Reward Attack on Stochastic Bandits with Non-stationary Rewards

Chenye Yang, Guanlin Liu and Lifeng Lai

***Abstract*—In this paper, we investigate rewards attacks on stochastic multi-armed bandit algorithms with non-stationary environment. The attacker's goal is to force the victim algorithm to choose a suboptimal arm most of the time while incurring a small attack cost. Three main attack scenarios are considered: easy attack scenario, general attack scenario, and general attack scenario with limited information of victim algorithm. These scenarios have different assumptions about the environment and accessible information. We propose three attack strategies, one for each considered scenario, and prove that they are successful in terms of expected target arm selection and attack cost. The simulation results validate our theoretical analysis.**

***Index Terms*—bandit, non-stationary reward, attack cost**

I. Introduction

Multi-armed bandit (MAB) problems is a class of sequential decision-making problems that have wide range of applications. This class of problems model the scenario where an agent algorithm must choose between multiple arms to maximize its cumulative reward. They have been applied to various fields, including online advertisement (optimizing ad selection), healthcare (personalized treatment strategies), and recommender systems (improving personalized recommendations). Existing works [1]–[3] have identified potential security issues of existing MAB algorithms. In particular, these work show that an attacker can force existing MAB algorithms to take unwanted actions, e.g., choose a suboptimal target arm, and may lead to severe real-world consequences (unfair business competition, health threats etc.).

Prior works investigate two attack methods: manipulate the reward signal [1] [2] or manipulate the action signal [4] [3]. Most of these existing work focused on the traditional stationary random rewards setting, in which the distribution of reward of each arm does not change over time. Some [2] studied the adversarial setting, in which the reward given by the environment can be arbitrarily chosen. It is important to note that in many real-world applications, the reward distribution may change over time, but with a specific restriction on the extent of changes. For example, the best product recommendation may vary when the user's interest slightly changes. In this case, it is more appropriate to model the problem as stochastic multi-armed bandit with non-stationary rewards [5]. In this paper, we study the reward attack on non-stationary MAB algorithms in the non-stationary reward setting with restriction on the extent of changes.

C. Yang, G. Liu and L. Lai are with the Department of Electrical and Computer Engineering, University of California, Davis, CA. This work was supported by the National Science Foundation under Grants ECCS-2000415 and CCF-2232907. Email:{cyyyang,glnliu,lflai}@ucdavis.edu.

The non-stationary reward structure will introduce additional challenges on both algorithm side and attacker side. In addition to the exploitation and exploration trade-off, the algorithm also needs to handle trade-off between 'remembering' and 'forgetting' since the estimation of expected rewards is based on past rewards observations, and will have larger overall regret since the best arm is always changing. At the same time, this also creates additional challenges for the attack design as well, as it costs more for the attacker to perform a successful attack, since every time when the victim algorithm 'forgets' history it tends to 'explore' all possible arms instead of 'exploit' the target arm. To our knowledge, this is the first work to successfully attack those specifically designed non-stationary MAB algorithm with a variation budget, which models the temporal uncertainty and changes in the non-stationary reward environment.

In this paper, three attack scenarios targeting non-stationary MAB algorithm are considered: easy attack scenario, general attack scenario, and general attack scenario with limited information of victim algorithm. The first scenario has the most strict restriction on the environment side, which asks for non-zero reward for the target arm. The second scenario relaxes the assumption on the environment side, but requires the attacker to know more detailed behavior of the victim algorithm. The third scenario further relaxes the additional requirement on the attacker side, and limits the attacker's knowledge to the victim algorithm. For each scenario, we propose the corresponding attack strategy and prove them to be successful.

The remainder of the paper is organized as follows. In Section II, we introduce the problem formulation for victim algorithm, environment, and attacker. In Section III, we model three attack scenarios and propose the attack strategies for each scenario. In Section IV, we provide theoretical analysis on the performance of the proposed attack strategies. In Section V, we validate our theoretical analysis with simulation results. Finally, we conclude the paper in Section VI.

II. Problem Formulation

A. Multi-armed bandit problem

Let $\mathcal{K} = \{1, 2, \ldots, K\}$ be the set of arms to be pulled (decisions to be made), $\mathcal{T} = \{1, 2, \ldots, T\}$ be the sequence of decision steps for the decision maker (agent). The agent pulls an arm $a_t \in \mathcal{K}$ at step $t \in \mathcal{T}$ and receives a reward $X_t(a_t)$ which is generated by the environment. $X_t(a_t) \in [0, 1]$ is a random variable with expectation $\mathbb{E}[X_t(a_t)]$. The goal of the agent is to maximize the total expected reward over a long time, while balancing exploration and exploitation.

B. Non-stationary environment

In many practical cases, the reward distributions of the arms in an MAB problem may change over time. In the existing works, there are two popular approaches to model the changing environment: adversarial environment and non-stationary environment. The adversarial environment allows the reward distribution to be arbitrarily chosen by the environment. The downside of this model is that it does not restrict the extent of changes and thus not capture the temporal uncertainty of the reward distribution. Another category is the non-stationary environment which allows the reward distribution to change over time, but also models the temporal uncertainty in the following two ways: allow a finite number of changes in the expected reward [6], or allow a bounded total variation of expected reward over the relevant time [5]. In this paper, we will perform attack within the non-stationary environment with a bounded total variation of the expected rewards.

