IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 May 2025, accepted 9 June 2025, date of publication 13 June 2025, date of current version 20 June 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3579418

APPLIED RESEARCH

Bandit Algorithms for Efficient Toxicity Detection in Competitive Online Video Games

JACOB MORRIER[1], RAFAL KOCIELNIK[2], AND R. MICHAEL ALVAREZ[1]
[1]Division of the Humanities and Social Sciences, California Institute of Technology, Pasadena, CA 91125, USA
[2]Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125, USA

Corresponding author: Jacob Morrier (jmorrier@caltech.edu)

This work was supported by a sponsored research grant from Activision®.

ABSTRACT This article considers the problem of efficient sampling for toxicity detection in competitive online video games. Video game service operators take proactive measures to detect and address undesirable behavior, seeking to focus these costly efforts where such behavior is most likely. To achieve this objective, service operators need estimates of the likelihood of toxic behavior. When no pre-existing predictive model of toxic behavior is available, one must be estimated in real-time. To this end, we propose a contextual bandit algorithm that uses a small set of variables, selected based on domain expertise, to guide monitoring decisions. This algorithm balances exploration and exploitation to optimize long-term performance and is designed intentionally for easy deployment in production environments. Using data from the popular first-person action game *Call of Duty®: Modern Warfare®III*, we show that our algorithm consistently outperforms baseline algorithms that rely solely on individual players' past behavior, achieving improvements in detection rate of up to 24.56 percentage points or 51.5%. These results have substantive implications for the nature of toxicity and illustrate how domain expertise can be harnessed to help video game service operators detect and address toxicity, ultimately fostering a safer and more enjoyable gaming experience.

INDEX TERMS *Call of Duty®: Modern Warfare®III*, competitive online video games, contextual bandit algorithms, toxicity detection.

I. INTRODUCTION

Toxicity in competitive online video games has well-documented and widely recognized detrimental effects, including reduced user engagement and potential harm to psychological well-being [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13]. These prompt video game service operators to take proactive measures to monitor and address such behavior. However, these efforts are resource-intensive. Consequently, service operators seek to direct their interventions where toxicity is most prevalent, thereby maximizing their impact.

This article considers the problem of efficient sampling for toxicity detection in competitive online video games. By leveraging advanced computational methods to enhance toxicity detection, we aim to equip video game service operators with tools to detect and eventually address toxicity more effectively while aligning these efforts with their resource constraints. The ultimate goal is to create a safer, more enjoyable gaming experience for players.

We focus our attention on a setting wherein video game service operators must choose whether to monitor each player's in-game voice interactions at the start of every match. In doing so, their objective is to maximize the detection of toxic behavior while minimizing monitoring costs, as measured by the volume of monitored voice interactions. In this context, efficiency requires that no alternative policy can achieve a higher detection rate at equal cost or the same detection rate at a lower cost. To meet this objective, service operators seek to monitor only interactions where toxicity is sufficiently likely to occur.

To make informed monitoring decisions, video game service operators need estimates of the likelihood of toxic behavior. However, a predictive model of toxic behavior may

The associate editor coordinating the review of this manuscript and approving it for publication was Bhaskar P. Rimal.

not always be available, such as immediately after a game's release. In these cases, service operators must estimate a predictive model in real-time, a process inherently involving some exploration. In this context, service operators may find it valuable to monitor players' in-game voice interactions not only when they are confident that toxic behavior is likely, making optimal monitoring decisions based on existing data, but also when uncertainty is high, even if toxicity seems *a priori* unlikely, to gather more data and improve the accuracy of future predictions. Throughout this process, the ultimate goal of service operators is to balance exploration and exploitation in order to optimize long-term performance.

To address the tension between static and dynamic incentives, we propose a contextual bandit algorithm that adaptively and dynamically learns where to optimally allocate computational resources for toxicity detection. This algorithm determines which players' in-game voice interactions to monitor to maximize the detection of toxic behavior while minimizing monitoring costs, that is, by monitoring the fewest players in the fewest matches necessary. It bases its decisions on a handful of readily observable contextual features that are, according to domain expertise, associated with toxic behavior. It is designed intentionally for ease of deployment in production environments.

We compare the performance of our proposed algorithm against two baseline rule-based algorithms that reflect standard practices in the video game industry. Whereas our algorithm leverages contextual features to inform decision-making, these baseline algorithms make decisions solely based on individual players' past behavior—specifically, whether they have previously engaged in toxic behavior. We perform this analysis within the context of the popular first-person action video game *Call of Duty®: Modern Warfare®III*, focusing specifically on its primary multiplayer game mode, Team Deathmatch. The results show that our proposed algorithm consistently outperforms the baseline algorithms, achieving considerable improvements in the detection rate of toxic behavior, with increases of up to 24.56 percentage points (pp.) or 51.5%.

Substantively, these findings imply that some contextual factors are strongly associated with a higher likelihood of players engaging in toxic behavior, allowing for the effective optimization of monitoring strategies. Remarkably, monitoring decisions based on these factors have a systematically better performance than those made solely based on individual players' past tendencies toward toxicity, challenging the view that toxicity is an inherently idiosyncratic phenomenon.

This article is structured as follows. We begin by discussing how our work relates to previous research. Next, we describe the data used in our analysis. We then define the optimization problem faced by video game service operators, review factors correlated with toxicity, and outline our proposed algorithm. Afterward, we present the results of an experiment that simulates and compares the performance of our proposed algorithm with two baseline algorithms. We conclude by discussing the implications of our findings and outlining potential avenues for future research.

