range of different rewards, durations, and skill requirements. This makes it difficult for workers to find tasks that match their preferences and their skill set. As a consequence, recommendation systems for matching tasks and workers gain more and more importance. In this work we have a look on how these recommendation systems may influence different fairness aspects for workers like the success rate and the earnings. To draw generalizable conclusions, we use a simple simulation model that allows us to consider different types of crowdsourcing platforms, workers, and tasks in the evaluation. We show that even simple recommendation systems lead to improvements for most platform users. However, our results also indicate and shall raise the awareness that a small fraction of users is also negatively affected by those systems.

## 1 INTRODUCTION

In recent years the diversity of crowdsourcing services and applications has dramatically grown. Especially commercial crowdsourcing platforms focusing on micro tasking, e.g. Amazon Mechanical Turk [1] or Microworkers [2], accumulate a huge variety of different task types. These tasks, e.g. tagging images or answering surveys, are mostly repetitive, simple and their completion requires only a short amount of time. Regardless of their simplicity, most tasks still need a certain skill set on the worker's side for a successful completion.

The large number and variety of tasks and their individual requirements calls for an automatic solution to help the workers to find suitable tasks which fit their individual interests and capabilities, e.g. by using personalized task recommendation systems. Contributing in such tasks may lead to a higher success rate of the workers and thus ultimately to a higher income. However, it is not clear if the integration of task recommendation systems in crowdsourcing platforms has solely positive effects. Recommendation systems might also lead to unfairness, as some workers might get assigned to only a small number of tasks or only low paid tasks.

This paper aims at raising the awareness for such potential negative effects of recommendation systems in crowdsourcing platforms. We use a simple simulation model that includes components and processes of a crowdsourcing platform on an abstract level for quantifying and analysing the effects of a task recommendation system. A very basic task recommendation algorithm and a random based approach as baseline are used for the task suggestions. This simple setup allows us to illustrate the benefits and potential drawbacks of recommendation systems in the context of crowdsourcing platforms from a high-level point of view.

The remainder of the paper is structured as followed. The related work in the second section provides an overview of recommendation mechanisms in the context of crowdsourcing. The simulation model is described in the third section, including the models for tasks, workers, recommendation and selection of tasks, as well as the chosen evaluation metrics. The evaluation is presented in the fourth section, where key influence parameters of the simulation model are identified, and a main effect analysis is used to deduce the settings for evaluating the impact of task recommendation for diverse platforms. Further, the evaluation section provides an analysis of the impact of mechanisms on the workers' earning on the diverse platforms. The fifth section concludes the paper with a discussion of the findings.

## 2 RELATED WORK

Crowdsourcing tasks differ significantly in their complexity and the skills required by the worker completing those tasks [5, 16]. Thus, one possibility to leverage the benefits of recommendation systems in the crowdsourcing context is using them to automatically find suitable tasks for the workers. Several approaches for such task recommendation systems have already been proposed. An overview over different task recommendation approaches and evaluation methods for crowdsourcing in several areas is given by Geiger and Schader [6]. Numerous mechanisms are based on content knowledge, e.g. characteristics of previously completed tasks such as category, reward or allocated time [8, 17]. In addition, Yuen et al. [18] consider the workers interactions, e.g. searching for tasks. Such previous behavior of the workers on the platforms is also used for collaborative filtering algorithms [1, 11]. In contrast to recommending tasks to workers, the concept developed by Difallah et al. [3] realizes a push methodology to find the best suited worker

---

[1]https://www.mturk.com/ Accessed: Jun. 2017
[2]https://www.microworkers.com/ Accessed: Jun. 2017

for a task by extracting interests and skills from an online social network. Still, the evaluation of all these recommendation approaches is limited to the accuracy of the recommendations or the improvement of the quality of the worker input by a practical research or offline experiments. The framework for optimizing task assignment in the field of knowledge intensive tasks introduced by [13] prevents an over or under utilization of the workers but the influence on the involved actors, e.g. reduced earnings, is not investigated.

There are already studies about the disparate impact of algorithms and computational unfairness in several fields [4, 12], e.g. algorithms used in online advertising systems [2]. Further, there are several approaches to overcome the disparate treatment or impact in the area of decision making algorithms [9, 19]. However, to the best of our knowledge, there is no study about the impact of task recommendation systems on the workers in crowdsourcing systems.