Denote $\mu_t^k = \mathbb{E}[X_t(a_t)] : a_t = k$ and $\mu_t^* = \max_{k\in\mathcal{K}}\{\mu_t^k\}$, where $\mathbb{E}$ is taken with respect to reward $X_t(a_t)$ at step t. In this paper, we focus on the non-stationary environment with a bounded total variation of the expected rewards $\mathbb{E}[X_t(a_t)]$:

$$\sum_{t=1}^{T-1} \sup_{k\in\mathcal{K}} \left|\mu_t^k - \mu_{t+1}^k\right| \leq V_T,$$

where V_T is the variation budget for the entire horizon T of the problem. We define the temporal uncertainty set as the set of reward sequences that are subject to the variation budget V_T over the set of step epochs $\{1,\ldots,T\}$:

$$\mathcal{V} = \left\{ \mu \in [0,1]^{K\times T} : \sum_{t=1}^{T-1} \sup_{k} \left|\mu_t^k - \mu_{t+1}^k\right| \leq V_T \right\}.$$

Note that $V_T = 0$ corresponds to the stationary environment while $V_T = O(T)$ corresponds to the adversarial environment [7].

C. MAB algorithm performance

The performance of a multi-armed bandit algorithm is measured by its regret. For these adversarial bandit problems, the regret R_T is defined against a static oracle, which is the best arm in hindsight over the whole horizon [7] [2]:

$$R_T = \max_a \sum_{t=1}^{T} X_t(a) - \mathbb{E}^\pi \left[\sum_{t=1}^{T} X_t(a_t) \right],$$

where the $\mathbb{E}^\pi$ is taken with respect to the noisy rewards and policy's actions.

One may use adaptive algorithms such as Exponential-weight algorithm for Exploration and Exploitation (*Exp3*) to handle adversarial environments. The static oracle regret of *Exp3* is $O(\sqrt{KT\log K})$ [8].

In the non-stationary setting, the regret of the algorithm over the entire horizon is defined as the worst case difference between the expected performance of pulling at each epoch t the arm which has the highest expected reward at epoch t and the expected performance under policy π, which is also known as the regret measured against the dynamic oracle [5] [7]:

$$R^\pi(\mathcal{V},T) = \sup_{\mu\in\mathcal{V}} \left\{ \sum_{t=1}^{T} \mu_t^* - \mathbb{E}^\pi \left[\sum_{t=1}^{T} \mu_t^\pi \right] \right\}.$$

D. Non-stationary victim algorithm

If there is no attacker, a widely used strategy for stochastic non-stationary MAB problem is *Rexp3*, Algorithm 1, proposed in [5], which is able to handle the temporal uncertainty by variation budget. Furthermore, it is proved that *Rexp3* is nearly minimax optimal with dynamic oracle regret of order ${V_T}^{1/3}T^{2/3}$, which is the best achievable performance under worst-case regret.

Algorithm 1 *Rexp3*

Parameters: A learning rate η, and a batch size Δ_T.
for Batch $j = 1, 2, \ldots, m = \lceil \frac{T}{\Delta_T} \rceil$ **do**
  Initialization: $w_{t,a} = 1, \forall a \in \mathcal{K}$.
  for $t = (j-1)\Delta_T + 1 \leq t \leq \min\{j\Delta_T, T\}$ **do**
    Define $\pi_{t,a} = (1-\eta)\frac{w_{t,a}}{\sum_a w_{t,a}} + \frac{\eta}{K}$
    Draw $a_t \sim \{\pi_{t,a}\}$, and observe reward $X_t(a_t)$
    for $a = 1, 2, \ldots, K$ **do**

$$w_{t+1,a} = \begin{cases} w_{t,a} & , a \neq a_t \\ w_{t,a}\exp\left(\frac{\eta}{K}\frac{X_t(a_t)}{\pi_{t,a}}\right) & , a = a_t \end{cases}$$

    end for
  end for
end for

In *Rexp3*, to handle the non-stationary environment, the total horizon $\mathcal{T}$ is split into many batches $(\mathcal{T}_1,\ldots,\mathcal{T}_m)$ with fixed size Δ_T each (except, possibly the last batch):

$$\mathcal{T}_j = \{t : (j-1)\Delta_T + 1 \leq t \leq \min\{j\Delta_T, T\}\},$$

$\forall j = 1,\ldots,m$, where $m = \lceil \frac{T}{\Delta_T} \rceil$ is the number of batches. The *Exp3* algorithm will restart itself at the beginning of each batch, to forget all its memory and handle the changing environment, as illustrated in Figure 1.

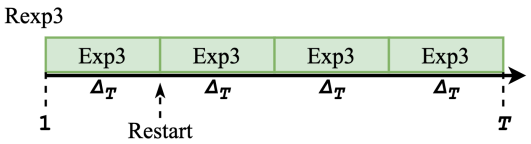


Fig. 1: Timeline of Rexp3

With the assumption that variation budget V_T is known to algorithm, *Rexp3* chooses the batch size $\Delta_T = \left\lceil (K\log K)^{1/3}(T/V_T)^{2/3} \right\rceil$, and achieves the dynamic oracle regret mentioned above.