II. RELATED WORK

This paper contributes to the growing research on reinforcement learning for content moderation on online platforms [14]. Like prior work in this area, we address the challenge of efficiently allocating costly detection resources, such as expensive computational models or human moderators, to potential content violations by combining diverse features and signals. Our primary contribution is applied: we demonstrate how video game service operators, drawing on their domain expertise, can implement a contextual bandit algorithm to optimize their detection and moderation efforts. Specifically, we adapt an algorithm initially designed for personalized news recommendation to the unique demands of toxicity detection in competitive online video games [15].

Our study also relates to the extensive literature on content moderation and the detection of undesirable behavior in online communities [16], [17], [18], [19], [20], [21], [22], [23], [24]. This work has highlighted the considerable challenges in detecting and addressing undesirable behavior at scale. In response, our study proposes a novel approach to improve the efficiency of these computationally demanding efforts, offering a potential solution to enhance their scalability.

Lastly, we draw on substantive findings from the literature on toxicity in competitive online video games to identify contextual features that may predict toxic behavior and inform monitoring decisions [5], [12], [13], [25], [26], [27], [28], [29], [30], [31], [32], [33]. These variables include skill level, disparities in skill between the player and others, the presence of teammates from the same party, the number of matches previously played in the current session, and moderation reports filed by or against the player. The variables are further described and discussed in Section V.

III. DATA

We analyze proprietary data from *Call of Duty*, a popular first-person action video game franchise published by Activision®. We focus on *Call of Duty: Modern Warfare III* and its most popular multiplayer game mode, Team Deathmatch. In this game mode, players are divided into two equally sized teams and compete to achieve the highest number of eliminations. After a brief pause, eliminated players reappear at a different location on the map. A team wins by reaching a predetermined elimination limit first or accumulating the most eliminations by the end of the match.

Since 2023, Activision has partnered with Modulate™, a startup developing intelligent voice technology to identify and combat online toxicity, and integrated its proprietary voice chat moderation technology, ToxMod™, into its gaming platforms [34]. ToxMod is a voice moderation technology that analyzes online speech for emotion, volume, transcribed content, intention, and other related signals to identify harmful or malicious content [35]. These signals are input

TABLE 1. Reward structure.

		Player Behavior	
		Toxicity	*No Toxicity*
Monitoring Decision	*Monitor*	$1-c$	$-c$
	Not Monitor	0	

into machine learning models that classify the primary type of harm present in an audio clip. The voice chat moderation technology's initial beta rollout began in North America on August 30, 2023, within *Call of Duty: Modern Warfare II* and *Call of Duty: Warzone*™, followed by a global release (excluding the Asia-Pacific region) that coincided with the launch of *Call of Duty: Modern Warfare III* on November 10, 2023. During our observation period, ToxMod only supported English.[1]

ToxMod offers exceptional data on toxic behavior in competitive online video games, serving as the basis of our analysis. Our dataset comprises data from a representative sample of matches in Team Deathmatch mode monitored by ToxMod during the first month following the game's launch, from November 10 to December 10, 2023. Our sample consists of 207,338,296 observations, each representing a player in a match, drawn from 15,644,547 matches and 8,798,876 players.[2] We categorize a player as having engaged in toxic behavior during a game if ToxMod flagged at least one of their voice interactions as toxic during that match, thereby binarizing ToxMod's output.

IV. PROBLEM FORMULATION

We formalize the decision problem faced by video game service operators in choosing whether to monitor a player's in-game voice interactions. We focus our attention on a setting wherein service operators can choose, at the start of every match, to monitor a player's in-game voice interactions for the duration of that match. Service operators seek to monitor in-game voice interactions to detect and ultimately address toxic behavior. Monitoring is costly and provides no actionable insight unless toxicity is detected. Consequently, service operators seek to maximize the detection of offenses while minimizing the volume of monitored in-game voice interactions.

We assume that if a video game service operator opts to monitor a player's in-game voice interactions, they incur a fixed cost $c > 0$, which erodes their reward. In turn, the service operator earns rewards that depend on the player's behavior: they receive a reward of 1 if toxic behavior is detected and a reward of 0 otherwise. On the other hand, if the service operator chooses not to monitor the player's voice interactions, they receive a fixed reward of 0 regardless of the player's behavior. Rewards conditional on the monitoring decision and the player's behavior are summarized in Table 1. Note that this reward structure treats false positives—players who are monitored despite not engaging in toxicity—and false negatives—players who engage in toxicity but are not monitored—asymmetrically. In particular, it penalizes the former but not the latter.

In this framework, video game service operators seek to selectively monitor players' in-game voice interactions based on their behavior. Specifically, they prefer to monitor a player's in-game voice interactions when they engage in toxicity and not to monitor them when they do not engage in such behavior. Given the uncertainty of the player's behavior, these preferences translate into the following static decision rule: it is optimal to monitor a player's in-game voice interactions when the likelihood of toxic behavior surpasses the cost of monitoring:

$$P(\text{Player engages in toxicity}) > c. \tag{1}$$

In other words, to optimize toxicity detection resources, video game service operators should monitor a player's in-game voice interactions only when the likelihood of toxic behavior exceeds some threshold, ensuring that the gathered data is likely to be actionable.

V. TOXICITY CORRELATES

To implement the static decision rule defined above, video game service operators need estimates of the likelihood of toxic behavior. Previous research on toxicity in competitive online video games has identified contextual features correlated with such behavior. Many of these features are easily observable before the beginning of a match, making them valuable for estimating the likelihood of toxic behavior. We can then use these predictions to inform and guide monitoring decisions.

