Multi-agent Perspective of Fake Feedback Attacks on Stochastic Multi-armed Bandits

Charles A. N. Costa[1], **Célia Ghedini Ralha**[1]

[1]Computer Science Department – University of Brasília (UnB)
Campus Dary Ribeiro – Asa Norte – 70.910-900 – Brasília – DF – Brazil

charles.costa@aluno.unb.br, ghedini@unb.br

Abstract. *The problem of false feedback attacks on stochastic Multi-Armed Bandits (MAB) algorithms is the focus of this article from the perspective of Multi-Agent Systems (MAS). We present the roles, beliefs, interactions, and goals of the agents involved in false feedback attacks according to performance metrics to evaluate the MAB algorithms' performance. Three types of attacks illustrate the problem: constant, adaptive, and Jun's adversarial attacks relaxed version, specifically built to exploit vulnerabilities in the e-Greedy and UCB1 algorithms. Experiments exploiting the vulnerabilities inherent in stochastic MAB algorithms present results revealing the useful characteristics for those who design MAS requiring defenses against false feedback attacks.*

1. Introduction

Multi-Armed Bandits (MAB) are online machine-learning algorithms designed to deal with the trade-off between exploration and exploitation. On the metaphorical version of the problem, a player faces a set of levers, usually called arms, each potentially returning a different reward. The player has no previous knowledge about the arms. With a limited number of tries at her disposal, the player has to strategically decide which arm to pull at each step of the game to maximize the accumulated reward while acquiring knowledge about arms' reward functions. The problem appears in many real-world situations, such as the administration of drugs to patients [Bastani and Bayati 2020], showing advertising to website visitors [Shen et al. 2015], and even deciding which stock to buy [Schwartz et al. 2017]. Overall, [Bouneffouf et al. 2020] provides a comprehensive overview of practical applications of MAB algorithms, also discussing their potential to help advance machine learning in many domains.

A way to classify MAB algorithms is by the set of assumptions they make about the reward functions. The stochastic MAB algorithms assume that rewards are sampled from stationary distributions [Slivkins 2019]. There exist many stochastic MAB algorithms that implement different strategies to balance between the exploration and exploitation phases. In this work, we tested two commonly used ones, the ϵ-Greedy and the UCB1. The ϵ-Greedy algorithm uses a constant ϵ in the range [0,1], which is the probability of using exploration. At each round, there is a probability of $\epsilon - 1$ that the learner will choose an option by chance and a ϵ that it will make the greedy decision, i.e., picking the option with the highest average reward [Vermorel and Mohri 2005]. The UCB1 estimates arm rewards' confidence intervals and selects the one with the highest upper bound. Since confidence intervals' lengths are inversely proportional to the sample set length, with sufficient time, UCB1 will select every arm at least once [Auer et al. 2002].

When applicable, stochastic MAB algorithms provide robust results hardly surpassed by other approaches. However, it is well-known that stochastic MABs are vulnerable to data poisoning attacks. Data poisoning attacks are the corruption of reward information for MAB choices. Some works have approached the problem of poisoned stochastic rewards. [Lykouris et al. 2018] cited e-commerce fake reviews patronized by competitors and other non-malicious sources of corruption. [Niss and Tewari 2020] is motivated by the application of bandits in education, where reward information derives from human opinion, thus prone to deviation. [Jun et al. 2018] is motivated by the industrial application of contextual bandits [Li et al. 2010] usually employed in recommender systems. [Vallée et al. 2014] noted that some algorithms robust to one-source corruption are vulnerable to collusion of agents. [Guan et al. 2020] claims that some popular algorithms fail even when attacks are not large all the time. We argue that a fake feedback attack study helps design adequate defenses, especially for Multi-Agent Systems (MAS) design and development.

Although the sets for studying those attacks and defenses are populated by agents performing diverse roles, very few works approach the problem of fake feedback manipulative attacks to stochastic bandits as a multi-agent problem. A multi-agent definition of the problem would help researchers respond to some of the questions related to this scenario, i.e., which roles do agents perform, what capacities must the agents involved have, which information must they pursue, and what are their motivations and commitments, among other questions. The main contribution of this paper is the experimental exploration and analysis of attacks on stochastic MAB from a multi-agent perspective. It presents practical but effective attacks on stochastic MAB, followed by their experimental evaluation. It sheds light on challenges and opportunities related to weak attacks on stochastic MAB within an agent framework.