Note that in the j-th batch $\mathcal{T}_j$, the dynamic regret is:

$$\mathbb{E}^\pi \left[\sum_{t\in\mathcal{T}_j} \left(\max_{a\in\mathcal{K}} \{\mathbb{E}[X_t(a)]\} - \mathbb{E}[X_t(a_t^\pi)] \right) \right].$$

E. Attack the reward

In this paper, as shown in Figure 2, we consider a setup where there is an attacker that can intercept and manipulate the reward signal generated by the environment. We denote the manipulated reward as $\tilde{X}_t(a_t)$. The attacker's goal is to force the agent to choose a suboptimal target arm $a^\dagger$ as often as possible while inducing an attack cost as low as possible. Specifically, we attack the *Rexp3* algorithm in the non-stationary reward setting with a variation budget V_T.

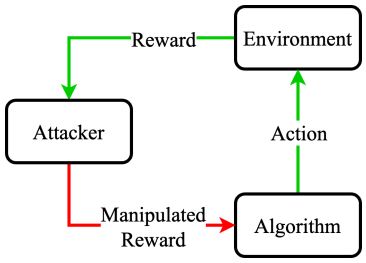


Fig. 2: Reward Attack

There are two metrics to measure the performance of attacker: expected attack cost $\mathbb{E}^\pi[C_T]$, and expected number of target arm selection $\mathbb{E}^\pi[N_\mathcal{T}(a^\dagger)]$:

$$\mathbb{E}^\pi[C_T] = \mathbb{E}^\pi\left[\sum_{t=1}^{T}\left|\tilde{X}_t(a_t) - X_t(a_t)\right|\right],$$
$$\mathbb{E}^\pi[N_\mathcal{T}(a^\dagger)] = \mathbb{E}^\pi\left[\sum_{t=1}^{T}\mathbf{1}\left[a_t = a^\dagger\right]\right].$$

The goal of attacker is to maximize the $\mathbb{E}^\pi[N_\mathcal{T}(a^\dagger)]$ while incurring a small attack cost $\mathbb{E}^\pi[C_T]$.

III. Proposed Method

In this paper we consider three increasingly complex scenarios and propose three attack strategies, one for each considered scenario.

A. Easy attack scenario

In this scenario, on the environment side, we assume that there exists a constant $\rho \in (0, 1]$ such that $X_t(a^\dagger) \in [\rho, 1]$, $\forall t \in \mathcal{T}$. This scenario has the most strict assumption on the environment side, in which the target arm will always have a non-zero reward.

In this scenario, the attacker follows the scheme in Algorithm 2. If the agent algorithm chooses a non-target arm, the attacker will reduce the reward to the minimum, i.e., 0. However, if the agent chooses the target arm, the attacker will not manipulate the reward signal $X_t(a^\dagger)$. In other words, the manipulated reward $\tilde{X}_t(a_t)$ will be:

$$\tilde{X}_t(a_t) = \begin{cases} X_t(a^\dagger) & a_t = a^\dagger \\ 0 & a_t \neq a^\dagger \end{cases}.$$

Algorithm 2 Easy Attack Scenario

```
Initialize total horizon T, step t = 1
while t ≤ T do
    Victim algorithm, e.g., Rexp3, chooses arm a_t
    Environment gives the reward X_t(a_t)
    Attacker manipulates the reward:
    if a_t = a† then
        X̃_t(a_t) = X_t(a_t)
    else
        X̃_t(a_t) = 0
    end if
    Victim updates itself based on X̃_t(a_t)
    t = t + 1
end while
```

B. General attack scenario

In this scenario, there's no constraint on the environment side, which means that sometimes $X_t(a^\dagger) = 0$. This scenario relaxes the assumption on the environment side. However, we assume the attacker has information of when victim algorithm restarts, i.e., the batch size Δ_T. Note that this assumption will be removed in the next scenario.

In this scenario, the attacker follows the scheme in Algorithm 3. It keeps a diminish function $\tilde{t}^{\alpha+\epsilon-1}$ as the lower-bound of manipulated reward for target arm, where $\tilde{t}$ is the relative step in each batch and $\epsilon < 1 - \alpha$. Within each batch, if the victim algorithm chooses a non-target arm, the attacker will reduce the reward to 0. However, if the victim algorithm chooses the target arm, the attacker will manipulate the reward to the maximal value between the original reward and $\tilde{t}^{\alpha+\epsilon-1}$. The relative step $\tilde{t}$ will increase along with the absolute step $t \in \mathcal{T}$, but will be reset to 1 at the beginning of a new batch. The attacker is able to restart the diminish function simultaneously with the algorithm by the assumption of knowing Δ_T. In other words, the manipulated reward $\tilde{X}_t(a_t)$ will be:

$$\tilde{X}_t(a_t) = \begin{cases} \max\left\{\tilde{t}^{\alpha+\epsilon-1}, X_t(a_t)\right\} & a_t = a^\dagger \\ 0 & a_t \neq a^\dagger \end{cases}.$$

C. General attack scenario without knowing Δ_T

We now consider the scenario where there's no constraint on the environment side. Furthermore, the attacker does NOT know when the algorithm restarts, i.e., the attacker does not know the batch size Δ_T. This scenario further relaxes the additional requirement on the attacker side, and limits the attacker's knowledge to the victim algorithm.

