# Enhancing Crowdworkers' Vigilance

**Avshalom Elmalech**[1], **David Sarne**[2], **Esther David** [3], **Chen Hajaj**[4]
[1] Harvard University, USA
[2] Bar-Ilan University, Israel
[3] Ashkelon Academic College, Israel
[4] Vanderbilt University, USA
elmalech@seas.harvard.edu, sarned@cs.biu.ac.il,
astrdod@acad.ash-college.ac.il, chen.hajaj@vanderbilt.edu

## Abstract

This paper presents methods for improving the attention span of workers in tasks that heavily rely on their attention to the occurrence of rare events. The underlying idea in our approach is to dynamically augment the task with some dummy (artificial) events at different times throughout the task, rewarding the worker upon identifying and reporting them. The proposed approach is an alternative to the traditional approach of exclusively relying on rewarding the worker for successfully identifying the event of interest itself. We propose three methods for timing the dummy events throughout the task. Two of these methods are static and determine the timing of the dummy events at random or uniformly throughout the task. The third method is dynamic and uses the identification (or misidentification) of dummy events as a signal for the worker's attention to the task, adjusting the rate of dummy events generation accordingly.

## 1 Introduction

The past decade has seen significant growth in crowdsourcing applications. The crowdsourcing model encapsulates a key question that has captured the attention of researchers for a long time– *How to motivate workers so they perform higher quality work*. This question is of great importance especially when there is evidence for a group of workers who are primarily interested in producing quick labor rather than quality [Laws *et al.*, 2011; Akkaya *et al.*, 2010]. Various methods have been proposed to address this problem. Some works examine the use of different financial compensation mechanisms which are based on the workers' performance [Yin and Chen, 2015; Mason and Watts, 2010; Gao *et al.*, 2012; Feng *et al.*, 2014]. Another line of work [Law *et al.*, 2016; Kaufmann *et al.*, 2011] suggests the use of intrinsic behavioral factors to motivate workers. The mechanisms described above were found to be successful in scenarios where the lack of motivation is due to low engagement level of some workers. In this paper we focus on domains where the lack of motivation is due to the task structure and requirements. More specific, we focus on simple tasks that require moderate, yet continuous, attention on the workers side, with a very low cognitive load. Examples for such tasks include watching suitcases passing through an X-ray machine (e.g., at airports) with the aim of detecting sharp objects or explosives, and watching streams arriving from Closed Circuit Televisions (CCTVs) with the aim of identifying crime. The performance of a worker in such tasks is critically correlated with the extent she is tuned to the continuous sequence of events, as even with the slightest loss of attention the event of interest may be missed. Furthermore, common to all the above examples, that the work is highly monotonous and normally workers' attention degrade with time [Rahman, 2012].

In this paper we present a method for overcoming the degradation in workers' attention span over time in monitoring tasks. While other solutions consider fully rational players[Rahman, 2012], our approach considers human workers that are known to act irrationally [Hajaj *et al.*, 2015; 2016; Elmalech *et al.*, 2016a; 2015a; 2015b]. We propose a mechanism that is based both of financial incentives and intrinsic behavioral factors to motivate workers. Our approach is based on intelligent intersperse of artificial ("dummy") events, rewarding the worker upon successfully identifying them. While the underlying idea itself is quite simple, the challenging aspect of the proposed method is the determination of when to introduce dummy events. We propose and provide a thorough evaluation of three methods for generating dummy events. The first two suggest a simple scheme of introducing a pre-specified number of dummy events either at random times or uniformly (i.e., at fixed intervals) throughout the task. The advantage of these schemes is mainly in the bound they put on the payment to the worker, as the number of dummy events generated is fixed and pre-determined. The third method we propose is inherently dynamic. It uses the dummy events (and their identification and misidentification) for modeling the worker's attentional state and makes decisions concerning the introduction of additional dummy events based on this measure, on the fly. This way, dummy events are introduced only when necessary, resulting in lower expected expense overall. This is an abridged report, for detailed version see [Elmalech *et al.*, 2016b].