## 3    SIMULATION MODEL

We use a simulation model to evaluate the impact of task recommendation mechanisms in crowdsourcing platforms. In contrast to a real-world implementation in an existing commercial crowdsourcing platform, this allows us to analyse the impact of a broad range of different parameter settings. The remainder of this section gives a brief description of the model components and structure of the implementation. Furthermore, the implementation of the evaluated recommendation algorithms is described. Finally, we introduce the evaluation metrics used to quantify the impact of the task recommendation algorithm on the users.

### 3.1    Simulation Description

The simulation implements different components of a crowdsourcing platform, such as tasks of various categories, workers and their interactions. Each simulation run is divided into two parts, the initialization of the workers and tasks, and an event based simulation process modelling the interactions of the workers and the platform.

The discrete event simulation is again divided into three steps, the worker selection, the task selection and the task execution. We assume that every idle worker of the worker pool is searching for a task. Thus in the first step (1), we start with the selection of an idle worker. The selection follows a random uniform distribution. In the next step, the task selection (2), the recommendation algorithm determines an available task from the pool of tasks to recommend. In case the recommended task does not fit the worker's skills, with a certain probability the worker selects a suitable task randomly from the task pool by himself. If no such task is available, he accepts the recommendation and starts to work on the selected task. During the task execution step (3), the worker is busy and does not accept other tasks. The duration of the execution process is defined by the required completion time of the task. During this process, the result of the task is computed based on the skills of the worker in the requirements of the task. If the task is successfully completed, the task status will be changed to completed and removed from the system. Otherwise, the task will become available again. At this point one iteration of the event based simulation is completed, the simulation time is updated, and the worker returns to the idle state. The simulation is terminated after a specified time period.

### 3.2    Simulation Components

In the following we have a closer look at how tasks and workers are represented in the simulation. Models are based on typical structures and characteristics of real micro-tasking platforms, e.g., Amazon Mechanical Turk or Microworkers. Moreover, we explain the implementation of the recommendation algorithm and the used baseline. In the last part of this section we give an overview of the parameters of the simulation model used to specify the characteristics of the simulated platforms.

*3.2.1    Task and Category Model.* In our model, a *task* requires a set of worker skills to be completed correctly. The required skills are determined by the *category* of the task. Additionally, a task belongs to a *campaign* that groups identical tasks, as they would be submitted by a requester in a real-world platform. A campaign defines the payment, the time required for completion, and the number of identical tasks, as well as the creation time for all of its tasks.

All tasks for one simulation run are created during an initialization phase to optimize the runtime of the simulation. In a first step, $m$ categories are created. Thereafter, the campaigns are generated with negative exponentially distributed inter-arrival times. Negative exponentially distributed inter-arrival times are often a feasible assumption if a large number of traffic sources, or in this case employers, are present. This also allows us to reduce the total number of model parameters, as the higher moments of the arrival process are directly dependent on the mean inter-arrival time, even if other distributions might be more realistic, c.f. [15]. Each campaign is then randomly assigned to a category and the associated campaign properties are added. The last step initializes tasks and adds them to the pool. However, the campaigns and tasks are not directly available at the beginning of the discrete event simulation. During the simulation the state of the tasks is changed to active at the arrival time of the associated campaign.

*3.2.2    Worker Model.* In our model we assume that there are two basic *worker types*: (1) The specialized worker ($sw$), who prefers tasks of only one category and (2) the average worker ($aw$), who favors multiple categories. The amount of favored categories of the average worker varies between two up to $m$ categories.

Beside the amount of favored categories the worker types differ concerning their skills. The skills are defined by the success probability in each category. The specialized workers $sw$ are high skilled in their preferred category. Thus, in their favored categories the success probability $p_{sw}$ is very high. The success probability $p_{aw}$ of the favored categories of average workers is medium, since they do not exclusively focus on one type of tasks but have certain knowledge in a broader spectrum of different task types. Both worker types have a low success probability for less preferred categories in common. In addition to the skill set, the worker model stores the measured success rate per category and additional statistics, e.g. the total amount of completed tasks.

By using this model, the worker pool is initialized iteratively. In the first step, a newly created worker is assigned to one of two worker types. The worker type is chosen in respect to the specified share $f_{sw}$ of specialized workers. Accordingly, the amount of $aw$ is $1 - f_{sw}$. Based on the type, the preferred categories are selected out of the pool of $m$ categories. The selection follows a random uniform

distribution. In the last step, the success probability for each category is added depending on the favored worker's categories. The iteration of the creation process is completed by adding the worker to the worker pool. These steps are repeated until the predefined number of workers $w$ is reached.