In this scenario, the attacker follows the scheme in Algorithm 4. It also keeps a diminish function $\tilde{t}^{\alpha+\epsilon-1}$. If the victim algorithm chooses a non-target arm, the attacker will reduce the reward to 0, and reset the relative step $\tilde{t}$ to 1. However, if the victim algorithm chooses the target arm, the attacker will manipulate the reward to the maximal value between the

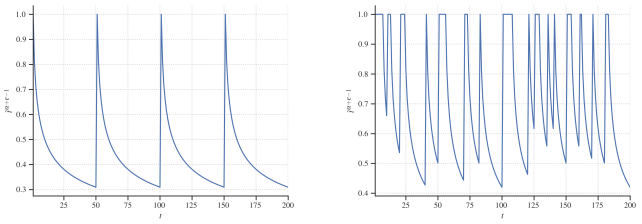
Algorithm 3 General Attack Scenario

Initialize total horizon T
Let absolute step $t = 1$, let relative step $\tilde{t} = 1$
while $t \leq T$ **do**
  Victim algorithm, e.g., *Rexp3*, chooses arm a_t
  Environment gives the reward $X_t(a_t)$
  Attacker manipulates the reward:
  if $a_t = a^\dagger$ **then**
    $\tilde{X}_t(a_t) = \max\{\tilde{t}^{\alpha+\epsilon-1}, X_t(a_t)\}$
  else
    $\tilde{X}_t(a_t) = 0$
  end if
  if $\tilde{t} \leq \Delta_T$ **then** ▷ $\tilde{t} \leq T - (m-1)\Delta_T$ for last batch
    $\tilde{t} = \tilde{t} + 1$ ▷ continue the diminish function
  else
    reset $\tilde{t} = 1$ ▷ restart the diminish function
  end if
  Victim updates itself based on $\tilde{X}_t(a_t)$
  $t = t + 1$
end while

original reward and $\tilde{t}^{\alpha+\epsilon-1}$, and then increase the relative step $\tilde{t}$ by 1. Similarly, the manipulated reward $\tilde{X}_t(a_t)$ will be:

$$\tilde{X}_t(a_t) = \begin{cases} \max\left\{\tilde{t}^{\alpha+\epsilon-1}, X_t(a_t)\right\} & a_t = a^\dagger \\ 0 & a_t \neq a^\dagger \end{cases}.$$

The main difference between the attacker's behavior in this scenario and the previous scenario is when the attacker resets its diminishing function. In the previous scenario, the attacker is able to reset the diminishing function at the beginning of each batch simultaneously with the victim algorithm by the assumption of knowing Δ_T. However, in this scenario, the attacker does not have that information, and thus will reset the diminishing function anytime when a non-target arm is pulled. Figure 3 illustrates how the diminish functions behave in the two scenarios.



(a) General attack (b) General attack without Δ_T

Fig. 3: Comparisons of the diminishing functions in two scenarios

The above strategy helps to give more reward to the target arm when it is selected after any non-target arm selection. There's always a possibility for the target arm been selected because of the "exploration" of bandit algorithm. This attack strategy Algorithm 4 ensures that the victim algorithm receives a larger target arm reward and thus tends to choose the target arm with higher probability, compared with the first two strategies Algorithm 2 and 3.

Algorithm 4 General Attack Scenario without Δ_T

Initialize total horizon T
Let absolute step $t = 1$, let relative step $\tilde{t} = 1$
while $t \leq T$ **do**
  Victim algorithm, e.g., *Rexp3*, chooses arm a_t
  Environment gives the reward $X_t(a_t)$
  Attacker manipulates the reward $\tilde{X}_t(a_t)$:
  if the chosen arm is the target arm: $a_t = a^\dagger$ **then**
    $\tilde{X}_t(a_t) = \max\{\tilde{t}^{\alpha+\epsilon-1}, X_t(a_t)\}$
    $\tilde{t} = \tilde{t} + 1$ ▷ continue the diminish function
  else if $a_t \neq a^\dagger$ **then**
    $\tilde{X}_t(a_t) = 0$
    reset $\tilde{t} = 1$ ▷ restart the diminish function
  end if
  Victim updates itself based on $\tilde{X}_t(a_t)$
  $t = t + 1$
end while

IV. Theoretical analysis

A. Easy attack scenario

In this section, we analyze the performance of the attacker in the easy attack scenario.

Theorem IV.1. *Assume that there exists some constant $\rho \in (0, 1]$ such that $X_t(a^\dagger) \in [\rho, 1]$ $\forall t \in [T]$, the victim algorithm has static oracle regret $R_T = O\left({\Delta_T}^\alpha\right)$ for some $\alpha \in [\frac{1}{2}, 1)$ in each batch and follows the batch strategy to handle non-stationary reward, and the attack is performed as Algorithm 2. Then the expected number of target arm selection and the expected attack cost satisfies:*

$$\begin{aligned} \mathbb{E}^\pi\left[N_T\left(a^\dagger\right)\right] &\geq T - \frac{2M}{\rho} \cdot T{\Delta_T}^{\alpha-1}, \\ \mathbb{E}^\pi\left[C_T\right] &\leq \frac{2M}{\rho} \cdot T{\Delta_T}^{\alpha-1}, \end{aligned} \tag{1}$$

where M is a constant.

Theorem IV.1 reveals that in the adversarial case when $\Delta_T = T$, meaning no batch behavior and no consideration of variation budget V_T, the attacker will achieve $O(T^\alpha)$ expected attack cost. For example, for *Exp3*, when $\alpha = \frac{1}{2}$, the expected attack cost is $O(\sqrt{T})$.