## 2 Generating Dummy Events

Our proposed method for improving workers' performance in tasks that require the worker's attention relies on artificially embedding dummy events throughout the task, in an intelligent controlled manner. A dummy event is an event of interest for which the worker is compensated if identified on time, despite the fact that its identification is useless for the employer. The introduction of dummy events throughout a task is challenging in the sense that there are several parameters affecting its effectiveness. An intelligent design of a dummy-events based mechanism should properly determine the number of dummy events to introduce to workers, the timing of the insertion of the dummy events along the task, the payment for each successful identification of a dummy event and the payment for the primary event of interest. In this paper, we therefore focus on the mechanisms for timing the insertion of the dummy events. Two intuitive methods for generating dummy events along the task are to spread the events evenly (uniformly) along the task and to randomly draw the timings when such events should appear. Both methods guarantee that the dummy events are spread along the task to avoid long periods of time with no dummy events (that can potentially bore the worker and push her to abandon the task or temporarily focus in something else). The uniform spread guarantees a more steady flow of dummy events, hence convincing the worker that it is beneficial to keep focused in the task. It does, however, have a drawback in the sense that the worker may quickly learn the dummy-event generation pattern and consequently switch to other tasks in between, as she knows when the next dummy event will appear. Furthermore, once recognizing the dummy-event generation pattern the worker will be able to assess the expected payment resulting from detecting the dummy events. Since our goal at the end of the day is to increase the attention span with a lower effective payment overall, the worker, realizing it is not rewarding enough, is likely to become disappointed and abandon the task. Using random timings (the second method) resolves this latter problem, however can lead to relatively long periods of time where no dummy event is presented to the worker. In addition to the above methods, we propose a third mechanism that does not determine the exact times for introducing dummy events a priori. Instead it dynamically allocates dummy events in a way that motivates workers to stay tuned to the task (hence we refer to it onwards as DDEA - Dynamic Dummy-Event Allocation). The decision to introduce a dummy event at any given time is probabilistic, where the probability of such event depends on the time elapsed since the last introduction of a dummy event and to some extent also on the weighted aggregated prior behaviors exhibited by the worker, as captured by the results (success or failure in identification) of previous dummy events. These two factors influence the value of an attentiveness measure that the mechanism maintains, denoted $F$, aiming to capture the worker's attentional state at each time. The value of $F$ ranges between $0-1$, where 0 represents no attention to the task and 1 represents full attention. Failing to identify a dummy event will result in a relatively sharp decrease in the value of $F$, whereas a correct identification will result in a sharp increase. Both the increase and the decrease in the value of $F$ use exponential smoothing techniques such that the new value depends on all prior values of the measure, with an exponentially decreasing weight to each prior value according to the time elapsed since it was set. Additional adaptations to the value of $F$ occur based on the time elapsed since the last evidence of the worker's attentional state was received, i.e., based on the time that elapsed since the last dummy event was introduced, in a way that exhausts the value of $F$ over time. The decrease of the value of $F$ over time reflects (to some extent) typical people's attention span model (e.g., Figure 6.4, page 60, in [Aarabi, 2007]). Naturally, we expect the third method to outperform the first two, as unlike them it correlates the choice of introducing a dummy event with some prediction of the worker's current attentional state. Still, we believe it is important to study the first two methods due to their simplicity, intuitiveness and the fact they put a bound on the total expense.

The adaptation process is compactly captured in Algorithm1. The value of $F$ is first initialized to 1, as it is most likely that at the beginning of the task the worker is fully tuned to it. The decision concerning the introduction of a dummy event takes place every few seconds (modeled using the parameter $DecisionPointsInterval$). Once a new decision point is reached (Step 2), the mechanism reduces $F$ by a factor of $\delta$. The $F$ value corresponds to the probability that the worker is not focusing (i.e., doing something else), which decreases as time goes by and no other indication of her attentiveness was received. The choice of the proper $\delta$ value depends on the variable $DecisionPointsInterval$—the greater the value of $DecisionPointsInterval$, i.e., the greater the time elapsed since the last time the value of $F$ was reduced, the greater the reduction in $F$ should be. For example, the discounting of $F$ from 1 to, say, third, whenever discounting every 3 seconds ($DecisionPointsInterval = 3sec$) requires $\delta = 0.99$, if discounting over 5 minutes and $\delta = 0.997$ if discounting over 20 minutes. These of course refer to the case where the value is continuously discounted, without receiving any new information from the introduction of a dummy event.