The rest of this article is as follows: Section 2 presents a multi-agent definition of the problem, Section 3 presents an overview of attacks on stochastic MAB, Section 4 presents the performance measures used on three stochastic MAB attacks, Section 5 brings our experimental outcome and analysis, and Section 6 the conclusion.

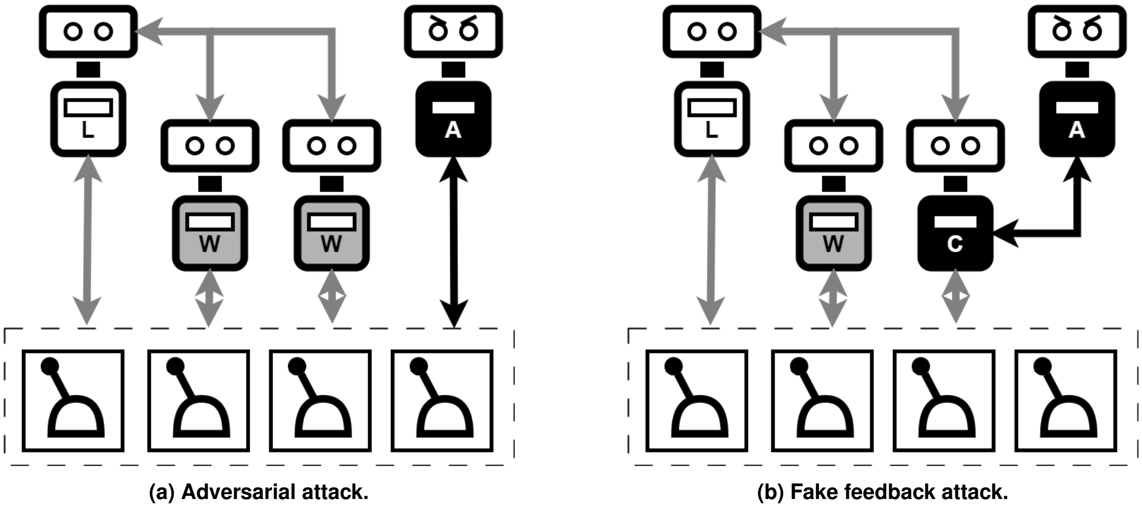
2. Multi-agent Perspective

MAS are systems where agents sharing the same environment act autonomously to achieve common or conflicting goals, in most practical cases, on behalf of users [Wooldridge 2009]. To provide a MAS perspective on the problem, we define the roles, goals and beliefs database, required to reach an understanding.

Following the classification system described in [Wooldridge 2009], the stochastic MAB environment is non-deterministic, episodic, static, and discrete. In this work, we assume no noise or communication cost, which means that any agent can freely communicate, and received messages are equal to the sent ones, despite the communication cost and noise can influence attack efficiency in the real world. We also take that more than one agent can draw the same arm in a round without blocking.

We represent our problem as a game played by a set of agents. As we can see in

Figure 1b, the agents are the Learner (l), the Attacker (a), and a set of Witnesses (W). The set of all agents AG is $\{l, a\} \cup W$. We represent roles and individual agents in lowercase and sets by capital letters, except in Figure 1 for readability. AG's possible actions are drawing arms to collect rewards, communicate, and read information from the environment. The set of arms K is $\{k_1, k_2, ..., k_n\}$. The witnesses are other agents playing in the same environment from whom the Learner can obtain feedback reports about past arm trials. Witnesses' feedback reports help the Learner to evaluate the arms. The co-opted witnesses (C), a subset of W, respond to a's instructions to corrupt the information sent to l. The members of C receive messages from a about the level of corruption they should introduce in the rewards. It is worth noting that $|C| < |W|$, i.e., at least one witness is honest.



(a) Adversarial attack.

(b) Fake feedback attack.

Figure 1. Comparing adversarial attack (1a), and fake feedback attack scenarios (1b). Agents are: Learner (L), Attacker (A), Witnesses (W), and Co-opted witness (C).

Learner

The Learner's goal is to collect rewards from the arms while minimizing regret. We define regret measure in Section 4 (Performance measures). The Learner maintains a belief database with all reward information, including rewards obtained by itself and that informed by the witnesses. Equation 1 describes the Learner's belief database (B_l), where t is the round time, α is the agent informing its collected reward, k is the pulled arm, r is the reward.

$$B_l = \{(t, \alpha, k, r) | t \geq 1, \alpha \in AG, k \in K, r \in [0, 1]\} \tag{1}$$

Based on B_l state in t, the Learner employs a policy π which returns the arm that l must draw. The policy π must be defined in $t = 0$, when $B_p = \emptyset$.

$$\pi(B_l, t) = k_t \tag{2}$$

In experiments, π is to be substituted by an MAB policy.