In the non-stationary cases, if the victim algorithm sets batch size as $\Delta_T = B\left(\frac{T}{V_T}\right)^\beta$, where $\beta \in [0, 1]$, and B is a constant independent of T, V_T and β, the expected attack cost $\mathbb{E}^\pi[C_T]$ will be:

$$\mathbb{E}^\pi\left[C_T\right] = O\left({V_T}^{\beta(1-\alpha)} T^{1-\beta(1-\alpha)}\right).$$

When the variation budget V_T is sublinear in T, i.e., $V_T = O(T^\gamma)$ where $\gamma \in (0, 1)$, the expected attack cost $\mathbb{E}^\pi[C_T]$ will be:

$$\mathbb{E}^\pi\left[C_T\right] = O\left(T^{1-(1-\gamma)\beta(1-\alpha)}\right),$$

which is sublinear in T. Meanwhile, the target arm selection $\mathbb{E}^{\pi}\left[N_{\mathcal{T}}\left(a^{\dagger}\right)\right]$ will increase almost linear in T.

B. General attack scenario

In this section, we analyze the performance of the attacker in the general attack scenario.

Theorem IV.2. *Assume that the reward $X_t(a_t) \in [0,1]$, the victim algorithm has static oracle regret $R_T = O\left({\Delta_T}^{\alpha}\right)$ for some $\alpha \in [\frac{1}{2}, 1)$ in each batch and follows the batch strategy to handle non-stationary reward, and the attack is performed as Algorithm 3. Then the expected number of target arm selection and the expected attack cost satisfies:*

$$\begin{aligned}
&\mathbb{E}^{\pi}\left[N_{\mathcal{T}}\left(a^{\dagger}\right)\right] \\
&\geq T - \left(2MT{\Delta_T}^{-\epsilon} + \frac{2}{\alpha+\epsilon}T{\Delta_T}^{-\alpha-\epsilon}\right), \\
&\mathbb{E}^{\pi}\left[C_T\right] \\
&\leq 2MT{\Delta_T}^{-\epsilon} + \frac{2}{\alpha+\epsilon}T{\Delta_T}^{-\alpha-\epsilon} + \frac{2}{\alpha+\epsilon}T{\Delta_T}^{\alpha+\epsilon-1},
\end{aligned} \tag{2}$$

where M is a constant, and $\epsilon < 1-\alpha$.

Theorem IV.2 reveals that in the adversarial case when $\Delta_T = T$, meaning no batch behavior and no consideration of variation budget V_T, the attacker will achieve $O(T^{1-\epsilon}+T^{\alpha+\epsilon})$ expected attack cost. For example, for *Exp3*, when $\alpha = \frac{1}{2}$, the expected attack cost is $O(T^{1-\epsilon}+T^{\frac{1}{2}+\epsilon})$. The ϵ can be selected as $\frac{1}{4}$ to satisfy that $\epsilon < 1-\alpha$. Then, the expected attack cost is $O(T^{\frac{3}{4}})$.

In the non-stationary cases, if the victim algorithm sets batch size as $\Delta_T = B\left(\frac{T}{V_T}\right)^{\beta}$, where $\beta \in [0,1]$, and B is a constant independent of T, V_T and β, the expected attack cost $\mathbb{E}^{\pi}\left[C_T\right]$ will be:

$$\mathbb{E}^{\pi}\left[C_T\right] = O\left({V_T}^{\epsilon\beta}T^{1-\epsilon\beta} + {V_T}^{\beta(1-\alpha-\epsilon)}T^{1-\beta(1-\alpha-\epsilon)}\right).$$

When the variation budget V_T is sublinear in T, i.e., $V_T = O(T^{\gamma})$ where $\gamma \in (0,1)$, the expected attack cost $\mathbb{E}^{\pi}[C_T]$ will be:

$$\mathbb{E}^{\pi}\left[C_T\right] = O\left(T^{1-(1-\gamma)\epsilon\beta} + T^{1-(1-\gamma)\beta(1-\alpha-\epsilon)}\right),$$

which is sublinear in T. Meanwhile, the target arm selection $\mathbb{E}^{\pi}\left[N_{\mathcal{T}}\left(a^{\dagger}\right)\right]$ will increase almost linear in T.

C. General attack scenario without knowing Δ_T

In this section, we analyze the performance of the attacker in the general attack scenario without knowing Δ_T.

Theorem IV.3. *Assume that the reward $X_t(a_t) \in [0,1]$, the victim algorithm has static oracle regret $R_T = O\left({\Delta_T}^{\alpha}\right)$ for some $\alpha \in [\frac{1}{2}, 1)$ in each batch and follows the batch strategy to handle non-stationary reward, and the attack is performed as Algorithm 4. Then the expected number of target arm selection and the expected attack cost satisfies:*

$$\begin{aligned}
&\mathbb{E}^{\pi}\left[N_{\mathcal{T}}\left(a^{\dagger}\right)\right] \\
&\geq T - 2MT{\Delta_T}^{\alpha-1}, \\
&\mathbb{E}^{\pi}\left[C_T\right] \\
&\leq 2MT{\Delta_T}^{\alpha-1} + \frac{(2M)^{1-\alpha-\epsilon}}{\alpha+\epsilon}T{\Delta_T}^{2\alpha-\alpha^2-1+\epsilon-\alpha\epsilon},
\end{aligned} \tag{3}$$

where M is a constant, and $\epsilon < 1-\alpha$.