## 3.3 Recommendation System

In this work we focuses solely on illustrating the potential impact of recommendation mechanisms. Thus we decided not to compare current state of the art algorithms but only use a simple content based recommendation algorithm, which recommends tasks based on characteristics of previously completed tasks. The algorithm includes an initialization phase to learn favored task categories of new users. Additionally, we implement a random based task selection as baseline for the evaluation of the recommendation mechanism. The detailed process of each approach is described in the following.

*3.3.1 Random selection.* The random selection does not consider the qualification of the workers. This means the success rate of each category is not used to determine the workers' best category. The mechanism chooses a task randomly among the available tasks.

*3.3.2 Content based selection.* The content based algorithm recommends the worker a task of the category in which his success rate ($s_c$) is greater than a threshold of 50%. We define the threshold at this level, because it is improbable that the worker receives $s_c$ greater than 50% in an unskilled category. In the case that $s_c$ is less than the threshold in all categories, the algorithm computes the category with the highest value of $s_c$. If there is more than one category with a success rate of the maximal $s_c$ or their value of $s_c$ is greater than 50%, one of them is selected by a random uniform distribution. While choosing tasks, the mechanism considers only category types of which the system contains open tasks. If there are more than one task of the selected category available the algorithm determine the earnings per minute for each campaign and then recommends the best paid task to the worker. We include this aspect, as Schnitzer et al. [14] show that workers are focused on time and money criteria while selecting tasks.

As the algorithm requires a working history, we integrate a training phase for new workers. During this phase the workers have to finish a certain amount of training tasks and their success rate is included in the computation of $s_c$. Thus, the event based simulation process is extended by an additional step, the training phase. The phase is initiated before starting the worker selection. Here, every worker has to complete the specified amount of training tasks per category. These tasks are not part of the task pool and they only differ concerning the associated category.

## 3.4 Parameter Settings

As mentioned in the description of the simulation process and its models there are several parameters which can be specified in each simulation run. These parameters are separated into two sets summarized in Table 1. The parameters of the first set define the characteristic of the simulated platform. The amount of categories $m$ describes the diversity of the task types. The share of specialized workers $f_{sw}$, their success probability $p_{sw}$ and the success probability of the average workers $p_{aw}$ characterize the workers.

| Parameter | Role | Description |
|-----------|------|-------------|
| $m$ | specification | The number of categories in the category pool of a simulation run |
| $f_{sw}$ | specification | Share of specialized workers |
| $p_{sw}$ | specification | Success probability of specialized workers in their preferred category |
| $p_{aw}$ | specification | Success probability of average workers in their favored categories |
| $w$ | workload | Total amount of workers in the simulation run |
| $t$ | workload | Mean campaign inter-arrival time in minutes. |

**Table 1: Functionality of the parameters of the simulation.**

The second set of parameters, the total amount of workers $w$ and the mean campaign inter-arrival time specify the workload of the simulated platform.

For our following evaluation we choose the parameters based on the work by Hirth et al. [7]. We use a maximum of 20 categories and realize the varying popularity by adding a higher occurrence to some of these categories. Each category is associated with three campaign types which differ concerning the payment, required time and number of tasks. We choose the payment in a range between $0.1 and $1.5 and the required completion time varies from a few minutes up to an hour for an amount of tasks from 30 to 500 per campaign. We use a rate of 0.5 for rejecting unsuitable recommendations by the workers.

## 3.5 Evaluation Metrics

Since the integration of a task recommendation mechanism may influence the dynamic of the platform, the aim of our analysis is to quantify these influences. Therefore, we define different metrics that consider the viewpoint of the workers. From a worker's perspective his success rate and the earnings are important. To evaluate the influence of the recommendation algorithms on the success rate and the earnings of the workers, we compute the average success rate per hour $s$ of each worker, as well as their average hourly earnings $e$.