Attacker

The Attacker's goal is to maximize the number of times the Learner pulls a sub-optimal target arm. The Attacker can observe each arm's estimated average reward and Learner's pull count. Equation 3 describes the Attacker's belief database, where $\hat{\mu}$ is the estimated average reward of $k \in K$, and n is the pull count of k.

$$B_a = \{(t, k, \mu, n) | t \geq 1, k \in K, \mu \in [0, 1], n \geq 0\} \tag{3}$$

Thus, based on its belief database, the Attacker applies a policy ρ to calculate the level of corruption that the Co-opted Witnesses will insert into the feedback provided to the Learner, where k_t is the target arm ($k_t \in K$). The ρ function is the fake feedback attack policy.

$$\rho(B_a, k_t, t) = \{(k, c_{t+1}) | k \in K\} \tag{4}$$

Witnesses and Co-opted Witnesses

For generality, we refrain from clearly defining Witnesses' goals and policies. Instead, we describe the Witnesses' behavior in three steps. During each round, a witness will draw an arm, collect the reward, and respond to the Learner with the required information. Witnesses' belief database only stores information about their collected rewards, as described in Equation 5.

$$B_w = \{(t, k, r) | t \geq 1, k \in K, r \in [0, 1]\} \tag{5}$$

However, Co-opted Witnesses have secretly the goal of helping the attacker to manipulate the Learner. They execute an additional step before responding to the Learner, which is receiving the new corruption level to insert in the information provided to the Learner. Thus, Co-opted Witnesses must maintain the current corruption level informed by the Attacker in their database, like in Equation 6.

$$B_c = \{c | c \in [0, 1]\} \cup \{(t, k, r) | t \geq 1, k \in K, r \in [0, 1]\} \tag{6}$$

Interaction Protocol

Here, the protocol of the T-round game is described. The protocol is executed at each $1 \leq t \leq T$. We omit the steps for receiving and sending messages and storing information in the belief databases for brevity. All databases are initialized as empty sets.

1. l applies $\pi(B_l, t)$ to obtain k_t.
2. l draws k_t and collects r_{K_t}.
3. l requires from W feedback reports.
4. a collects the information about K.
5. a applies $\rho(B_a, k_t, t)$ to calculate c_{t+1}.
6. a informs C about c_{t+1}.
7. each w in W draws an arm at random and collects the reward.
8. each c in C add the informed c_{t+1}.

3. Attacks on Stochastic MAB

Manipulative attacks on MAB refer to strategies to manipulate them into increasing the counting pull of a specific arm, frequently a sub-optimal arm, in a way that affects the system's performance or justice. In adversarial attacks, an attacker has control of the reward perceived by the Learner agent [Rangi et al. 2022]. Let us use the slot machine metaphor to explain how adversarial attacks work. When a player pulls a chosen machine, a mechanism informs the adversary of the player and the machine's identities. Then, the adversary will select in a control board which prizes the player will receive, if any.

Compared to adversarial attackers, weak attackers have two flaws: they cannot corrupt all reward reports since some are secured; they have to decide about the corruption before the actual reward is revealed [Rangi et al. 2022]. The weak attacker cannot access the machine prizing mechanism of the slot machines. However, it can co-opt witnesses and instruct them in advance to poison the report provided to the Learner.

Poisoning attacks to stochastic MAB are those in which the attacker corrupts the reward information by adding a carefully calculated value to the actual reward to manipulate the Learner to make decisions in a specific way [Liu and Shroff 2019]. When the weak attack employs Co-opted Witnesses to corrupt the reports provided to the Learner, it is called a fake feedback attack. Figure 1a shows information flow on adversarial attacks and fake-feedback attacks (Figure 1b), where the Attacker (A) counts on Co-opted Witnesses (C) to corrupt the information provided to the Learner (L).

Fake feedback presents a problem for many online, open, and dynamic real-world environments. Let us take the reputation systems employed in many e-commerce platforms. Some sellers contract counterfeiters to insert manipulative reviews of products and sellers, whether to improve their reputation or harm competitors. The work on [Zhang et al. 2013] illustrates a history of generations of fake feedback attacks on the trust system of the Chinese e-commerce platform Taobao.

4. Weak Attacks on Stochastic MAB

This Section presents the performance measures used to evaluate the MAB algorithms' performance, followed by weak attacks on stochastic bandits examples that can succeed when no defense is present in the system.