Theorem IV.3 reveals that in the adversarial case when $\Delta_T = T$, meaning no batch behavior and no consideration of variation budget V_T, the attacker will achieve $O\left(T^{\alpha} + T^{2\alpha-\alpha^2+\epsilon-\alpha\epsilon}\right)$ expected attack cost, and the ϵ can be selected such that $2\alpha-\alpha^2+\epsilon-\alpha\epsilon < 1$, which is reasonable since $2\alpha-\alpha^2 < 1$ and $1-\alpha \in (0, \frac{1}{2}]$ for $\alpha \in [\frac{1}{2}, 1)$.

For example, for *Exp3*, when $\alpha = \frac{1}{2}$, the expected attack cost is $O\left(T^{\frac{1}{2}} + T^{\frac{3}{4}+\frac{1}{2}\epsilon}\right)$. The ϵ can be selected as $\frac{1}{10}$ to satisfy that $\epsilon < 1-\alpha$. Then, the expected attack cost $\mathbb{E}^{\pi}\left[C_T\right]$ will be $O\left(T^{\frac{4}{5}}\right)$.

In the non-stationary cases, if the victim algorithm sets batch size as $\Delta_T = B\left(\frac{T}{V_T}\right)^{\beta}$, where $\beta \in [0,1]$, and B is a constant independent of T, V_T and β, the expected attack cost $\mathbb{E}^{\pi}\left[C_T\right]$ will be:

$$\begin{aligned}
\mathbb{E}^{\pi}\left[C_T\right] = O\Big({V_T}^{\beta(1-\alpha)}T^{1-\beta(1-\alpha)} \\
+ {V_T}^{\beta(1-2\alpha+\alpha^2-\epsilon+\alpha\epsilon)}T^{1-\beta(1-2\alpha+\alpha^2-\epsilon+\alpha\epsilon)}\Big).
\end{aligned}$$

When the variation budget V_T is sublinear in T, i.e., $V_T = O(T^{\gamma})$ where $\gamma \in (0,1)$, the expected attack cost $\mathbb{E}^{\pi}[C_T]$ will be:

$$\mathbb{E}^{\pi}\left[C_T\right] = O\left(T^{1-(1-\gamma)\beta(1-\alpha)} + T^{1-(1-\gamma)\beta(1-2\alpha+\alpha^2-\epsilon+\alpha\epsilon)}\right),$$

which is sublinear in T. Meanwhile, the target arm selection $\mathbb{E}^{\pi}\left[N_{\mathcal{T}}\left(a^{\dagger}\right)\right]$ will increase almost linear in T.

D. Lower-bound of the expected attack cost

We have shown the upper-bound of the expected attack cost and lower-bound of the expected target arm selection of our attack strategies for three scenarios in Theorem IV.1, IV.2 and IV.3. We have proved that our attack strategies can successfully control the victim algorithm's behavior and induce a small cost. In this section, we show that our attack strategies are near optimal. In particular, we show that if an attacker achieves $T-o(T)$ expected target arm selection, and it is also victim-agnostic to non-stationary bandit algorithm that has the batch behavior, then the attacker must induce at least expected attack cost $\Omega\left(T{\Delta_T}^{\alpha-1}\right)$, where Δ_T is the batch size. Here, victim-agnostic means that the attacker does not know what is exactly the victim algorithm, but only knows that the non-stationary algorithm has sublinear static oracle regret in each batch and follows the batch strategy.

Since we are looking for the victim-agnostic lower-bound, it is sufficient to pick a particular victim non-stationary algorithm that guarantees $O(T^{\alpha})$ static oracle regret in each batch,

under one bandit environment. Then we need to show that any victim-agnostic attacker must induce at least some attack cost to achieve $T - o(T)$ expected target arm selection on this particular victim algorithm. Specifically, we consider the *Exp3* algorithm with batch behavior, whose static oracle regret is $O(T^{\frac{1}{2}})$ in each batch. The main result for lower-bound of the expected attack cost is provided in Theorem IV.4.

Theorem IV.4. *Assume some victim-agnostic attack algorithm achieves* $\mathbb{E}^{\pi}\left[N_T(a^{\dagger})\right] = T - o(T)$ *on all victim bandit algorithms that has static oracle regret* $O\left({\Delta_T}^{\alpha}\right)$ *in each batch and follows the batch strategy to handle non-stationary reward, where* $\alpha \in \left[\frac{1}{2}, 1\right)$. *Then there exists a bandit task such that the attacker must induce at least expected attack cost* $\mathbb{E}^{\pi}\left[C_T\right] = \Omega\left(T{\Delta_T}^{\alpha-1}\right)$ *on some victim algorithm, where* Δ_T *is the fixed batch size.*

Theorem IV.4 reveals that the best achievable performance of attacker is $\Omega\left(T{\Delta_T}^{\alpha-1}\right)$, and our attack method is near optimal in the easy attack scenario as Algorithm 2. For the general scenarios as Algorithm 3 and Algorithm 4, our methods may introduce a small additional cost depending on the choice of parameters β and ϵ.