Table 2 lists eight of these variables, describing their expected relationship with toxicity and supported by relevant references. Descriptive statistics for these variables and toxic behavior in our dataset are presented in Table 3. Table 4 presents the estimates of linear and logistic regressions of a player's behavior with these covariates. Note that the linear regression coefficient and standard error estimates are all scaled by a 10^{-6} factor to enhance interpretability.

Regression results indicate that all covariates, except for the average skill difference with teammates and the number of matches played in the current session, consistently exhibit a statistically significant relationship with toxic behavior. These results confirm that these variables can predict toxic behavior and, in turn, guide monitoring decisions.

VI. BANDIT ALGORITHMS

Conclusions drawn in the previous section on toxicity correlates were derived *a posteriori* from the entire dataset. However, such historical data may not always be available

[1]ToxMod now supports English, Spanish, and Portuguese, with French and German coming soon in upcoming game releases.

[2]To protect confidential business information, we cannot disclose the exact proportion of the universe represented by this sample. However, we can assure readers that the data was carefully sampled to be representative.

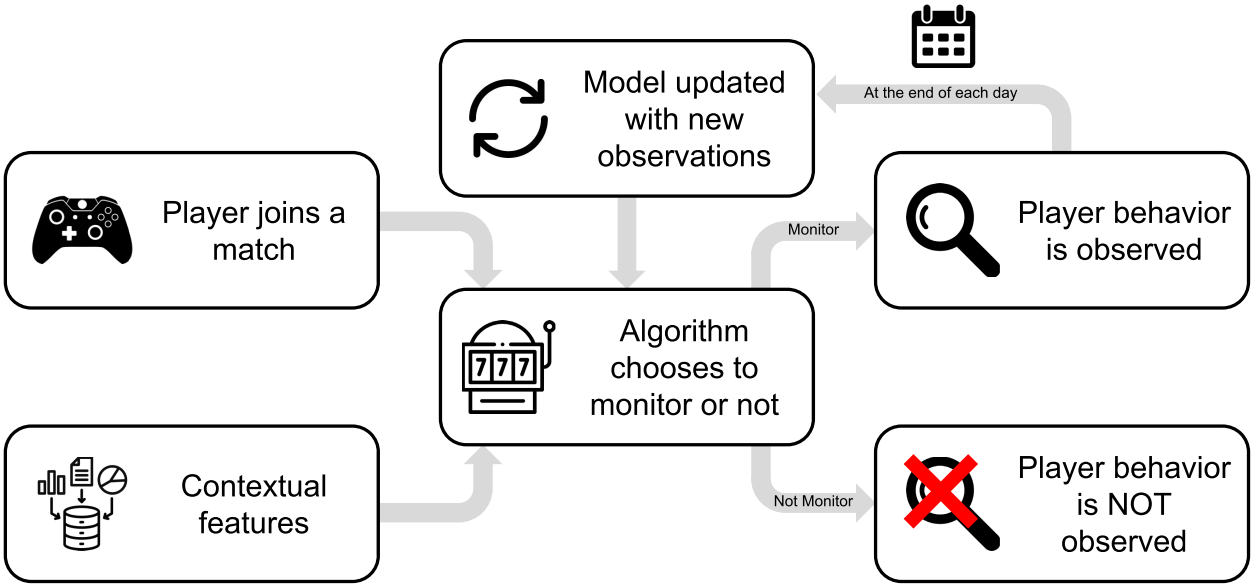

FIGURE 1. Algorithm workflow.

Algorithm 1 LinUCB Algorithm

Hyperparameters: $\delta \in \mathbb{R}_+$, $c \in \mathbb{R}_+$
$\boldsymbol{A} \leftarrow \boldsymbol{I}_d$ {d-dimensional identity matrix}
$\boldsymbol{b} \leftarrow \boldsymbol{0}_d$ {d-dimensional zero vector}
for $t = 0$ to $T - 1$ **do** {Loop over days}
  $\hat{\theta} \leftarrow \boldsymbol{A}^{-1}\boldsymbol{b}$ {Update coefficients daily}
  for $n = 0$ to $N_t - 1$ **do** {Loop over observations}
    Observe contextual features: $\boldsymbol{x}_{t,n} \in \mathbb{R}^d$
    $p_{t,n} \leftarrow \theta' \boldsymbol{x}_{t,n} + \delta\sqrt{\boldsymbol{x}'_{t,n}\boldsymbol{A}^{-1}\boldsymbol{x}_{t,n}}$
    Monitor if $p_{t,n} > c$, do not monitor otherwise
    if Monitor **then**
      Observe player behavior: $r_{t,n} \in \{0, 1\}$
      $\boldsymbol{A} \leftarrow \boldsymbol{A} + \boldsymbol{x}_{t,n}\boldsymbol{x}'_{t,n}$ {If monitored, update data}
      $\boldsymbol{b} \leftarrow \boldsymbol{b} + r_{t,n}\boldsymbol{x}_{t,n}$
    end if
  end for
end for

when predictions are needed, such as immediately after a game's release. In these cases, video game service operators must estimate a model of the likelihood of toxic behavior in real-time. This requires algorithms that can make monitoring decisions based on currently available data while continuously learning from new data to improve future decisions.

In addition to solving the static optimization problem outlined in Section IV, video game service operators face a dynamic trade-off when estimating models in real time: they may want to monitor players' in-game voice interactions not only when they are confident that toxic behavior is likely, in which case they *exploit* the available data, but also when uncertainty is high, in which case they *explore* to refine future predictions and improve future decisions. As service operators explore and collect more data, they seek to optimize long-term performance by exploiting as much information as possible. This exploration-exploitation trade-off is precisely the target of bandit algorithms.