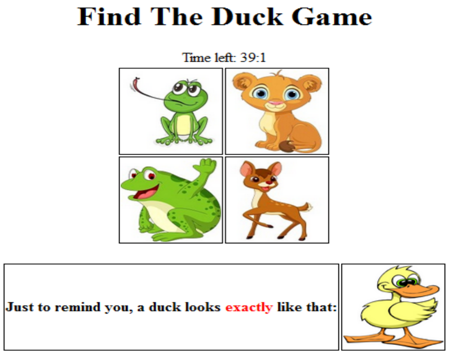
The $F$ value is then used for deciding whether or not to introduce a dummy event (Step 4). This is achieved by comparing $F$ to a random number drawn from a uniform probability distribution function in the range $0-1$. In case a dummy event is introduced and identified by the worker (Step 7), the value of $F$ is increased. The increase has a fixed component, represented by $\alpha$ and the remaining increase is positively correlated with the current value of $F$. Suggested values for $\alpha$ are thus within the range of $0.80 - 0.95$, representing a relatively high confidence in having the worker's attention fully focused in the task based on a successful identification of a dummy event. In case the dummy event was not identified upon its introduction to the worker, the value of $F$ decreases by a factor of $\beta$ (Step 10). The value of $\beta$ should be substantially smaller than $\delta$ (in at least one order of magnitude), as the events are very different—while $\beta$ corresponds to the event of a dummy event introduced to the worker and not properly identified, $\delta$ corresponds simply to the increased chance of losing focus as time goes by. Still, the idea is that the values for $\beta$ will not be too small (e.g., such that $F$ will become too close to zero), because it is possible that the worker is generally tuned to the task, but due to a temporary disturbance missed the dummy event.

**Algorithm 1:** Dynamic Dummy-Event Allocation (DDEA).

```
input : $DecisionPointsInterval$
initialization:
  $NextDecisionPoint = CurrentTime();F = 1;$
  while $TaskIsOn$ do
    if $CurrentTime \geq NextDecisionPoint$ then
      $F = F * \delta;$
      if $Random() > F$ then
        IntroduceDummyEvent();
        NextDecisionPoint+=DecisionPointsInterval;
        if worker identified DummyEvent then
          $F = \alpha + (1 - \alpha)F;$
        else
          $F = \beta F;$
        end
      end
    end
end
```

## 3 Experimental Design

For the experiments, we used an Internet game called "Find the Duck". The game's GUI is composed of four tiles visible to the worker, each with a different picture from a repository of 45 cartoon animals. Figure 1 presents a screen-shot of this game. Each $\sim$ 4 seconds the picture on one of the tiles is replaced by a different one from the repository, where both the tile that will be changed and the new picture are chosen randomly. The worker gains rewards in the game whenever clicking on a tile that has one of some pre-specified pictures appearing on it. The worker receives a graphical indication (a summary of the number of missed pictures of interest, appearing at the bottom of the screen) for every event that was not identified on time. The length of the game was set to 40 minutes.



Find The Duck Game

Time left: 39:1

Just to remind you, a duck looks exactly like that:

Figure 1: A screen-shot of the game.

Participants received an explanation about the compensation structure, which was composed of a show-up fee (fixed wage) of 5¢ and a bonus which depended on whether or not the event of interest (a duck) was identified and the number of dummy events (ducks) spotted by the participant throughout the experiment. The information regarding the events that was provided to participants specified that one duck will appear at some unknown time (equivalent to settings where we know a crime event happened throughout the task, yet the exact time is unknown) whereas the number of goats that will appear (for the treatments where the dummy-event methods were tested) remained unknown. Participants were recruited through Amazon Mechanical Turk (AMT) and were assigned to one of 19 game sessions of 4 treatments as specified in the following table:

| Treatment | Duck (primary) | Goats (dummy) |
|---|---|---|
| No dummies (8 variants) | $bonus \in \{0\text{¢}, 10\text{¢}, 20\text{¢}, 40\text{¢}, 60\text{¢}, 80\text{¢}, 100\text{¢}, 200\text{¢}\}$ | N/A |
| Random (5 variants) | $bonus = 10\text{¢}$ | $bonus = 1\text{¢}$, # of goats$\in \{10, 20, 30, 40, 50\}$ |
| Uniform (5 variants) | $bonus = 10\text{¢}$ | $bonus = 1\text{¢}$, # of goats$\in \{10, 20, 30, 40, 50\}$ |
| DDEA (1 variant) | $bonus = 10\text{¢}$ | $bonus = 1\text{¢}$, # of goats=according to DDEA algorithm |

Table 1: Summary of the different experiment conditions.

In the first treatment ("No dummies") we aimed to test the method of controlling the worker's attentional state through the reward she receives for identifying the primary event of interest. Hence no goats were used and the bonus promised exclusively depended on whether or not the duck was found. We had 8 variants of this treatment, differing in the bonus awarded for finding the duck: 0,10,20,40,60,80, 100 and 200 cents. The other three treatments aim to test the dummy-events based methods: generating dummy events in uniform intervals, at random times and using the DDEA technique. In all three the workers were promised a 10¢ bonus for spotting the duck and 1¢ for each goat (dummy event). The treatments "Random" and "Uniform" test generating dummy events at random and uniform times, respectively. Each such treatment was used with five variants, differing in the number of goats used $(10, 20, 30, 40, 50)$. The treatment DDEA used the method of dynamic dummy event augmentation according to the DDEA technique.