In the following $h$ defines the total simulation time in hours and $sn_i$ is the amount of successfully completed tasks within hour $i$. Equation 1 describes the computation of $s$, where $n_i$ represents the number of total completed tasks within hour $i$. We only consider hours in which the worker completed at least one task.

$$s = \frac{1}{h} \sum_{i=1}^{h} \frac{sn_i}{n_i} \qquad (1)$$

We determine the average earnings per hour $e$ by Equation 2. The payment of task $j$ contained in $sn$ is represented by $e_j$.

$$e = \frac{1}{h} \sum_{i=1}^{h} \sum_{j=1}^{sn_i} e_j \qquad (2)$$

## 4 EVALUATION

In this section we evaluate the impact of the task recommendation algorithm in platforms with different characteristics. To identify simulation settings representative for a large number of real-world crowdsourcing platforms, we first analyse the effects of the platform parameters on the workers' success rate and income. Furthermore, we compare the average success rate and the average earnings per

hour of the workers achieved in platforms integrating the recommendation mechanism and the baseline.

## 4.1 Identification of Key Influence Factors

To evaluate the influences of platform characteristics on the results of the task recommendation mechanisms, we investigate which simulation parameters are the key influences factors. As mentioned earlier, there are two sets of parameters. The first set specifies the platform characteristics, i.e. the amount of categories $m$, the share of specialized workers $f_{sw}$, and the success probability of specialized workers $p_{sw}$ and average workers $p_{aw}$. The second parameter set, describes the workload of the platform. These parameters are the total amount of workers $w$ and the mean inter-arrival time $t$.

To assess the impact of the different parameters on the success rate $s$ and the earnings $e$, we run a factor analysis. We define two levels of each simulation parameter and use a $2^k$ factorial design [10]. This approach requires only a small number of simulation runs to receive results for all setting combinations. For each setting we run 1000 simulations each with a duration of six hours. The transient phase of the simulation is not excluded from the evaluation as it describes the case of new users registering in the system.

Figure 1 shows the influence of the factors on $s$ by using the recommendation approach. Each x-axis of the figure depicts the two levels of the parameter. The y-axis shows the values of $s$. The results for random based task selection are similar and therefore not shown.

The first graph displays the effect caused by the number $m$ of different task categories. The low level depicts $m = 4$ categories. We choose this value due to the average workers' characteristic of preferring at least two categories. Thus, by using $m = 4$ there are still differences between the average workers concerning the amount of favored categories. The high level $m = 20$ is equal to the maximal amount of defined categories of our simulation model. The value of $s$ observed for $m = 20$ is lower than for $m = 4$. This is due to the availability of tasks in the skilled categories of a worker. The lower the amount of categories the higher the probability that a suitable task is available. In case of four categories the probability of availability of a preferred task of a specialized worker is 25%. The probability in case of an average worker is 50% or more, because he favors between two to four categories.

The second diagram shows the influence of the share of specialized workers $f_{sw}$. The share of average workers is $1 - f_{sw}$. Thus, the low level of $f_{sw}$ describes a share of 10% of specialized workers and 90% of average workers initialized in the platform. By increasing $f_{sw}$, a lower success rate $s$ is seen. The difference between the values for the two levels is explained by the main characteristic of specialized workers. They are only skilled in one category. If there is no task available of their preferred category the probability of successfully completing a task in one of the other unskilled categories is very low. Thus, the higher the normalized amount of specialized workers is the lower is the average success rate.

The influence of the success probability of specialized workers $p_{sw}$ is visualized in the third graph. The probability to complete a task successfully is 75% at the lower level of $p_{sw}$. The upper level specifies a success probability of 90%. As expected there is a higher success rate measured by using the upper level. Here, the specialized worker completes more tasks successfully.
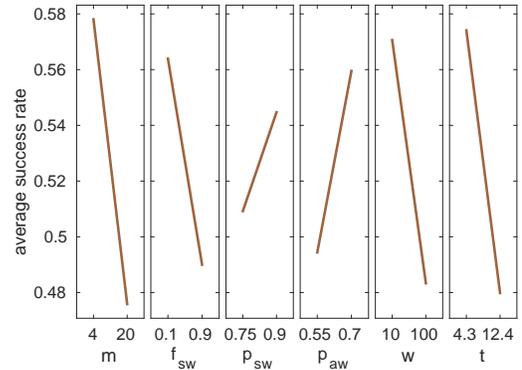


**Figure 1: Success rate per hour.**

The fourth graph shows the values of $s$ for the two levels of the success probability $p_{aw}$ of average workers. The levels are 55% and 70%. We specify these values to receive a natural order concerning $p_{sw}$. The upper level affects higher values of $s$. The reason for this effect is the same as explained in the description of graph three. The higher $p_{aw}$ the more tasks will be completed successfully.