Performance Measures

MAB algorithms' performance is evaluated using a metric called regret. Regret is the difference between the reward accumulated by pulling only the best arm overall and the real accumulated reward. According to [Vermorel and Mohri 2005], the regret at the round T is defined by Equation 7, where $\mu(k)$ is the mean of the rewards obtained from the arm $k \in K$, K the set of all available arms. Exists an k^* for which $\mu(k^*) \geq max_{k \in K}(\mu(k))$, that, for simplicity, we call $\mu(k^*)$, μ^*. Considering that there were T rounds, and $r_{k,L}(t)$ is the reward the Learner collected from the arm k at the round $t \leq T$.

$$R_L(T) = \mu^* \cdot T - \sum_{t=1}^{T} r_{k,L}(t) \tag{7}$$

Take $r^i_{k,w}(t)$ as the reward that a witness w provided to the Learner in round t. The corruption that the witness w inserted in the system is the difference between the reward it collected and the reward it informed. If the Attacker did not co-opt w, then the corruption it inserts is always zero. The total corruption level inserted into the game until the final round T is expressed by Equation 8.

$$C(T) = \sum_{w \in W} \sum_{t=0}^{T} |r^{i}_{k,w}(t) - r_{k,w}(t)| \tag{8}$$

Regarding the agent's objectives in the game, we define success by comparing scenarios when corruption exists or not. Take $N(k,t,c)$ as the counting of times an arm k was pulled by the Learner until some t when the Attacker submitted the game to a level of corruption c. We call Achieved Pulls (AP) the increase that the Attacker could achieve in the pull counting of the target arm using the policy ρ until the horizon T, as described in Equation 9, where k_t is the target arm.

$$AP(T) = N(k_t, T, C(T)) - N(k_t, T, 0) \tag{9}$$

The Attacker's cost for each additional pull is defined by Equation 10.

$$CP(T) = C(T)/AP(T) \tag{10}$$

We say that the policy ρ was successful in a horizon T with a level of corruption $C(T)$ if $AP(T) > \theta$, where θ is a constant greater than zero, arbitrarily defined. This criterion is adequate since in [Jun et al. 2018] authors analyze that if the Oracle attack succeeds, then $\mathbb{E}[N(k_t, T, C(T))] = T - o(T)$, which is coherent with our definition. A cost-efficient attack policy is if it can succeed with a low $CP(T)$.

In the sequence, we present the three types of attacks on stochastic MAB used in this article to illustrate the problem.

Constant Attack

The constant attack policy idea is that all the co-opted witnesses add a fixed corruption value in all reports provided to the learner. Equation 11 illustrates this idea. If the arm i is the target arm k, the co-opted witness will add a value c to reward values. For all other arms, rewards are subtracted by c. This policy is agnostic concerning the MAB policy, requires only one message to inform the appropriate c to the witnesses, and presents a fixed cost budget. However, the c value must be the same during the game.

$$\alpha(k,t) = \begin{cases} c & \text{if } k = k_t \\ -c & \text{otherwise} \end{cases} \tag{11}$$

Adaptive Attack

The adaptive attack goal is to try to maintain the pull count of the target arm in a range. We can see this attack as an improvement of the Constant attack defined in Section 4 where the corruption is a function of the time and the target arm pull counting (Equation 12). If the target arm pulls counting in the previous step ($n_{t-1} = N(k_t, t-1, C(t-1))$) is below a threshold, the attacker raises the corruption level by a small constant amount of

c. Then, the corruption level will rise until it forces the pull count above the lower bound. However, if the pull count of the target arm surpasses a higher threshold, the corruption level decreases by the same small amount. Only note that $\alpha_{i=k}(t) = -1 \cdot \alpha_{i \neq k}(t)$, as in Equation 11. The higher bound helps prevent the total cost of corruption from growing without a limit. The parameters are c_I, the inferior corruption level, c_S, the superior corruption level, c, the increase/decrease factor, and (R^S, R^I) are the superior and the inferior ratio bounds, respectively. R^S and R^I are in the $[0, 1]$ interval and $R^S > R^I$.

$$c(t, n_{t-1}, c_{t-1}) = \begin{cases} c_I & \text{if } t \leq 1 \\ min(c_S, c_{t-1} + c) & \text{if } t > 1 \text{ and } n_{t-1} < t \cdot R^I \\ max(c_I, c_{t-1} - c) & \text{if } t > 1 \text{ and } n_{t-1} > t \cdot R^S \\ c_{t-1} & \text{otherwise} \end{cases} \tag{12}$$

This attack has the advantage of being agnostic to the MAB policy, and the level of corruption will adapt to a desired target arm pull. The cost budget has the upper bound $CP(T) \leq T \cdot c_S$. However, the Attacker agent cannot know the cost budget in advance, although it might be possible by analyzing the stochastic MAB policy, which hurts agnosticism.