For anyone familiar, there is a clear analogy between the optimization problem defined above and bandit problems: at the start of every match and for each player, video game service operators must choose whether to pull the "monitor" arm or the "not monitor" arm. The "monitor" arm generates stochastic rewards determined by the player's behavior, whereas the "not monitor" arm generates fixed known rewards.

We propose to make monitoring decisions based on the LinUCB algorithm, with the input covariates listed in Section V [15]. These variables were carefully selected based on domain expertise. We retain all features, including those with statistically insignificant coefficients, to emulate the performance of an algorithm that does not initially have access to the complete dataset. In contrast, the statistical significance of these features is assessed using all available data in that section. Retaining all covariates provides a conservative estimate of our algorithm's performance. Although excluding insignificant variables may improve performance, we defer such optimization to future work.[3] In that regard, regularization methods such as LASSO or elastic net could enable the model to select features endogenously, offering a promising direction for future study.

[3]The Supplementary Material includes a table presenting the results of a feature ablation study in which individual features were removed one at a time. The results reveal that omitting certain features can lead to modest improvements in our algorithm's performance.

TABLE 2. Toxicity correlates.

Variable	Expected Relationship with Toxicity	References
Skill Level	Experienced and highly skilled players may exhibit a greater tendency toward toxic behavior.	[25], [27], [30]
Average Skill Difference with Opponents	Players often display toxic behavior toward those they view as "outsiders," such as lower-skilled players.	[5], [28], [30]
Average Skill Difference with Teammates	Heightened competitiveness can contribute to increased toxicity.	[28], [30]
Presence of Teammates from the Same Party	Toxic behavior may occur more frequently in the presence of familiar individuals, such as teammates from the same party.	[12], [13], [25], [27]–[30]
Proportion of Teammates from the Same Party		
Matches Played in the Current Session	Players may be more likely to exhibit toxic behavior after playing many matches in a single session.	[26], [33]
Reports Filed Against the Player in the Last 24 Hours	Reports against players can indicate a history of toxic behavior, and those with a record of past toxicity are more likely to engage in such behavior in the future.	[12], [25], [31], [32]
Reports Filed by the Player in the Last 24 Hours	Players exposed to toxic behavior are more prone to engage in similar behavior, and filed reports may serve as markers of prior exposure to toxicity.	[5], [27], [32]

Note: A party consists of a group of players voluntarily playing together as a single, cohesive unit.

TABLE 3. Descriptive statistics.

Variable	Mean	Std. Dev.	Min.	Median	Max.
Toxic Behavior	0.000372	0.0193	0	0	1
Skill Level	−43.994	207.828	−736	−45	716
Average Skill Difference with Opponents	96.025	76.750	0	75	1,061.8
Average Skill Difference with Teammates	103.586	85.075	0	78.833	1,022.8
Presence of Teammates from the Same Party	0.336	0.472	0	0	1
Proportion of Teammates from the Same Party	0.104	0.185	0	0	1
Matches Played in the Current Session	3.751	5.840	0	2	150
Reports Filed Against the Player in the Last 24 Hours	0.0364	0.236	0	0	153
Reports Filed by the Player in the Last 24 Hours	0.0449	0.914	0	0	304

The LinUCB algorithm is formally outlined in Algorithm 1. Henceforth, let d denote the number of contextual features, t index the days (with a total of T), and n index the observations on each day (with N_t total observations on day t). Each observation represents a single player in a single match.

The algorithm models the expected reward from monitoring a player's in-game voice interactions or, in other words, the likelihood that a player engages in toxic behavior as a linear function of the covariates, with an unknown coefficient vector θ estimated using ridge regression:

$$\hat{\theta} = \left(\boldsymbol{X}'\boldsymbol{X} + \boldsymbol{I}\right)^{-1} \boldsymbol{X}'\boldsymbol{r}, \tag{2}$$

where $\boldsymbol{X}$ is the matrix formed by concatenating the covariate values from all previously monitored players and matches, and $\boldsymbol{r}$ is a vector indicating the presence of toxicity in each case.[4] The linear specification streamlines estimation and inference, simplifying the algorithm's deployment in production environments. It also promotes model parsimony, mitigating overfitting risks. Future research could explore non-linear methods, such as random forests or kernelized UCB [36], [37], though their application may be limited by computational challenges associated with our dataset's size.

For each new observation, the algorithm estimates the expected rewards from monitoring based on the data collected on previous days. The algorithm then applies an Upper Confidence Bound (UCB) arm-selection strategy: the "monitor"

[4]Note that Equation (2) can be evaluated even in the absence of historical data. In such cases, one can replace $\boldsymbol{X}$ with a zero vector.

TABLE 4. Regression results.

	(1) Linear†	(2) Logistic
Skill Level	0.046*** (0.001)	0.001*** (0.000)
Average Skill Difference with Opponents	-0.036*** (0.005)	-0.001*** (0.000)
Average Skill Difference with Teammates	-0.004 (0.005)	-0.000* (0.000)
Presence of Teammates from the Same Party	48.603*** (0.656)	1.566*** (0.011)
Proportion of Teammates from the Same Party	66.239*** (1.928)	0.697*** (0.020)
Matches Played in the Current Session	-0.145*** (0.024)	-0.001 (0.001)
Reports Filed Against the Player in the Last 24 Hours	62.794*** (1.520)	0.251*** (0.004)
Reports Filed by the Player in the Last 24 Hours	6.134*** (0.299)	0.033*** (0.001)

Note: † $\times 10^{-6}$; $^{*}p < 0.1$; $^{**}p < 0.05$; $^{***}p < 0.01$

arm is pulled if and only if the expected rewards plus the product of this expectation's standard error and a predefined exploration factor δ exceeds the monitoring cost c:

$$\underbrace{\boldsymbol{x}'_{t,n}\hat{\theta}}_{\text{Expected Rewards}} + \delta \underbrace{\sqrt{\boldsymbol{x}'_{t,n}\left(\boldsymbol{X}'\boldsymbol{X}+\boldsymbol{I}\right)^{-1}\boldsymbol{x}_{t,n}}}_{\text{Standard Error}} > c, \tag{3}$$

where $\boldsymbol{x}_{t,n}$ is the covariate vector for observation n on day t. If a player's behavior is monitored during a match, a binary variable $r_{t,n}$ encodes whether they acted in a toxic manner.

