Dual-Mandate Patrols: Bandit-Based Learning in Green Security Domains

Lily Xu[1*], **Elizabeth Bondi**[1], **Fei Fang**[2], **Andrew Perrault**[1], **Kai Wang**[1] and **Milind Tambe**[1]
[1]Harvard University
[2]Carnegie Mellon University
{lily_xu, ebondi, aperrault, kaiwang}@g.harvard.edu, feif@cs.cmu.edu, milind_tambe@harvard.edu

Abstract

Conservation efforts in green security domains to protect wildlife and forests are constrained by the limited availability of defenders (i.e., patrollers), who must patrol vast areas to protect from attackers (e.g., poachers or illegal loggers). Past work in these domains mostly focuses on exploiting historical data to build predictive models of attacker behavior and plan patrols based on these models. However, this approach ignores the potential to use patrols to improve the predictive model and maximize future reward. The few exceptions that follow a no-regret learning paradigm neglect characteristics of the reward function such as decomposability and smoothness in patrolling effort and target feature space. We (i) formulate the problem as a multi-armed bandit with Lipschitz continuity to exploit smoothness, where each arm represents a patrol strategy; (ii) present a no-regret approach for arm selection that uses an integer program to exploit decomposability; and (iii) demonstrate that our algorithm, LIZARD, improves performance on real-world poaching data from Cambodia.

1 Introduction

Green security efforts to protect wildlife, forests, and fisheries require defenders (law enforcement agencies) to conduct patrols across protected areas to guard against attacks (illegal activities) [Lober, 1992]. For example, to combat poaching, rangers conduct patrols to remove snares laid out to trap animals (Fig. 1). Green security games have been proposed to apply Stackelberg security games to green security domains such as the prevention of illegal logging, poaching, or overfishing [Fang *et al.*, 2015]. In subsequent work using real-world data, the focus has shifted to applying machine learning to predict attack hotspots, then using game-theoretic planning to design patrols based on the learned model [Nguyen *et al.*, 2016; Gholami *et al.*, 2018]. However, in many protected areas, there is inadequate or biased patrolling, disabling us from learning a reasonable adversary model in the first place [Moreto and Lemieux, 2015]. As green security games get deployed on an ever-larger scale in hundred of protected areas around the world [Xu *et al.*, 2020], addressing this information-gathering challenge becomes crucial.

Figure 1: Rangers searching for snares (right) near a waterhole (left) in Srepok Wildlife Sanctuary in Cambodia. The waterhole is frequented by deer, pig, and bison, which are targeted by poachers.

Motivated by these practical needs, we focus on conducting *dual-mandate patrols*, whose goal is to simultaneously detect illegal activities and collect valuable data to improve our predictive model, achieving higher long-term reward. These dual-mandate patrols could be conducted in an initial phase prior to the use of the aforementioned green security game approaches, as many national parks around the world suffer from highly biased or inadequate patrol data.

The key challenge when conducting dual-mandate patrols is the exploration–exploitation tradeoff: whether to follow the best patrol strategy in history or conduct new patrols that may help us get a better understanding of the attackers. Some recent work proposes to use multi-armed bandits (MAB) to formulate the problem [Xu *et al.*, 2016; Gholami *et al.*, 2019], but these initial attempts neglect characteristics of the reward function such as decomposability and smoothness in patrolling effort and target feature space where each target represents a region in the protected area. Further, we show that these approaches often require unrealistically long time horizons to achieve good performance; in the real world, these initial losses are less tolerable and can lead to stakeholders losing trust and abandoning such patrol-assistance systems.

In this paper, we address real-world characteristics of green security domains to design dual-mandate patrols. We account for (i) decomposability of the reward function, (ii) smoothness of decomposed reward function over features,

*Contact Author

(iii) monotonicity, and (iv) integration of historical data. Our algorithm, which we call LIZARD, is a decomposable bandit with Lipschitz continuity. We provide theorems to show a no-regret bound for the algorithm. We show that LIZARD beats existing algorithms in domain-realistic settings on both synthetic data that approximate real-world conditions and real poaching data from Cambodia.

1.1 Related Work

Green security domains have been extensively modeled as green security games [Fang *et al.*, 2016; Kamra *et al.*, 2018; Mc Carthy *et al.*, 2016; Haskell *et al.*, 2014]. In these games, resource-constrained defenders protect large areas (e.g., forests, savannas, wetlands) from adversaries who repeatedly attack these areas. Thus, most work has focused on learning attacker behavior models from historical data [Nguyen *et al.*, 2016; Gholami *et al.*, 2018] and using these models to plan patrols with limited lookahead to prevent future attacks [Xu *et al.*, 2017; Fang *et al.*, 2015].