Jun's Adversarial Relaxed Attack

The attacks to ϵ-Greedy and UCB1 defined in [Jun et al. 2018] are adversarial. The general idea is to carefully craft the corruption value to subtract from the reward of other arms but the target arm to minimize the total corruption. We can derive a weaker attack from [Jun et al. 2018] by relaxing the assumption that the attacker can know in advance which reward an arm will give.

Let us substitute the reward in $t + 1$, $r_k(t + 1)$ by $\mathbb{E}[r_k(t + 1)] = \mu_k(t)$, since the sample mean is an unbiased estimator for $\mathbb{E}[r_k(t + 1)]$ in the stochastic MAB problem. Making this relaxation and rearranging the equations from [Jun et al. 2018], Jun's relaxed attack to ϵ-Greedy is presented in Equation 15. $N_i(t)$ is the number of times that the Learner has pulled the arm i until the round t, $\hat{\mu}_i$ is the sample mean of the arm i, being that k is the index of the target arm and σ and Δ_0 are parameters.

$$\beta(N) = \sqrt{\frac{2\sigma^2}{N} log \frac{\pi^2 K N^2}{3\delta}} \tag{13}$$

$$r_k(t) = \hat{\mu}_K(t-1) - 2\beta(N_k(t-1)) \tag{14}$$

$$\alpha_i(t) = (N_i(t-1) + 1) \cdot (\hat{\mu}_i(t-1) - r_k(t-1)) \tag{15}$$

By its turn, Jun's relaxed attack to UCB1 is presented in Equation 18.

$$A_i(t) = \sum_{s=1}^{t} \alpha_i(s) \tag{16}$$

$$r'_k(t) = r_k(t) - \Delta_0 \tag{17}$$

$$\alpha'_i(t) = (N_i(t-1) + 1) \cdot (\hat{\mu}_i(t-1) - r'_k(t-1)) - A_i(t-1) \tag{18}$$

Our relaxed version of Jun's attacks presented in [Jun et al. 2018] produces a higher corruption level when compared to the original ones since it attacks every arm that is not the target one in every round. The attacks must be indiscriminate since the attacker cannot know each one the Learner will choose. Because of this, we can no longer maintain the guarantee that cost will be limited by $O(log(T))$, as claimed in [Jun et al. 2018]. The cost tends to grow faster than $O(Klog(T))$, as in the experiments described in Section 5.

5. Experiments

This section presents experiments with the weak attacks proposed in Section 4 on manipulating the Learner agent employing UCB1 and ϵ-Greedy strategies, using the metrics defined in that section. The objective is to provide insights into the strengths and limitations of each approach, facilitating their application in the development of defenses against this kind of manipulation in MAS.

We implemented a multi-agent framework using the Java Agent Development Framework (JADE) middle-ware [Bellifemine et al. 2007]. The framework code is available at `github.com/charlesANC/BanditsExperiment`. The agent population is the Learner l, the Attacker a, nine honest witnesses, and five co-opted ones. There are more agents than arms to guarantee overlapping evaluations. For the experimental setting, $K = \{A1, B1, B2, C1, C2\}$. We set up the arm's reward functions as stochastic, stationary, sampled from a Gaussian distribution accordingly with the following profiles: A1 reward distribution is $\mathcal{N}(0.9, 0.1)$; B1 and B2's are $\mathcal{N}(0.85, 0.30)$; and C1 and C2's are $\mathcal{N}(0.75, 0.50)$. In all runs, the target arm is C2. We run games of 2,000 rounds varying learner's and attacker's policies. The Attacker can use Constant, Adaptive, and Jun's relaxed attacks described in Section 4. The Learner can use ϵ-Greedy and UCB1. For calculating the achieved pulls ($AP(T)$), we also ran games where the Learner employed the stochastic MAB with no Attacker agent. We repeated each combination 30 times. Table 1 shows the general parameters of our experiments.