The exploration factor δ reflects the value that video game service operators place on gathering more information to reduce predictive uncertainty and improve future decisions. A higher value of this parameter reflects a greater inclination for exploration. On the other hand, the cost parameter c captures how strongly service operators prefer to confine monitoring to contexts with a high likelihood of toxicity. It regulates the volume of monitored observations, with higher values resulting in less monitoring.

Instead of updating the model coefficients after every observation, the model coefficients are updated at the end of each day based on the observations collected throughout that day. The updated model is then used to inform monitoring decisions for the next day. Daily updates make the model easier to deploy in production environments. In particular, daily updates enable batch processing, significantly reducing the computational costs of updating the model coefficients. While this approach still supports learning over time, it may slow the overall learning speed. Increasing the frequency of updates (e.g., hourly updates) can mitigate this by enabling the model to learn more quickly from new data. However, our analysis suggests that hourly updates offer negligible performance improvements.[5] The workflow of the algorithm is illustrated in Figure 1.

This bandit algorithm addresses the "cold-start problem" encountered by video game service operators upon a game's release. With no pre-existing model to predict toxic behavior, one must be estimated in real-time, which inherently involves some exploration. In this context, monitoring a player's voice interactions can still be valuable even when the immediate cost exceeds the expected rewards as long as the collected data makes future predictions and decisions more accurate. Over time, as more data is collected, the algorithm gradually shifts toward exploitation, relying primarily on expected rewards to make monitoring decisions. However, when the algorithm lacks enough information to make an informed decision for a specific set of contextual features, it retains the option to keep exploring. Also, by continually updating model coefficients, the algorithm adapts to evolving conditions.

We compare the performance of our proposed LinUCB algorithm against two baseline algorithms that reflect current practices in the video game industry: the deterministic and probabilistic Explore-Then-Commit algorithms [38]. Unlike LinUCB, which optimizes decisions by pooling information across all players, these algorithms identify optimal monitoring decisions based on individual players' prior history of toxic behavior. This is based on the premise that past toxic players are more likely to engage in similar behavior in the future [12], [25], [31], [32]. The deterministic Explore-Then-Commit algorithm monitors each player for a fixed and predetermined number of matches and continues monitoring a player beyond this probationary period if they are observed to engage in toxic behavior at least once. In contrast, the probabilistic Explore-Then-Commit algorithm randomly monitors a fixed share of observations and continues monitoring a player if they are observed to engage in toxic behavior at least once. This algorithm is equivalent to an ε-greedy algorithm: with probability ε, service operators explore by monitoring, and with probability $1-\varepsilon$, their decision to monitor is based on the player's observed past behavior.

VII. EXPERIMENTAL METHODOLOGY

We simulate the performance of our proposed algorithm and that of baseline algorithms on the dataset described in Section III. In this experiment, only a subset of the matches and players is monitored, with monitoring decisions made dynamically by the bandit algorithms described earlier. These algorithms rely solely on the data available at the time of decision-making. On the other hand, their performance is assessed on future data that we, as analysts, can observe but remains unobserved by the algorithms unless and until they choose to observe the corresponding match and player.

[5] The Supplementary Material includes a table comparing the performance of our proposed LinUCB algorithm under hourly and daily update schedules. The results indicate that the algorithm performs nearly identically in both scenarios.

TABLE 5. Bandit algorithms performance comparison.

Share of Monitored Observations		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Detection Rate										
Deterministic Explore-Then-Commit		0.1008	0.2163	0.3402	0.4575	0.5776	0.6881	0.7889	0.8806	0.9567
Probabilistic Explore-Then-Commit		0.2196	0.3634	0.4769	0.5751	0.6604	0.7383	0.8112	0.8779	0.9405
LinUCB		0.3202	0.55	0.7225	0.8035	0.8496	0.8897	0.9219	0.9506	0.9792
Improvement in Detection Rate										
Deterministic Explore-Then-Commit → *LinUCB*	*pp.*	21.94	33.37	38.23	34.60	27.20	20.16	13.30	7.00	2.25
	%	217.66	154.28	112.28	75.63	47.09	29.30	16.86	7.95	2.35
Probabilistic Explore-Then-Commit → *LinUCB*	*pp.*	10.06	18.66	24.56	22.84	18.92	15.14	11.07	7.27	3.87
	%	45.81	51.35	51.5	39.71	28.65	20.51	13.65	8.28	4.11

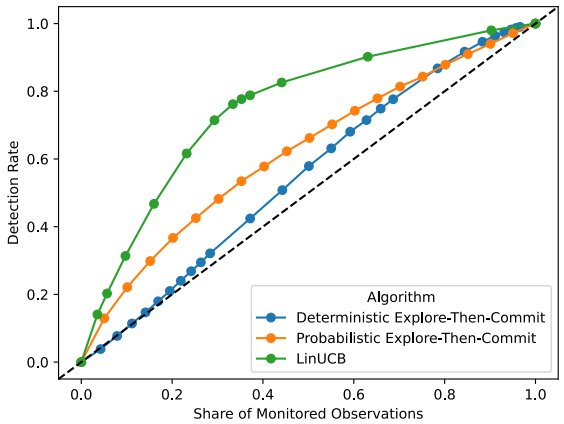

FIGURE 2. Bandit algorithms performance.