Furthermore, the analysis shows that the total amount of workers $w$ in the platform also affects the average success rate. The effect is shown in graph five. The low level is defined by ten workers and the upper level is represented by hundred workers. These values describe the amount of employees of a small- and mid-sized business. There is a greater value of $s$ observed for the lower amount of workers. This is caused by the workload of the workers. A greater amount of workers decreases the probability that a suitable task is suggested to the requesting worker.

A similar effect is seen for the different levels of the mean inter-arrival time of campaigns $t$, which is displayed in graph six. As mentioned the inter-arrival time is described by a negative exponential function. Thus, the factor levels vary regarding the mean $t$ of this function. The upper level of $t$ is about 12.4 minutes. It is based on the results of the analysis of the campaign inter-arrival time of Microworkers. The lower level describes an average inter-arrival time of 4.3 minutes which is approximately one third of the upper level. The value of the average success rate is greater for the shorter inter-arrival time than for the upper factor level. This is due to the amount of open tasks in the platform. The lower the inter-arrival time the more campaigns will be created and the more tasks will be available in the platform. Thus, the probability of selecting tasks which fit the skills of the requesting worker is very high.

Concluding the average success rate is influenced positively by a small amount of categories, a small share of specialized workers, a high success probability of specialized and average workers, and a small total amount of workers, as well as a short campaign inter-arrival time.

The results of the factor analysis concerning the average hourly earnings of the workers are similar to the influences as described for the average success rate per hour. The similarity is caused by the dependency between the successful completion of tasks and getting paid. This means by completing more tasks successfully the

| Platform Type | $m$ | $f_{sw}$ | $p_{sw}$ | $p_{aw}$ | $w$ | $t$ |
|---|---|---|---|---|---|---|
| Specialized platform | 4 | 0.1 | 90% | 70% | 10 | 4.3 |
| Unspecialized platform | 20 | 0.9 | 75% | 55% | 100 | 12.4 |

**Table 2: Settings of a specialized and an unspecialized platform, defined by the amount of categories $m$, the share of specialized workers $f_{sw}$, the success probability of specialized workers $p_{sw}$ and average workers $p_{aw}$, the amount of workers $w$, and the mean inter-arrival time $t$.**

earnings increase. Thus, each factor which influences the success rate positively will also affect the earnings in a positive way.

## 4.2 Deductive Key Scenarios

To evaluate the influence of recommendation algorithms in platforms with different characteristics and different workload we combine the levels of the parameters which affect the success rate and earnings positively and the levels which influences are negative. The resulting simulation settings are shown in Table 2. Having a closer look at the resulting platform characteristics, we can identify two platform types.

The first platform type is specialized on a small amount of different categories and the amount of registered workers is low. Due to the small amount of categories they are not specialized on one category. This means the share of average workers is great. In addition, they are very high skilled in their preferred categories. Consequently, the probability of completing tasks of favored categories successfully is very high. The inter-arrival time of campaigns is low. The small amount of workers and the large amount of campaigns defined by the short inter-arrival time describes a high workload of the platform. This workload influences the success rate and the earnings positively.

The other platform type described by the second setting combination shown in Table 2, represents a non-specialized crowdsourcing platform. The platform offers a great amount of various task categories, which results in a lower success rate and hourly earnings. This results in a specialization of a great part of workers specified by $f_{sw} = 0.9$. Overall the success probability of all workers is lower than in the other platform type. However, there are more workers registered in the platform. Due to the longer inter-arrival time, there are less campaigns created in this platform type and thus, the workload is low.

The workload of both platform types can be varied by changing the ratio of workers and created campaigns. This means, by the reduction of registered workers and the decrease of the mean inter-arrival time, the workload increases.

In the next subsection we investigate the impact of the task recommendation algorithm and the baseline on the average success rate per hour and the hourly earnings per worker by setting up the simulation model with the parameters of the two platform types.