Table 1. General experimental parameters

Parameter	**Value**
Number of rounds	2,000
Number of repetitions	30
ϵ value	0.80
Evaluation values range	$[0, 1]$
Honest witnesses	9
co-opted witnesses	5
c (Constant)	1.00
c_I	0
c_S	1.00
c (Adaptive)	0.20
Adaptive 1 (R^I, R^S)	(0.40, 0.60)
Adaptive 2 (R^I, R^S)	(0,80, 0.90)
δ	0.025
σ	0.001
Δ_0	0.10

For the Constant attack, we set the parameter c as 1 since the rewards' range is $[0, 1]$. For the Adaptive attack, the initial corruption value C_I was 0, and we experimented with two ranges of $[R^I, R^S]$ which were $[0.4, 0.6]$ and $[0.8, 0.9]$, which we differentiate as Adaptive 1 and 2. For Jun's relaxed attack, we set the value of parameter δ as 0.025, and Δ_0 as 0.10, such as in experiments described in [Jun et al. 2018], and σ as 0.001. It is worth noting that authors in [Jun et al. 2018] ran experiments with σ values varying from 0.05 to 0.5, depending on the scenario. However, we observed that even 0.05 was too high for our set-up, resulting in higher costs and lower efficiency. Co-opted witnesses cropped informed corrupted rewards into the $[0, 1]$ range to avoid providing out-of-scale values.

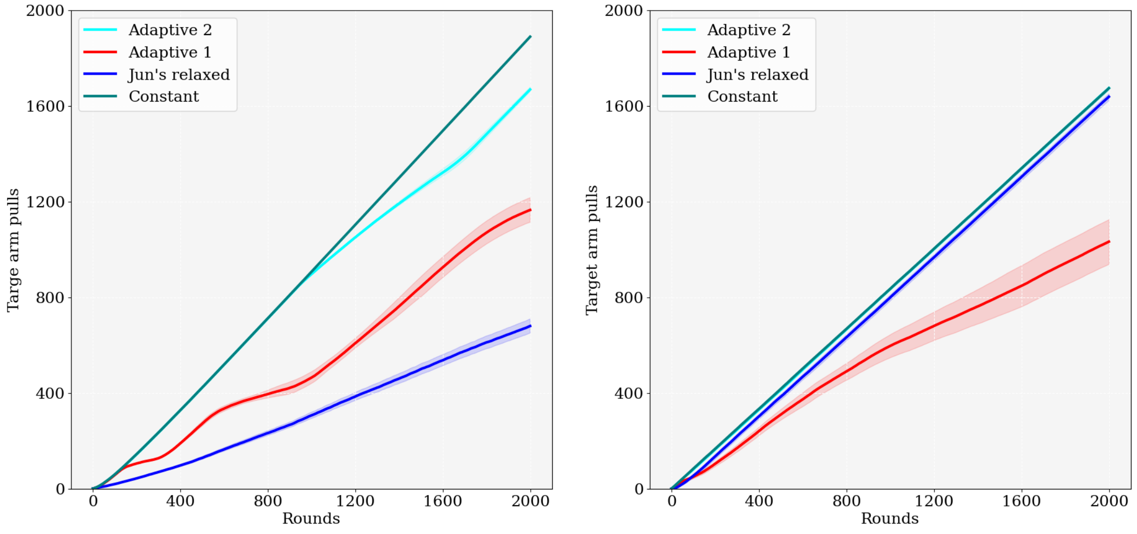
Table 2 resumes the experimental outcome. The columns show the MAB algorithm, the employed attack, regret, target arm pulls, total cost, and cost per additional target pull, respectively, as defined in Section 4. When there is no employed attack, $C(T) = 0$, thus in lines 1 and 6 $N(k_t, T, C(T)) = N(k_t, T, 0)$. However, we need $N(k_t, T, C(T))$ in lines 1 and 6 for calculating $AP(T)$ and $CP(T)$.

Table 2. Resumed measures over MAB algorithms and attacks. The values represent the mean with the standard variation in parentheses.