This approach amounts to dynamically shifting training and evaluation data, mirroring real-world constraints whereby bandit algorithms can only learn from previously collected data.

We compare the performance of the different algorithms in selecting which players and matches to monitor. To reflect the objectives of video game service operators, we assess algorithm performance based on detection rate or recall, defined as the share of toxic behavior detected (after being monitored) relative to all toxic behavior that occurred. We can calculate this metric because, as analysts, we have access to the complete dataset, including occurrences of toxicity not monitored by the algorithms in our simulation. For a given proportion of matches and players monitored, the optimal algorithm is the one that maximizes the detection rate.

VIII. RESULTS

Figure 2 illustrates the detection rate of toxic behavior for each algorithm as a function of the proportion of monitored observations. Each point corresponds to a different value of the exploration parameter δ for LinUCB, a different number of exploration steps for the deterministic Explore-Then-Commit algorithm, and a different exploration probability for the probabilistic Explore-Then-Commit algorithm. The exploration parameter for LinUCB was tuned for convenience, and the results are not sensitive to its exact value. The dashed diagonal line represents the performance of uniform random sampling.

Our proposed LinUCB algorithm consistently outperforms the baseline algorithms for any given share of monitored observations. The probabilistic Explore-Then-Commit algorithm ranks second, except for the highest share of monitored observations, for which the deterministic Explore-Then-Commit algorithm ranks second.

Table 5 presents a detailed comparison of the performance of our proposed algorithm against baseline algorithms for different shares of monitored observations. The results for the second-best alternative to our proposed algorithm are highlighted in grey.

Again, the performance of our proposed LinUCB algorithm is consistently superior to baseline algorithms, increasing the detection rate by up to 24.56 pp. or 51.5%. In other words, holding monitoring costs constant, our proposed algorithm can detect up to 51.5% more offenses than the second-best alternative, providing an equivalent volume of additional actionable insights.

We note that the detection rate increases with the share of monitored observations for all algorithms. However, the marginal gains in detection rate diminish as the share of monitored observations increases, meaning that detecting more offenses becomes increasingly costly. Finally, the absolute and relative improvements in detection rate achieved by our proposed algorithm follow an inverted-U shape, decreasing as monitoring coverage expands, except for the lowest monitoring levels.

IX. DISCUSSION AND CONCLUSION

In this paper, we have considered the problem of efficient toxicity detection in competitive online video games. Using data from the popular first-person action video game *Call of Duty: Modern Warfare III*, we simulated the performance of various bandit algorithms in optimizing monitoring decisions. Our proposed LinUCB algorithm, optimizing monitoring decisions based on a small set of contextual features, consistently outperformed random sampling and two baseline rule-based algorithms that reflect standard

practices in the video game industry. Therefore, our proposed algorithm can considerably enhance the efficiency of toxicity detection and support video game service operators in fostering a safer and more enjoyable gaming environment. The design of our algorithm prioritizes ease of deployment, streamlining its practical implementation at scale.

The superior performance of our contextual algorithm over benchmark algorithms that rely on individual players' history of toxic behavior has substantive implications for the nature of toxicity. One perspective views toxicity as an inherently idiosyncratic phenomenon, largely independent of context, with a small set of players spontaneously and repeatedly engaging in such behavior. An alternative perspective sees toxicity as the reflection of specific contextual factors that nudge players toward toxic behavior. Our findings challenge the first perspective by revealing that: (i) a handful of contextual features are strongly associated with an increased likelihood of toxic behavior, and (ii) monitoring decisions made based on these factors are more effective than those based on individual players' history of toxic behavior.

To conclude, multiple avenues exist for expanding on this study. For instance, future research should explore additional covariates to improve the accuracy of toxic behavior predictions, including accounting for past moderation actions. Evaluating our proposed algorithm's performance beyond one month and across video games from different genres would also be valuable. On this point, we anticipate that various games and genres will require different features to predict toxicity accurately. Lastly, further investigating the application of bandit algorithms to enhance and optimize human moderation efforts, arguably even more costly than automated toxicity detection, offers an exciting direction for further research.

ACKNOWLEDGMENT

The authors extend their gratitude to Andrea Boonyarungsrit, Grant Cahill, MJ Kim, Jonathan Lane, Amine Mahmassani, Myrl Marmarelis, Gary Quan, Deshawn Sambrano, Feri Soltani, and Michael Vance for invaluable feedback and support in writing this article. The views and opinions expressed in this article are solely those of the authors and do not reflect those of Activision®.

REFERENCES

[1] S. Caplan, D. Williams, and N. Yee, "Problematic internet use and psychosocial well-being among MMO players," *Comput. Hum. Behav.*, vol. 25, no. 6, pp. 1312–1319, Nov. 2009.

[2] K. L. Gray, "Deviant bodies, stigmatized identities, and racist acts: Examining the experiences of African-American gamers in xbox live," *New Rev. Hypermedia Multimedia*, vol. 18, no. 4, pp. 261–276, Dec. 2012.

[3] A. Salter and B. Blodgett, "Hypermasculinity & dickwolves: The contentious role of women in the new gaming public," *J. Broadcast. Electron. Media*, vol. 56, no. 3, pp. 401–416, Jul. 2012.