## 4.3 Influence on Success Rate and Earnings

To evaluate the impact of the recommendation system on the average earnings $e$ of each worker in combination with the received average success rates $s$, we normalize the hourly earnings by the highest seen income per simulation run. The maximal amount of



**(a) Specialized platform**
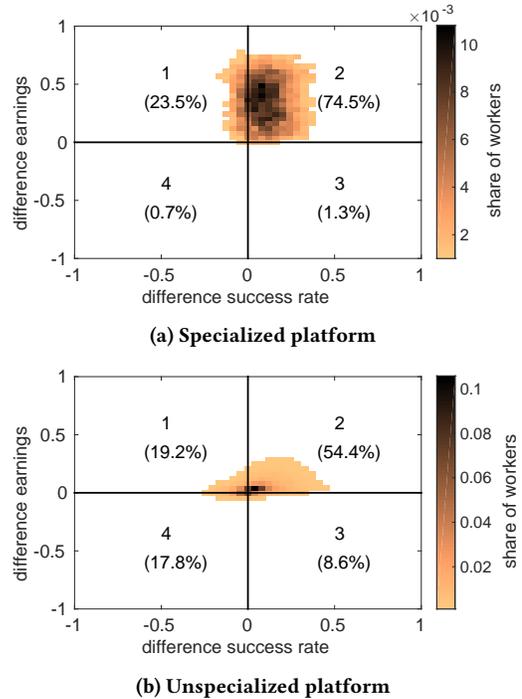


**(b) Unspecialized platform**

**Figure 2: Differences between random and content based recommendation concerning the success rate and the earnings**

money to earn per run depends on the task categories which are contained in this model. Based on these values we compute the differences of $e$ and $s$ gathered while using the content based system and the random based approach. The differences quantify the improvement when using the content based mechanism. Larger values of these differences imply a greater enhancement. This means the hourly wages and the success rate is higher.

To obtain comparable results we run both task selection mechanisms on the same generated models. This includes all model configurations which means tasks and workers.

The improvement per worker measured in a specialized platform is visualized in a 3D-histogram in Figure 2a. The colored areas describe the amount of workers with a specific difference of $s$ and $e$ normalized by the total amount of workers. The darker the color of an area, the greater is the share of workers. We omit outliers which are represented by areas containing a share of workers less than 1‰. By separating the figure in four sections, we group the workers based on their difference values. Thus, we can analyse the amount of workers with an increase of $s$ and $e$, an increase of only one of these values or those who are earned less in combination with a lower success rate.

We observe a small negative average difference of the success rate of workers of section 1 in the upper left. Thus, the earnings are only increased. The workers of the second section which means the upper right quadrant, benefit concerning their success rate and their earnings when using the recommendation system. Here, the share of workers is 74.47%. The average success rate of the workers in the lower right section (3) is increased whereas their earnings are

not significantly decreased. There is no improvement for workers residing in section 4 in the lower left. The share of workers of section 3 and 4 is negligible small.

Concluding, the usage of the content based system increases the average earnings and the average success rate of 74.47% of the workers during a simulation time period of 6 hours in a specialized platform. For 23.55% of the workers only the earnings are increased while their success rate is not significant decreased.

Figure 2b shows a 3D-histogram of the workers registered in an unspecialized platform. In this case the upper left quadrant (1) contains 19.19% of the workers. The second section in the upper right which describes the case that $s$ and $e$ are increased contains 54.4% of the workers. 8.58% of the workers are grouped in section 3. The worst case is shown in the lower left section (4). Here, the earnings and the success rate are slightly decreased for 17.83% of the workers. In conclusion the content based system achieves an increase of the earnings and the success rate for 54.4% of the workers. The increase of the success rate is higher than for the earnings, due to the amount of available tasks in the platform specified by the workload. The probability that tasks of different campaigns of favored categories are available, is very low. Thus, the recommendation mechanism suggests the tasks without considering their payment.

Concluding, we observe that $e$ and $s$ are affected by integrating different task recommendation algorithms in both platform types. The content based technique results in a higher success rate and income for more workers than the baseline. The analysis of the influence on the hourly earnings $e$ shows also an increase of the earnings for the content based system by comparing the values to the random approach.

## 5 CONCLUSION

Recommendation systems are nowadays integrated in many services and applications to help coping with the tremendous amount of data and items available. This makes them also likely to be valuable tool in commercial crowdsourcing platforms, to help mapping tasks to workers who have the skills to complete them successfully. Even if there already exist several work in this direction, no systematic evaluation was available on how those systems affect workers on the platform.