Despite their successful deployment in some conservation areas, researchers have recognized that it is not always possible to have abundant historical data or exact knowledge of attacker payoff [Xu *et al.*, 2016; Gholami *et al.*, 2019]. Xu et al. [2016] follow an online learning paradigm to play against an adversarial attacker. The model is then solved using FPL-UE, a variant of Follow the Perturbed Leader (FPL) algorithm [Kalai and Vempala, 2005]. Gholami et al. [2019] propose a hybrid model, MINION, that chooses between the FPL-UE algorithm and a supervised learning model. However, in both cases, they ignore domain characteristics such as feature similarity between targets and do not incorporate existing historical data into the online learners.

There is a plethora of work in MAB [Bubeck *et al.*, 2012]. For stochastic MAB with finite arms, the standard UCB1 algorithm [Auer *et al.*, 2002] always chooses the arm with the highest upper confidence bound and achieves no-regret, i.e., the expected reward is sublinearly close to always pulling the best arm in hindsight. For continuous arms, Kleinberg et al. [2019] introduce Lipschitz MAB, which assume a Lipschitz-continuous payoff function. They propose the *zooming algorithm*, which extends UCB1 with an adaptive refinement step. Later work applies Lipschitz-continuity assumptions to bandits to generate tighter confidence bounds [Bubeck *et al.*, 2011; Magureanu *et al.*, 2014]. Another extension of UCB1 is the CUCB algorithm [Chen *et al.*, 2016], which solves a combinatorial case of MAB. Our algorithm outperforms these algorithms when applied directly to the problem of interest, as they do not fully exploit historical data and characteristics of the reward function.

2 Problem Description

The protected area for which we must plan patrols is discretized into N targets, each associated with a feature vector $\vec{y}_i \in \mathbb{R}^K$, which is static within the time horizon T. In the poaching prevention domain, for example, the K features include geospatial characteristics such as distance to river, forest cover, animal density, and slope.

In each round, the defender determines an *effort vector* $\vec{\beta} = (\beta_1, ..., \beta_N)$ which specifies the amount of effort to be spent on each target. β_i can represent, for example, the number of hours spent on foot patrolling in target i. The defender has a budget B for the total effort, i.e., $\sum_i \beta_i \leq B$.

The reward of a patrol corresponds to the total number of targets where attacks were detected and prevented [Critchlow *et al.*, 2015]. Let the expected reward of a patrol vector $\vec{\beta}$ be $\mu(\vec{\beta})$. Our objective is to specify an effort vector $\vec{\beta}^{(t)}$ for each round in an online fashion to minimize regret with respect to the optimal effort vector $\vec{\beta}^*$, where regret is defined as $T\mu(\vec{\beta}^*) - \sum_{t=0}^{T} \mu(\vec{\beta}^{(t)})$.

2.1 Domain Characteristics

We exploit the following four characteristics of green security domains to direct our approach.

Decomposability. The overall expected reward for the defender is decomposable and additive. We represent the expected reward of a patrol to the defender for target i as a function $\mu_i(\beta_i) \in [0, 1]$. We define random variables $X_i^{(t)}$ as the observed reward (attack or no attack) from target i at time t, with corresponding effort $\beta_i^{(t)}$. For executing a patrol with effort $\vec{\beta}$ across all targets, the expected composite reward is decomposable and additive, as $\mu(\vec{\beta}) = \sum_{i=1}^{N} \mu_i(\beta_i)$.

Lipschitz-continuity. As discussed, the expected reward to the defender is given by the function $\mu_i(\beta_i)$, which is dependent on effort β_i. Furthermore, the expected reward depends on the feature vector $\vec{y}_i$ of that target, that is $\mu_i(\beta_i) = \widetilde{\mu}(\vec{y}_i, \beta_i)$ for all i. We assume $\widetilde{\mu}(\cdot, \cdot)$ to be Lipschitz-continuous in effort and feature space over some distance function $\mathcal{D}$, such as Euclidean distance. That is, two distinct targets in the protected area with identical features will have identical reward functions, and two targets i and j with features $\vec{y}_i$, $\vec{y}_j$ and effort β_i, β_j have reward functions that differ by no more than

$$|\widetilde{\mu}(\vec{y}_i, \beta_i) - \widetilde{\mu}(\vec{y}_j, \beta_j)| \leq L \cdot \mathcal{D}((\vec{y}_i, \beta_i), (\vec{y}_j, \beta_j)) \qquad (1)$$

for some Lipschitz constant L.