[4] J. H. Kuznekoff and L. M. Rose, "Communication in multiplayer gaming: Examining player responses to gender cues," *New Media Soc.*, vol. 15, no. 4, pp. 541–556, Jun. 2013.

[5] J. Fox and W. Y. Tang, "Sexism in online video games: The role of conformity to masculine norms and social dominance orientation," *Comput. Hum. Behav.*, vol. 33, pp. 314–320, Apr. 2014.

[6] S. Chess and A. Shaw, "A conspiracy of fishes, or, how we learned to stop worrying about #GamerGate and embrace hegemonic masculinity," *J. Broadcast. Electron. Media*, vol. 59, no. 1, pp. 208–220, Jan. 2015.

[7] H. Kwak, J. Blackburn, and S. Han, "Exploring cyberbullying and other toxic behavior in team competition online games," in *Proc. 33rd Annu. ACM Conf. Human Factors Comput. Syst.*, 2015, pp. 3739–3748.

[8] M. E. Ballard and K. M. Welch, "Virtual warfare: Cyberbullying and cyber-victimization in MMOG play," *Games Culture*, vol. 12, no. 5, pp. 466–491, Jul. 2017.

[9] D. Madden, Y. Liu, H. Yu, M. F. Sonbudak, G. M. Troiano, and C. Harteveld, "'Why are you playing games? You are a girl!': Exploring gender biases in esports," in *Proc. CHI Conf. Human Factors Comput. Syst.*, May 2021, pp. 1–15.

[10] S. Türkay, J. Formosa, S. Adinolf, R. Cuthbert, and R. Altizer, "See no evil, hear no evil, speak no evil: How collegiate players define, experience and cope with toxicity," in *Proc. CHI Conf. Human Factors Comput. Syst.*, Apr. 2020, pp. 1–13.

[11] R. Kowert and E. Kilmer. (2023). *Toxic Gamers Are Alienating Your Core Demographic: The Business Case for Community Management*. [Online]. Available: https://www.takethis.org/wp-content/uploads/2023/08/ToxicGamersBottomLineReport_TakeThis.pdf

[12] Á. Zsila, R. Shabahang, M. S. Aruguete, and G. Orosz, "Toxic behaviors in online multiplayer games: Prevalence, perception, risk factors of victimization, and psychological consequences," *Aggressive Behav.*, vol. 48, no. 3, pp. 356–364, May 2022.

[13] J. Morrier, A. Mahmassani, and R. M. Alvarez, "Uncovering the effect of toxicity on player engagement and its propagation in competitive online video games," 2024, *arXiv:2407.09736*.

[14] V. Avadhanula, O. A. Baki, H. Bastani, O. Bastani, C. Gocmen, D. Haimovich, D. Hwang, D. Karamshuk, T. Leeper, J. Ma, G. Macnamara, J. Mullett, C. Palow, S. Park, V. S. Rajagopal, K. Schaeffer, P. Shah, D. Sinha, N. Stier-Moses, and P. Xu, "Bandits for online calibration: An application to content moderation on social media platforms," 2022, *arXiv:2211.06516*.

[15] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," 2010, *arXiv:1003.0146*.

[16] Y. Chen, Y. Zhou, S. Zhu, and H. Xu, "Detecting offensive language in social media to protect adolescent online safety," in *Proc. Int. Conf. Privacy, Secur., Risk Trust Int. Confernece Social Comput.*, Sep. 2012, pp. 71–80.

[17] M. Märtens, S. Shen, A. Iosup, and F. Kuipers, "Toxicity detection in multiplayer online games," in *Proc. Int. Workshop Netw. Syst. Support Games (NetGames)*, Dec. 2015, pp. 1–6.

[18] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang, "Abusive language detection in online user content," in *Proc. 25th Int. Conf. World Wide Web*, Apr. 2016, pp. 145–153.

[19] H. Zhong, H. Li, A. Squicciarini, S. Rajtmajer, C. Griffin, D. J. Miller, and C. Caragea, "Content-driven detection of cyberbullying on the Instagram social network," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, Jul. 2016, pp. 3952–3958.

[20] S. Malmasi and M. Zampieri, "Detecting hate speech in social media," in *Proc. Recent Adv. Natural Lang. Process. (RANLP)*, Nov. 2017, pp. 467–472.

[21] C. Van Hee, G. Jacobs, C. Emmery, B. Desmet, E. Lefever, B. Verhoeven, G. De Pauw, W. Daelemans, and V. Hoste, "Automatic detection of cyberbullying in social media text," *PLoS ONE*, vol. 13, no. 10, Oct. 2018, Art. no. e0203794.

[22] A. Sanzgiri, D. Austin, K. Sankaran, R. Woodard, A. Lissack, and S. Seljan, "Classifying sensitive content in online advertisements with deep learning," in *Proc. IEEE 5th Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2018, pp. 434–441.

[23] Cambridge Consultants. (2019). *Use of AI in Online Content Moderation*. [Online]. Available: https://www.ofcom.org.uk/siteassets/resources/documents/research-and-data/online-research/other/cambridge-consultants-ai-content-moderation.pdf?v=324081

[24] P. Gampa, A. A. Valsangkar, and S. Choubey, "Prioritised moderation for online advertising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2023, pp. 2004–2012.