To tackle this question, we built a simulation model of a crowdsourcing platform including recommendation mechanisms. Based on the analysis of influences of the simulation parameters, we identified key scenarios which describe two different platform types. We investigated the impact of a content based recommendation algorithm concerning the workers' success rate and the earnings.

The analysis of the results shows that hourly earnings and success rates are impacted by recommendation in both scenarios. For the non specialized platform scenario, the success rates and earnings are positively affected for a significant amount of workers, while a small share of workers (17.83%) is negatively affected.

There are still several quality criteria and aspects which could be investigated by using the simulation model. One aspect is the fairness of the task distribution between the workers. Furthermore, the variety of recommended tasks to workers who are skilled in more than one category could be evaluated.

## REFERENCES

[1] Vamshi Ambati, Stephan Vogel, and Jaime G. Carbonell. 2011. Towards Task Recommendation in Micro-Task Markets.. In *Workshop on Human Computation*.
[2] Amit Datta, Michael Carl Tschantz, and Anupam Datta. 2015. Automated experiments on Ad privacy settings. *Privacy Enhancing Technologies* (2015).
[3] Djellel Eddine Difallah, Gianluca Demartini, and Philippe Cudre-Mauroux. 2013. Pick-a-crowd: tell me what you like, and i'll tell you what to do.. In *International World Wide Web Conferences*.
[4] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. In *International Conference on Knowledge Discovery and Data Mining*.
[5] Ujwal Gadiraju, Ricardo Kawase, and Stefan Dietze. 2014. A taxonomy of microtasks on the web. In *Conference on Hypertext and social media*.
[6] David Geiger and Martin Schader. 2014. Personalized task recommendation in crowdsourcing information systems - Current state of the art. *Decision Support Systems* 65 (2014).
[7] Matthias Hirth, Tobias Hoßfeld, and Phuoc Tran-Gia. 2011. Anatomy of a crowdsourcing platform-using the example of microworkers.com. In *International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*.
[8] Chien-Ju Ho and Jennifer Wortman Vaughan. 2012. Online Task Assignment in Crowdsourcing Markets.. In *International Conference on Artificial Intelligence*.
[9] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. 2012. Fairness-aware classifier with prejudice remover regularizer. *Machine Learning and Knowledge Discovery in Databases* (2012).
[10] Averill M. Law and David M. Kelton. 1999. *Simulation Modeling and Analysis* (3rd ed.). McGraw-Hill Higher Education.
[11] Habibur Rahman, Lucas Joppa, and Senjuti Basu Roy. 2016. Feature Based Task Recommendation in Crowdsourcing with Implicit Observations. *arXiv preprint arXiv:1602.03291* (2016).
[12] Andrea Romei and Salvatore Ruggieri. 2014. A multidisciplinary survey on discrimination analysis. *The Knowledge Engineering Review* 29, 05 (2014).
[13] Senjuti Basu Roy, Ioanna Lykourentzou, Saravanan Thirumuruganathan, Sihem Amer-Yahia, and Gautam Das. 2015. Task assignment optimization in knowledge-intensive crowdsourcing. *The VLDB Journal* 24, 4 (2015).
[14] Steffen Schnitzer, Christoph Rensing, Sebastian Schmidt, Kathrin Borchert, Matthias Hirth, and Phuoc Tran-Gia. 2015. Demands on task recommendation in crowdsourcing platforms-the worker's perspective. In *CrowdRec Workshop*.
[15] Christian Schwartz, Kathrin Borchert, Matthias Hirth, and Phuoc Tran-Gia. 2015. Modeling crowdsourcing platforms to enable workforce dimensioning. In *International Conference on Telecommunication Networks and Applications Conference*.
[16] Jie Yang, Judith Redi, Gianluca DeMartini, and Alessandro Bozzon. 2016. Modeling Task Complexity in Crowdsourcing. In *Conference on Human Computation and Crowdsourcing*.
[17] Man-Ching Yuen, Irwin King, and Kwong-Sak Leung. 2011. Task matching in crowdsourcing. In *International Conference on Cyber, Physical and Social Computing*.
[18] Man-Ching Yuen, Irwin King, and Kwong-Sak Leung. 2015. Taskrec: A task recommendation framework in crowdsourcing systems. *Neural Processing Letters* 41, 2 (2015).
[19] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. 2016. Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment. *arXiv preprint arXiv:1610.08452* (2016).