V. Experimental Data and Results

We consider a bandit problem environment with $K = 5$ arms. The target arm $a^{\dagger} = 1$. The initial expected reward is:

$$\mathbb{E}\left[X_t(a)\right] = \begin{cases} 0.1, & a = 1 \\ 0.5, & a = 2, 3, 4 \\ 0.8, & a = 5 \end{cases}.$$

The non-stationary reward structure is simulated by random walk, which changes the expected reward at each step, and the total variation of expected reward is bounded by $V_T = (T/K)^{1/10}$. The reward signal in $[0, 1]$ given by environment at each step t is sampled from a Beta distribution. For all three attack scenarios, all rewards are scaled to $[\rho, 1]$, where $\rho = 0.1$ in our experiment. Using the same scaled rewards for all three cases makes it more reasonable to compare the results.

We attack the popular strategy for stochastic non-stationary MAB problem, *Rexp3* as described in Section II-D. Figure 4 shows the batch size $\Delta_T = \left\lceil 5(T/V_T)^{2/3} \right\rceil$ for different horizon T-s in our experiments. The diminishing function parameters are $\alpha = \frac{1}{2}$ and $\epsilon = \frac{1}{5}$. Expectation is taken over 5 independent runs.

The attack result is shown in Figure 5 and 6. Scenario 1, 2 and 3 correspond to three attack scenarios: easy attack (Algorithm 2), general attack (Algorithm 3), and general attack without knowing Δ_T (Algorithm 4).

Figure 5 shows that the number of target arm selection $\mathbb{E}^{\pi}\left[N_{\mathcal{T}}\left(a^{\dagger}\right)\right]$ increases significantly with attack. For example, as shown in Table I, the percentage of target arm selection increased from 1.2% to 69.23% in scenario 1, from 1.2% to 73.34% in scenario 2, and from 1.2% to 94.83% in scenario 3. Note that, in scenario 3, compared with scenarios 1 and 2, the victim algorithm tends to select $a^{\dagger}$ more often since the attacker gives more reward to the target arm every time after a non-target arm is chosen. However, as shown in Figure 6, Algorithm 4 also has disadvantage of having a larger attack cost. In particular, the expected attack cost for 500,000 steps are 88480.1, 77448.3 and 152052.8 for scenario 1, 2 and 3, respectively.

TABLE I: Expected target arm pulls and percentage for 500,000 steps

	With Attack		Without Attack	
Scenario	**Pulls**	**%**	**Pulls**	**%**
Scenario 1	346164.8	69.23	5994.8	1.20
Scenario 2	366715.8	73.34	5976.8	1.20
Scenario 3	474145.0	94.83	5997.4	1.20

The reason for the more target arm selection while less attack cost in scenario 2, compared with scenario 1, is that the attacker has a diminishing function in scenario 2. The diminishing function works as the lower-bound for manipulated reward of the target arm. As a result, the victim algorithm will receive more reward from the target arm at the beginning of each batch, and thus tends to choose the target arm more often. As long as the victim algorithm chooses the target arm, and the diminishing function is less than the original reward, which is usually the case after the first few steps in each batch, there will be no additional attack cost.

In general, our three attack strategies are successful: the expected attack cost, shown in Figure 6, is sublinear to T, when the victim algorithm is forced to select one suboptimal arm mostly and the number of selection increases almost linearly with T, shown in Figure 5.

Figure 7 shows the expected attack cost in the general attack scenario for different choice of parameter β for batch size $\Delta_T = \left\lceil 5(T/V_T)^{\beta} \right\rceil$, which is shown in Figure 8. The results verify Theorem IV.2 that if the order of Δ_T is larger, the attack cost will be smaller, since the power of Δ_T-s in Equation (2) is negative.

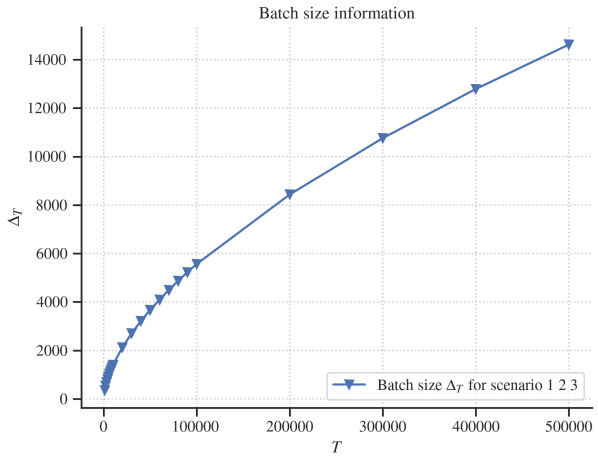


Fig. 4: Batch size Δ_T for different horizon T

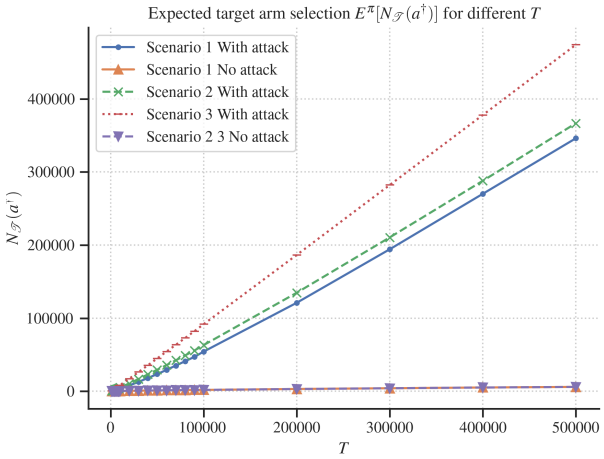


Fig. 5: Target arm selection $\mathbb{E}^{\pi}[N_{\mathcal{T}}(a^{\dagger})]$ for different horizon