MAB	**Attack**	$\mathbf{R_L(T)}$	$\mathbf{N(k_t, T, C(T))}$	$\mathbf{AP(T)}$	$\mathbf{C(T)}$	$\mathbf{CP(T)}$
UCB1	-	70.93 (8.77)	131.50 (9.70)	-	-	-
UCB1	Constant	297.20 (21.42)	1,889.07 (1.69)	1,757.57 (10.03)	51,549.73 (68.40)	29.33 (0.18)
UCB1	Adaptive 1	216.34 (24.91)	1,164.87 (134.40)	1,033.37 (135.04)	19,769.67 (3,620.93)	19.11 (2.51)
UCB1	Adaptive 2	275.42 (24.39)	1,668.43 (24.99)	1,536.93 (27.41)	30,803.83 (1,565.36)	20.04 (0.89)
UCB1	Jun's relaxed	181.52 (19.12)	679.83 (77.59)	548.33 (78.79)	13,601.23 (1,271.82)	25.04 (2.18)
ϵ-Greedy	-	31.45 (8.96)	79.67 (8.83)	-	-	-
ϵ-Greedy	Constant	274.45 (23.26)	1,673.80 (13.08)	1,594.13 (15.41)	51,581.90 (68.72)	33.45 (0.30)
ϵ-Greedy	Adaptive 1	180.72 (38.44)	1,032.57 (246.09)	952.90 (244.51)	17,421.66 (8,252.80)	20.12 (9.58)
ϵ-Greedy	Adaptive 2	269.08 (21.88)	1,594.90 (18.22)	1,674.57 (18.74)	50,468.31 (3,910.64)	31.65 (2.48)
ϵ-Greedy	Jun's relaxed	268.87 (20.86)	1,638.10 (37.38)	1,594.90 (38.61)	185,662.13 (26,423.06)	123.55 (19.79)

Note that all attacks successfully increased the target arm pull count. However, we can also observe significant differences in cost values. Due to the used parameters, the cost of constant attack dominated the two tested adaptive attack ranges, which resulted in a lower cost per additional pull.

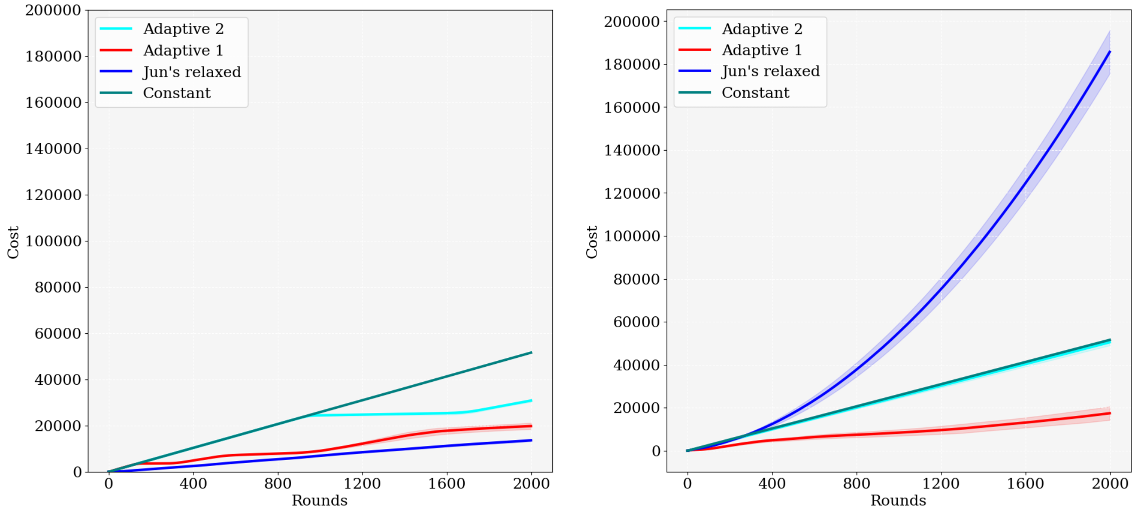
With the same parameters, Jun's relaxed attacks had very different performances depending on the MAB algorithms, despite both versions presenting the highest cost per additional pull compared to the other attack strategies. Note in line 5 of Table 2, against the UCB1 algorithm, the achieved pulls $AP(T)$ and the total cost $C(T)$ were significantly lower than other strategies' values. However, $CP(T)$ was higher. Against ϵ-Greedy, Line 10, the achieved pulls of Jun's relaxed attack were comparable to those obtained from Constant and Adaptive attacks. However, the cost measures were much higher than any other tested strategy. Also, note in Figure 2 that the target arm pulls over rounds in ϵ-Greedy experiments, that Jun's relaxed attack did not surpass the Constant attack at any moment. Those results allow us to conclude that Jun's relaxed attacks needed more work on tunning parameters, which is a disadvantage compared to Constant and Adaptive strategies. However, when testing a possible defense against data corruption, pursuing an attack that presents an exponentially growing cost curve is desirable.



(a) UCB1 - Target arm pulls.

(b) ϵ-Greedy - Target arm pulls.

Figure 2. Target arm pulls over MAB algorithms and attacks.



(a) UCB1 - Total Cost.

(b) ϵ-Greedy - Total Cost.

Figure 3. Cost of corruption over MAB algorithms and attacks.