## 4 Results

Figure 2 depicts the required bonus and the resulting expected effective payment as a function of the detection probability one aims to achieve when not using dummy events (the "No-dummies" treatment). The first curve is based on the results of the eight "No-dummies" treatment variants. Each of its data points represents the appropriate percentage of participants who managed to spot the duck in the appropriate session (horizontal axis) for a specific value tested as the bonus for finding the duck (vertical axis). The curve was smoothed by means of the *smoothing spline* method, resulting in $f(x) = 1575 * x^3 - 1271 * x^2 + 401.6 * x - 39.27$

(with $R^2 = 0.992$). The second curve which represents the effective expense (the actual payment required for guaranteeing the detection probability of the horizontal axis) is a direct transformation of the latter curve. The transformation is done by multiplying the proposed bonus by the probability it will actually be awarded (i.e., the probability on the horizontal axis) and adding the 5¢ fixed payment for the HIT.

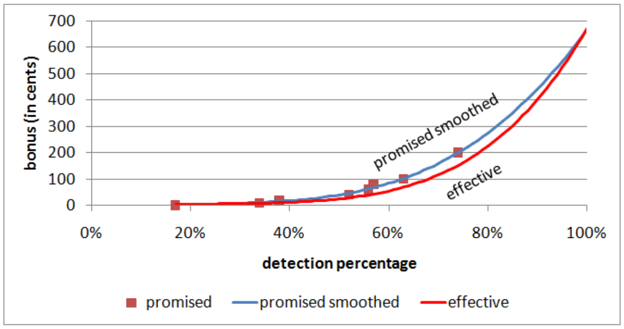


Figure 2: The correlation between detection percentage, reward promised and effective payment.

As expected, the probability that a worker will identify the primary event of interest increases as the reward promised for spotting it increases. The importance of the graph, however, is in enabling the extraction of a baseline for comparison, as the exact marginal improvement due to any additional cent promised as a bonus can only be found through experimentation and smoothing. Based on the tradeoff between the expected payment and the achieved precision encapsulated in Figure 2 one can choose a preferred working point and set the bonus accordingly.

## 4.1 Random and Uniform Dummy Events

Next, we report the results obtained with the use of dummy events when using random and uniform generation patterns. The resulted detection probability and the corresponding effective expense for each of the uniform and random dummy event generation method variants are given in Table 2. From Table 2 we observe that, as expected, the increase in the number of dummy events introduced into the task results in an increase in the detection probability, with the cost of an increase in the effective expense. Neither method (uniform or random dummy event generation) generally dominates the other and the detection probability is almost identical with the two for any number of dummy events tested. Similarly, the effective expense is similar, with the most notable difference when having 50 dummy events overall. Here, the expected expense with the random method is lower, possibly explained by people's ability to learn the pattern of generating dummy events with the uniform method. Still, one would expect a similar learning effect with the 30 and 40 treatment variants, which is not the case.

In comparison to the "no-dummies" approach, both the random and uniform dummy event generation methods suggest a substantially more competitive tradeoff—a detection probability of $67\% - 83\%$ is achieved with an effective expense of $17.6$¢ $- 49.8$¢ and $18.4$¢ $- 58.4$¢ for random and uniform, respectively, compared to a required effective expense of $94$¢ $- 269.7$¢ with "no dummies" for this interval of detection probabilities. These differences are statistically significant using t-test ($p < 0.01$).

| | random | | uniform | |
|---|---|---|---|---|
| | effective expense | detection percentage | effective expense | detection percentage |
| 10 goats | 17.6 | 67% | 18.4 | 67% |
| 20 goats | 24 | 70% | 27.3 | 70% |
| 30 goats | 39 | 80% | 37 | 81% |
| 40 goats | 49.8 | 83% | 48.3 | 81% |
| 50 goats | 52.9 | 83% | 58.4 | 83% |

Table 2: Detection percentage and effective expense as a function of number of goats.

## 4.2 Dynamic Dummy Events Generation

Finally, our DDEA-based method achieved a detection probability of $81\%$ with a corresponding effective expense of 24¢. This result is substantially better than those obtained with the first two methods—from Table 2 we observe that the effective expense required for achieving this level of precision dictates an expected expense of somewhere between $39 - 49.8$¢ with the method that generates dummy events at random times and 37¢ with the one that spreads them uniformly.

The difference in the expected expense is statistically significant using t-test (taking the $81\%$ detection probability as a baseline). Compared to the method that relies solely on rewarding the identification of the event of interest ("no-dummies"), the performance of DDEA is strikingly better—the corresponding expected expense for assuring a detection probability of $81\%$ is 239¢ (10 times more!) and the detection probability achieved in exchange for an effective expense of 24¢ is $49\%$, according to Figure 2.