Monotonicity. The more effort spent on a target, the higher the expected reward. That is, we assume $\mu(\beta_i)$ is monotonically non-decreasing in β_i. Additionally, we assume that 0 effort corresponds with 0 reward, as defenders cannot prevent attacks on targets they do not visit.

Historical data. Finally, many conservation areas have data from past patrols, which we use to warm start the online learning algorithm.

3 LIZARD Online Learning Algorithm

We present our algorithm, LIZARD (LIpschitZ Arms with Reward Decomposability), for online learning in green security domains.

Standard bandit algorithms suffer from the curse of dimensionality. The set of arms has size J^N. Instead, we cast the problem as a multi-armed bandit problem with decomposability. At each iteration, we choose a patrol scheme $\vec{\beta}$ that satisfies the budget constraint and observe the patrol outcome of

each target i under the chosen effort β_i. An *arm* is one effort level β_i on a specific target i; a *super arm* is $\vec{\beta}$, the collection of N arms. By tracking decomposed rewards, we only need to track observations from NJ arms. This require us to maintain exponentially fewer samples, and we can also transfer knowledge between super arms when similar effort levels and targets appear between super arms.

3.1 Upper Confidence Bounds with Similarity

We take an upper confidence bound (UCB) approach where the rewards are tracked separately for different targets. Due to additivity of the reward function, the UCB for the super arm is the sum of the individual UCBs of the arms. We show that we can incorporate Lipschitz-continuity of the reward functions into the UCB of each arm to tighten the confidence bounds.

Let $\bar{\mu}_t(i,j) = \texttt{reward}_t(i,\psi_j)/n_t(i,\psi_j)$ be the average reward of target i at effort ψ_j given cumulative empirical reward $\texttt{reward}_t(i,\psi_j)$ over $n_t(i,\psi_j)$ arm pulls. Let $r_t(i,j)$ be the *confidence radius* at timestep t defined as

$$r_t(i,j) = \sqrt{\frac{3\log(t)}{2n_t(i,\psi_j)}}\,. \tag{2}$$

We distinguish between UCB and a term we call self-UCB. The self-UCB of a target i with effort level j at time t is the UCB of an arm based only on its own observations, given by

$$\text{SELFUCB}_t(i,j) = \bar{\mu}_t(i,j) + r_t(i,j)\,. \tag{3}$$

The UCB of arm (i,j) is then computed by taking the minimum of the bounds of all self-UCBs as applied to the arm. These bounds are determined by adding the distance between arm (i,j) and the other arm to the self-UCB:

$$\text{UCB}_t(i,j) = \min_{\substack{u\in[N]\\ v\in\Psi_i}} \{\text{SELFUCB}_t(i,j) + L\mathcal{D}((\vec{y}_i,\psi_j),(\vec{y}_u,\psi_v))\} \tag{4}$$

which exploits Lipschitz continuity between the targets. We define $\text{UCB}_t(i,0) = 0$ for all $i \in [N]$ due to the assumption that zero effort yields zero reward. To address the fact that the reward function is monotonically non-decreasing, we modify the UCB to consider distance as $\max\{0, \mathcal{D}((i,\psi_{i,j}),(u,\psi_{u,v}))\}$.

3.2 Arm Selection

With the computed UCBs, arm selection can be framed as a knapsack optimization problem. We maximize the sum of the UCBs subject to a budget constraint (total effort).

$$\begin{aligned}
\max_{z} \quad & \sum_{i\in[N]}\sum_{j\in\Psi_i} z_{i,j}\cdot \text{UCB}_t(i,j) && (\mathcal{P})\\
\text{s.t.} \quad & z_{i,j}\in\{0,1\} && \forall i\in[N], j\in\Psi_i\\
& \sum_{j\in\Psi_i} z_{i,j} = 1 && \forall i\in[N]\\
& \sum_{i\in[N]}\sum_{j\in\Psi_i} z_{i,j}\psi_{i,j} \le B
\end{aligned}$$

The mathematical program to select which arm to use is given by the mixed integer linear program $\mathcal{P}$ with NJ variables and $N+1$ constraints. There is one auxiliary variable $z_{i,j}$ for each level of patrol effort for each target. Pseudocode for the LIZARD algorithm is given in Algorithm 1.