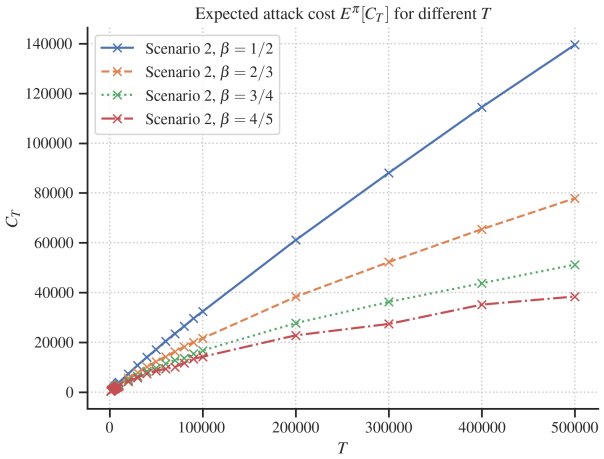


Fig. 7: Attack cost $\mathbb{E}^{\pi}[C_T]$ in scenario 2 for different β

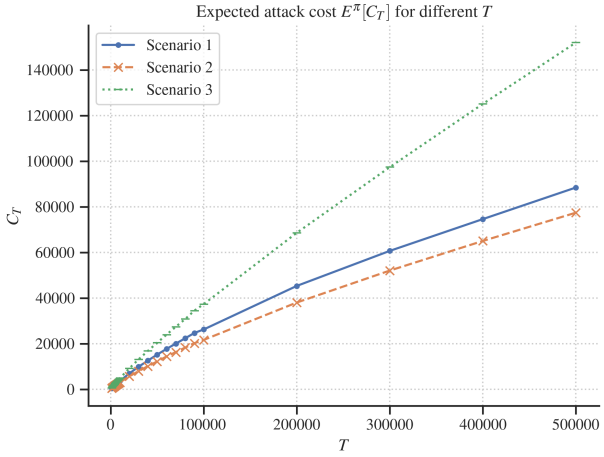


Fig. 6: Attack cost $\mathbb{E}^{\pi}[C_T]$ for different horizon T

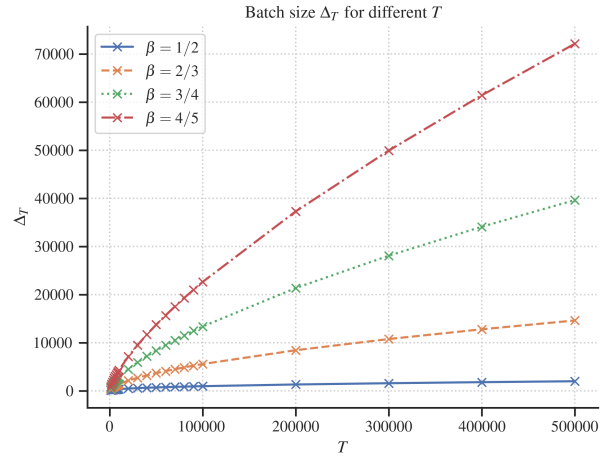


Fig. 8: Batch size Δ_T in scenario 2 due to different β

VI. Conclusion

In this paper, we have proposed three reward attacks scenarios and corresponding attack methods for stochastic non-stationary multi-armed bandit problem. Specifically, we have focused on attacking the popular batched strategy for non-stationary MAB problem, *Rexp3*, which inherently models the temporal uncertainty of the non-stationary reward structure by considering the variation budget V_T. We have proved that our attack methods are successful and can force the algorithm to pull the target arm $a^{\dagger}$ almost linear in T, while the expected attack cost is sublinear in T in all three attack scenarios. The experimental results verify our theoretical analysis. Moreover, we have derived a lower-bound of the expected attack cost when the attack is successful. This lower bound shows that our attack method is near optimal in the easy attack scenario and may introduce a small additional cost in the two general attack scenarios.

References

[1] Kwang-Sung Jun, Lihong Li, Yuzhe Ma, and Jerry Zhu. Adversarial attacks on stochastic bandits. In *Advances in Neural Information Processing Systems*, volume 31, Montréal, Canada, December 2018.

[2] Yuzhe Ma and Zhijin Zhou. Adversarial attacks on adversarial bandits. *arXiv preprint arXiv:2301.12595*, 2023.

[3] Guanlin Liu and Lifeng Lai. Efficient action poisoning attacks on linear contextual bandits. *arXiv preprint arXiv:2112.05367*, 2021.

[4] Guanlin Liu and Lifeng Lai. Action-manipulation attacks against stochastic bandits: Attacks and defense. *IEEE Transactions on Signal Processing*, 68:5152–5165, 2020.

[5] Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. In *Advances in Neural Information Processing Systems*, volume 27, Montréal, Canada, December 2014.

[6] Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In *Proceedings of International Conference on Algorithmic Learning Theory*, pages 174–188, Espoo, Finland, October 2011.

[7] Ningyuan Chen and Shuoguang Yang. Bridging adversarial and nonstationary multi-armed bandit. *arXiv preprint arXiv:2201.01628*, 2022.

[8] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.