[25] K. Grandprey-Shores, Y. He, K. L. Swanenburg, R. E. Kraut, and J. Riedl, "The identification of deviance and its impact on retention in a multiplayer game," in *Proc. 17th ACM Conf. Computer-Supported Cooperat. Work Social Comput.*, Feb. 2014, pp. 1356–1365.
[26] A. Chamarro, U. Oberst, R. Cladellas, and H. Fuster, "Effect of the frustration of psychological needs on addictive behaviors in mobile Videogamers—The mediating role of use expectancies and time spent gaming," *Int. J. Environ. Res. Public Health*, vol. 17, no. 17, p. 6429, Sep. 2020.
[27] B. Kordyaka, K. Jahn, and B. Niehaves, "Towards a unified theory of toxic behavior in video games," *Internet Res.*, vol. 30, no. 4, pp. 1081–1102, Jun. 2020.
[28] Y. Kou, "Toxic behaviors in team-based competitive gaming: The case of league of legends," in *Proc. Annu. Symp. Computer-Human Interact. Play*, Nov. 2020, pp. 81–92.
[29] D. McLean, F. Waddell, and J. Ivory, "Toxic teammates or obscene opponents? Influences of cooperation and competition on hostility between teammates and opponents in an online game," *J. Virtual Worlds Res.*, vol. 13, no. 1, pp. 1–15, Mar. 2020.
[30] C. Shen, Q. Sun, T. Kim, G. Wolff, R. Ratan, and D. Williams, "Viral vitriol: Predictors and contagion of online toxicity in world of tanks," *Comput. Hum. Behav.*, vol. 108, Jul. 2020, Art. no. 106343.
[31] N. A. Beres, J. Frommel, E. Reid, R. L. Mandryk, and M. Klarkowski, "Don't you know that you're toxic: Normalization of toxicity in online gaming," in *Proc. CHI Conf. Human Factors Comput. Syst. (CHI)*, 2021, pp. 1–15.
[32] R. Kocielnik, Z. Li, C. Kann, D. Sambrano, J. Morrier, M. Linegar, C. Taylor, M. Kim, N. Naqvie, F. Soltani, A. Dehpanah, G. Cahill, A. Anandkumar, and R. M. Alvarez, "Challenges in moderating disruptive player behavior in online competitive action games," *Frontiers Comput. Sci.*, vol. 6, pp. 1–11, Feb. 2024.
[33] A. Kumar, S. Dodda, N. Kamuni, and V. S. M. Vuppalapati, "The emotional impact of game duration: A framework for understanding player emotions in extended gameplay sessions," 2024, *arXiv:2404.00526*.
[34] Activision Publishing. (2023). *Call of Duty Takes Aim At Voice Chat Toxicity, Details Year-to-Date Moderation Progress*. [Online]. Available: https://www.callofduty.com/blog/2023/08/call-of-duty-modern-warfare-warzone-anti-toxicity-progress-report
[35] R. Kowert and L. Woodwell. (2022). *Moderation Challenges in Digital Gaming Spaces: Prevalence of Offensive Behaviors in Voice Chat*. [Online]. Available: https://www.takethis.org/wp-content/uploads/2022/12/takethismodulatereport.pdf
[36] M. Dimakopoulou, Z. Zhou, S. Athey, and G. Imbens, "Estimation considerations in contextual bandits," 2017, *arXiv:1711.07077*.
[37] H. Zenati, A. Bietti, E. Diemert, J. Mairal, M. Martin, and P. Gaillard, "Efficient kernel UCB for contextual bandits," 2022, *arXiv:2202.05638*.
[38] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2020.

JACOB MORRIER received the B.Sc. degree in economics and financial engineering from the Université Paris-Dauphine-PSL, in 2018, and the B.Sc degree the Université du Québec à Montréal, in 2019, and the M.S. degree in social science from California Institute of Technology (Caltech), in 2021, where he is currently pursuing the Ph.D. degree. His work has been published or is forthcoming in several leading academic journals, such as *The Journal of Politics*, *Political Analysis*, *Political Science Research and Methods*, *Political Research Quarterly*, *American Politics Research*, *PLOS One*, and *Frontiers in Computer Science*. His research interests span widely in economics, political science, and quantitative social science. He specializes in developing and applying advanced quantitative methodologies, including causal inference, econometrics, and machine learning, to analyze administrative, behavioral, and textual data. In recognition of his academic excellence, he was awarded the prestigious Canadian Governor General's Academic Medal, in 2019, for graduating as the top undergraduate student in his class.

RAFAL KOCIELNIK received the M.Sc. degree in computer science, Poland, the PDEng. degree in industrial design, The Netherlands, and the Ph.D. degree in human–centered design and engineering, USA. He is currently a Postdoctoral Researcher with the Caltech's Department of Computing and Mathematical Sciences, specializing in Human–Centered AI. He has more than 60 peer-reviewed publications and collaborations with Microsoft, NVIDIA, and Activision. His graduate research focused on designing engaging conversational interactions for reflection and behavior change. At Caltech, he works on AI for surgical training, social bias in generative AI, and combating online toxicity, work for which he has earned a 2020 CRA Computing Innovation Fellowship. He has also received recognition from CSCW, CUI, ML4H, Nature Digital Medicine, and JAMA.

R. MICHAEL ALVAREZ is currently the Flintridge Foundation Professor of political and computational social science with Caltech and the Co-Director of the Caltech/MIT Voting Technology Project. At Caltech, he is a Faculty Member with in Social Sciences, and an Associated Faculty Member with the Social and Decision Neuroscience Program. He recently became the Founding Co-Director of Caltech's Ronald and Maxine Linde Center for Science, Society, and Policy. He is currently the Faculty Liaison to Caltech's men's basketball program and is active in the community, working mainly with educational institutions, and non-profits. He has been a recognized for my mentoring work, both at Caltech by the Graduate Student Council (twice) and by the Society for Political Methodology. He is a fellow of the Society for Political Methodology and of American Academy of Arts and Sciences.