Algorithm 1: LIZARD

Inputs: Number of targets N, time horizon T, budget B, target features $\vec{y}_i$, Lipschitz constant L
Discretization levels $\Psi_i = \{0,1\}\ \forall i\in[N]$, gap $\psi = 1$
$T_k = \frac{N}{L^2 2^{3k}}\log\frac{N}{L^2 2^{3k}}\ \forall k\in\mathbb{N}\cup\{0\}$
// Initialize arms
$n(i,\psi_j) = 0, \texttt{reward}(i,\psi_j) = 0 \quad \forall i\in[N], \psi_j\in\Psi_i$
for $t = 1,2,\ldots,T$ **do**
  if $t > \sum_{j=0}^{k-1} T_j$ **then**
    Set $\psi = 2^{-k}$ and $\Psi_i = \{0,\psi,...,1\}\ \forall i$
  Solve $\mathcal{P}(n,\texttt{reward},\Psi,B,N,T)$ to find the $\vec{\beta}$ that maximizes UCB
  for $i = 1,2,\ldots,N$ **do**
    Sample binary reward $X_i^{(t)}$ from $\mu_i(\beta_i)$
    // Update rewards
    $\texttt{reward}(i,\beta_i) = \texttt{reward}(i,\beta_i) + X_i^{(t)}$
    $n(i,\beta_i) = n(i,\beta_i) + 1$

4 Regret Analysis

We provide the regret bound for the LIZARD algorithm. Our regret bound matches the regret bound of the zooming algorithm of Kleinberg et al. [2019] with covering dimension $d = 1$, although the zooming algorithm does not address combinatorial arm selection. This also extends the line of research on combinatorial bandits, as Chen et al. [2016] ignores Lipschitz-continuous input under constraints.

To achieve a no-regret guarantee with an infinite time horizon, we need varied discretization. As shown in Algorithm 1, we gradually switch to a finer discretization.

Theorem 1. *Given the minimum discretization gap ψ, number of targets N, and Lipschitz constant L, the regret bound of Algorithm 1 with* SELFUCB *can be given by*

$$\textit{Reg}(T) \le O\left(L^{\frac{4}{3}} N T^{\frac{2}{3}} (\log T)^{\frac{1}{3}}\right) \tag{5}$$

This regret bound matches the bound given by the zooming algorithm with covering dimension $d = 1$. In our case, however, our input variable is N-dimensional, which usually falls into a metric space with covering dimension $d = N$, and the regret bound for metric space with covering dimension d is $O(T^{\frac{d+1}{d+2}}(\log T)^{\frac{1}{d+2}})$ [Kleinberg *et al.*, 2008]. That implies our decomposed algorithm can successfully decouple the N-dimensional metric space into individual sub-dimensions and still enjoy the smaller regret order, showcasing the power of decomposibility. This theorem also extends the work of the combinatorial multi-armed bandit problem [Chen *et al.*, 2016], where we allow the combinatorial arm selection to be continuous instead of discrete.

5 Experiments

We conducted experiments using both synthetic data and real poaching data. The results validate the ability of our algorithm to effectively learn within a practical time horizon.

Table 1: Performance when varying parameters, throughout which LIZARD is nearly always superior.

	at $T = 200$				at $T = 500$				at $T = 200$				at $T = 500$			
$N =$	25	25	100	100	25	25	100	100	25	25	100	100	25	25	100	100
$B =$	1	5	5	10	1	5	5	10	1	5	5	10	1	5	5	10
	SYNTHETIC DATA								REAL-WORLD DATA							
LIZARD	**27.1**	**22.0**	**29.7**	**15.7**	**42.9**	**34.4**	**41.7**	**57.0**	**66.7**	11.0	**51.9**	**43.9**	**70.6**	**50.6**	**64.5**	**60.7**
CUCB	−12.0	14.5	2.5	−2.4	−17.9	24.1	−7.6	16.8	44.6	**34.1**	27.6	40.3	63.3	36.1	51.9	59.7
MINION	−25.3	−3.7	−11.8	−0.7	−12.8	−11.0	−18.7	−5.7	−36.2	−12.7	−14.3	−9.4	−74.6	−46.6	−3.9	−8.8
Zooming	−22.3	−7.9	−23.7	−23.2	−23.1	−1.0	−21.5	−18.1	−3.8	12.1	−37.0	−21.4	0.5	14.3	−31.4	−21.4