In summary, the experimental results offer valuable insights into the behavior and cost-effectiveness of the weak attacks presented in Section 4. One can observe that while all attacks were successful in manipulating the learner choices, performance measures varied significantly. Specifically, the Adaptive attack has the advantages of being agnostic regarding the MAB algorithm, easily configurable, and cost-efficient. Jun's relaxed attacks present advantages in scenarios for testing algorithmic defenses.

6. Conclusion

This article presents the problem of fake feedback attacks on stochastic MAB algorithms from a multi-agent perspective. Our work highlighted the vulnerability of stochastic MAB with two commonly used algorithms – ϵ-Greedy and UCB1 – illustrating the problem with three types of attacks – Constant, Adaptive, and Jun's Adversarial Relaxed. By considering the roles, beliefs, interactions, goals, and motivations behind these attacks, we contributed to the problem understanding within the MAS perspective, which facilitates the development of defenses against this kind of manipulative attack. Our outcome pointed to the cost-efficiency of Constant and Adaptive attacks, besides the advantages of Jun's relaxed attacks on testing potential defenses.

Our vision is that other works that studied data poisoning attacks on MAB without a strong MAS perspective, such as [Lykouris et al. 2018], [Liu and Shroff 2019], [Niss and Tewari 2020], and [Rangi et al. 2022], could benefit from comparison using the performance measures presented in Section 4 (Equations 7-10). Future work will compare them using the performance metrics proposed in Section 4. Another possible direction to further work should be the effective defense development against fake feedback attacks in MAS requirements design, which is useful for diverse real-world problems such as e-commerce marketing, the stock market, and drug administration.

References

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2/3):235–256.

Bastani, H. and Bayati, M. (2020). Online decision making with high-dimensional covariates. *Operations Research*, 68(1):276–294.

Bellifemine, F., Caire, G., and Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE.* John Wiley & Sons, Ltd, Chichester, West Sussex, England.

Bouneffouf, D., Rish, I., and Aggarwal, C. (2020). Survey on applications of multi-armed and contextual bandits. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, page 1–8. IEEE Press.

Guan, Z., Ji, K., Bucci Jr, D. J., Hu, T. Y., Palombo, J., Liston, M., and Liang, Y. (2020). Robust stochastic bandit algorithms under probabilistic unbounded adversarial attack. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4036–4043, New York, USA. AAAI Press.

Jun, K.-S., Li, L., Ma, Y., and Zhu, J. (2018). Adversarial attacks on stochastic bandits. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31, Montréal, Canada. Curran Associates, Inc.

Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670.

Liu, F. and Shroff, N. (2019). Data poisoning attacks on stochastic bandits. In *International Conference on Machine Learning*, pages 4042–4050. PMLR.

Lykouris, T., Mirrokni, V., and Leme, R. P. (2018). Stochastic bandits robust to adversarial corruptions. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, page 114–122, Los Angeles, USA. ACM.

Niss, L. and Tewari, A. (2020). What you see may not be what you get: Ucb bandit algorithms robust to ε-contamination. In Peters, J. and Sontag, D., editors, *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124 of *Proceedings of Machine Learning Research*, pages 450–459, Virtual. PMLR.

Rangi, A., Tran-Thanh, L., Xu, H., and Franceschetti, M. (2022). Saving stochastic bandits from poisoning attacks via limited data verification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36-7, pages 8054–8061, New York, USA. AAAI Press.

Schwartz, E. M., Bradlow, E. T., and Fader, P. S. (2017). Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science*, 36(4):500–522.

Shen, W., Wang, J., Jiang, Y.-G., and Zha, H. (2015). Portfolio choices with orthogonal bandit learning. In *Twenty-fourth international joint conference on artificial intelligence*.

Slivkins, A. (2019). Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286.

Vallée, T., Bonnet, G., and Bourdon, F. (2014). Multi-armed bandit policies for reputation systems. In Y., D., F., Z., J.M., C., and J., B., editors, *Advances in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection*, volume 8473 of *Lecture Notes in Computer Science*. Springer, Cham, Salamanca, Spain.

Vermorel, J. and Mohri, M. (2005). Multi-armed bandit algorithms and empirical evaluation. In *Proceedings of the 16th European Conference on Machine Learning*, ECML'05, page 437–448, Berlin, Heidelberg. Springer-Verlag.

Wooldridge, M. (2009). *An Introduction to Multiagent Systems*. Wiley, Chichester, UK, 2 edition.

Zhang, Y., Bian, J., and Zhu, W. (2013). Trust fraud: A crucial challenge for china's e-commerce market. *Electronic Commerce Research and Applications*, 12(5):299–308.