## 5 Conclusions

The encouraging results reported in the former section support our hypothesis that the effective way for increasing workers' attention span in such monotonous crowdsourcing monitoring tasks is the one that uses dummy events rather than the traditional method of increasing the reward for identifying the primary event of interest. As discussed throughout the paper, the benefits of the method are threefold. First, it enables workers to accumulate rewards throughout the task, rather than waiting for a single meaningful event, hence reducing the variance in the payment received. Secondly, the task as a whole may become more interesting to the worker. Lastly, it enables indications for the worker's attentional state throughout the task.

## Acknowledgements

The research presented in this paper was partially supported by the Israel Science Foundation (grant No. 1083/13),the ISF-NSFC joint research program (grant No. 2240/15), the Ministry of Science, Technology and Space, Israel with the National Science Council (NSC) of Taiwan, and a Harvard Center for Research on Computation and Society fellowship.

## References

[Aarabi, 2007] Parham Aarabi. *The art of lecturing: a practical guide to successful university lectures and business presentations*. Cambridge University Press, 2007.

[Akkaya *et al.*, 2010] Cem Akkaya, Alexander Conrad, Janyce Wiebe, and Rada Mihalcea. Amazon mechanical turk for subjectivity word sense disambiguation. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 195–203. Association for Computational Linguistics, 2010.

[Elmalech *et al.*, 2015a] Avshalom Elmalech, David Sarne, and Barbara J Grosz. Problem restructuring for better decision making in recurring decision situations. *Autonomous Agents and Multi-Agent Systems*, 29(1):1–39, 2015.

[Elmalech *et al.*, 2015b] Avshalom Elmalech, David Sarne, Avi Rosenfeld, and Eden Shalom Erez. When suboptimal rules. In *Proc. of AAAI*, pages 1313–1319, 2015.

[Elmalech *et al.*, 2016a] Avshalom Elmalech, David Sarne, and Noa Agmon. Agent development as a strategy shaper. *Autonomous Agents and Multi-Agent Systems*, 30(3):506–525, 2016.

[Elmalech *et al.*, 2016b] Avshalom Elmalech, David Sarne, Esther David, and Chen Hajaj. Extending workers' attention span through dummy events. In *Fourth AAAI Conference on Human Computation and Crowdsourcing*, 2016.

[Feng *et al.*, 2014] Zhenni Feng, Yanmin Zhu, Qian Zhang, Lionel M Ni, and Athanasios V Vasilakos. Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing. In *Proceedings of INFOCOM*, pages 1231–1239, 2014.

[Gao *et al.*, 2012] Xi Alice Gao, Yoram Bachrach, Peter Key, and Thore Graepel. Quality expectation-variance tradeoffs in crowdsourcing contests. In *Proc. of AAAI*, 2012.

[Hajaj *et al.*, 2015] Chen Hajaj, Noam Hazon, and David Sarne. Improving comparison shopping agents' competence through selective price disclosure. *Electronic Commerce Research and Applications*, 14(6):563–581, 2015.

[Hajaj *et al.*, 2016] C. Hajaj, N. Hazon, and D. Sarne. Enhancing comparison shopping agents through ordering and gradual information disclosure. *to appear in JAAMAS*, 2016.

[Kaufmann *et al.*, 2011] Nicolas Kaufmann, Thimo Schulze, and Daniel Veit. More than fun and money. worker motivation in crowdsourcing-a study on mechanical turk. In *AMCIS*, volume 11, pages 1–11, 2011.

[Law *et al.*, 2016] Edith Law, Ming Yin, Joslin Goh, Kevin Chen, Michael A Terry, and Krzysztof Z Gajos. Curiosity killed the cat, but makes crowdwork better. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4098–4110. ACM, 2016.

[Laws *et al.*, 2011] Florian Laws, Christian Scheible, and Hinrich Schütze. Active learning with amazon mechanical turk. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1546–1556. Association for Computational Linguistics, 2011.

[Mason and Watts, 2010] Winter Mason and Duncan J Watts. Financial incentives and the performance of crowds. *ACM SigKDD Explorations Newsletter*, 11(2):100–108, 2010.

[Rahman, 2012] David Rahman. But who will monitor the monitor? *The American Economic Review*, pages 2767–2797, 2012.

[Yin and Chen, 2015] Ming Yin and Yiling Chen. Bonus or not? learn to reward in crowdsourcing. In *Proc. of IJCAI*, pages 201–207, 2015.