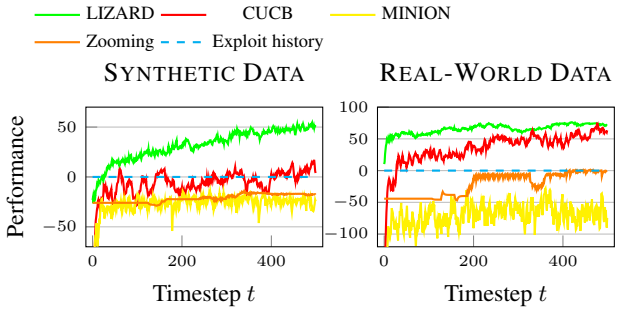

Figure 2: Performance over time, averaged over 30 trials with $N = 25$, $B = 1$. LIZARD (green) achieves the highest performance.

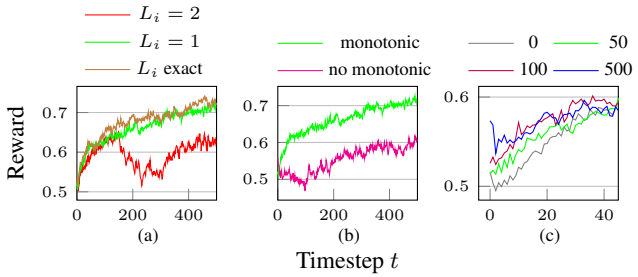

Figure 3: Impact of (a) different Lipschitz constants L_i, (b) monotonicity assumption, (c) varying amount of historical data. The green line in each plot depicts the setting used in LIZARD. Run on synthetic data with $N = 25$, $B = 1$, averaged over 30 trials.

We consider a patrol planning problem with $N = 25$ or 100 targets (each a 1 sq. km grid cell), representing the region reachable from a single patrol post, and time horizon $T = 500$ representing a year and a half of patrols. Effort $\beta_i = 1$ corresponds to the total amount of effort in a day that a single team can spend patrolling, such as 5 hours. The budget is therefore the number of teams, e.g., $B = 5$ corresponds to 5 teams of rangers. We use 50 timesteps of historical data.

Real-world data. We leverage real patrol data from Srepok Wildlife Sanctuary in Cambodia to learn the reward function, as dependent on features and effort [Xu *et al.*, 2020]. We train a machine learning classifier to predict whether rangers will detect poaching activity at each target for a given effort level. We then generate a piecewise-linear approximation of the reward function.

Synthetic data. To produce synthetic data, we generate piecewise-linear functions that mimic the behavior of real-world reward functions. We then define feature similarity as the maximum difference between the reward functions across all effort levels.

Algorithms. We compare to three baselines, which are all online algorithms: combinatorial upper confidence bound (*CUCB*) [Chen *et al.*, 2016], *zooming* [Kleinberg *et al.*, 2019], *MINION* hybrid [Gholami *et al.*, 2018], and *exploit history*. The last naively exploits historical data with a static strategy. We compute the optimal strategy exactly by solving a mixed-integer program over the piecewise-linear reward functions, subject to the budget constraint.

Fig. 2 shows performance on both real-world and synthetic data, evaluated as the reward achieved at timestep t, where the reward of historical exploit is 0 and that of optimal is 1. On the real-world data, LIZARD provides a significant advantage over CUCB in the early rounds and remains 7.3% higher by the time horizon. Table 1 shows the performance of each algorithm at $T = 200$ and 500, with varying values of N and B. LIZARD beats all baselines by timestep 500 in every scenario.

We return to the four characteristics of green security domains discussed in Sec. 2.1 and show that integrating each feature improves LIZARD. **Decomposability:** CUCB, a naive decomposed bandits algorithm, surpasses non-decomposed algorithms MINION and zooming and demonstrates the value of decomposition. Notably, non-decomposed algorithms MINION and zooming perform worse than historical exploit. **Lipschitz-continuity:** Fig. 3(a) reveals the value of information gained from knowing the exact value of the Lipschitz constant in each dimension (L_i exact). As shown, significantly overestimating L_i ($L_i = 2$) hinders performance. **Monotonicity:** Fig. 3(b) shows that the monotonicity assumption adds a significant boost to performance. **Historical data:** Fig. 3(c) demonstrates the value of adding historical information in early rounds.

6 Conclusion

We present LIZARD, an integrated algorithm for online learning in green security domains. Our experimental results validate the domain-inspired approach of treating real-world conditions not as constraints, but rather as useful features that enable us to improve over standard machine learning algorithms. Critically, we define our benchmark not solely by theoretical guarantees under infinite time horizons, but rather on significantly improved empirical performance within realistic time frames and practical constraints.

References

[Auer *et al.*, 2002] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

[Bubeck *et al.*, 2011] Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. X-armed bandits. *J. of Machine Learning Research*, 12(May), 2011.

[Bubeck *et al.*, 2012] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

[Chen *et al.*, 2016] Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *J. of Machine Learning Research*, 17(1):1746–1778, 2016.

[Critchlow *et al.*, 2015] Rob Critchlow, Andrew J Plumptre, Margaret Driciru, Aggrey Rwetsiba, Emma J Stokes, Charles Tumwesigye, Fred Wanyama, and CM Beale. Spatiotemporal trends of illegal activities from ranger-collected data in a ugandan national park. *Conservation Biology*, 29(5):1458–1470, 2015.

[Fang *et al.*, 2015] Fei Fang, Peter Stone, and Milind Tambe. When security games go green: Designing defender strategies to prevent poaching and illegal fishing. In *Proc. of IJCAI-15*, 2015.

[Fang *et al.*, 2016] Fei Fang, Thanh H Nguyen, Rob Pickles, Wai Y Lam, Gopalasamy R Clements, Bo An, Amandeep Singh, Milind Tambe, and Andrew Lemieux. Deploying paws: Field optimization of the protection assistant for wildlife security. In *Proc. of IAAI-16*, 2016.

[Gholami *et al.*, 2018] Shahrzad Gholami, Sara Mc Carthy, Bistra Dilkina, Andrew Plumptre, Milind Tambe, Margaret Driciru, Fred Wanyama, Aggrey Rwetsiba, Mustapha Nsubaga, Joshua Mabonga, et al. Adversary models account for imperfect crime data: Forecasting and planning against real-world poachers. In *Proc. of AAMAS-18*, pages 823–831, 2018.

[Gholami *et al.*, 2019] Shahrzad Gholami, Amulya Yadav, Long Tran-Thanh, Bistra Dilkina, and Milind Tambe. Don't put all your strategies in one basket: Playing green security games with imperfect prior knowledge. In *Proc. of AAMAS-19*, pages 395–403, 2019.

[Haskell *et al.*, 2014] William Haskell, Debarun Kar, Fei Fang, Milind Tambe, Sam Cheung, and Elizabeth Denicola. Robust protection of fisheries with compass. In *Proc. of IAAI-14*, 2014.

[Kalai and Vempala, 2005] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *J. of Computer and System Sciences*, 71(3):291–307, 2005.

[Kamra *et al.*, 2018] Nitin Kamra, Umang Gupta, Fei Fang, Yan Liu, and Milind Tambe. Policy learning for continuous space security games using neural networks. In *Proc. of AAAI-18*, 2018.

[Kleinberg *et al.*, 2008] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proc. of STOC-08*, pages 681–690. ACM, 2008.

[Kleinberg *et al.*, 2019] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Bandits and experts in metric spaces. *Journal of the ACM (JACM)*, 2019.

[Lober, 1992] Douglas J Lober. Using forest guards to protect a biological reserve in costa rica: one step towards linking parks to people. *Journal of Environmental Planning and Management*, 35(1):17–41, 1992.

[Magureanu *et al.*, 2014] Stefan Magureanu, Richard Combes, and Alexandre Proutiere. Lipschitz bandits: Regret lower bounds and optimal algorithms. In *Proc. of COLT-14*, 2014.

[Mc Carthy *et al.*, 2016] Sara Marie Mc Carthy, Milind Tambe, Christopher Kiekintveld, Meredith L Gore, and Alex Killion. Preventing illegal logging: Simultaneous optimization of resource teams and tactics for security. In *Proc. of AAAI-16*, 2016.

[Moreto and Lemieux, 2015] William D Moreto and Andrew M Lemieux. Poaching in uganda: Perspectives of law enforcement rangers. *Deviant Behavior*, 36(11):853–873, 2015.

[Nguyen *et al.*, 2016] Thanh H Nguyen, Arunesh Sinha, Shahrzad Gholami, Andrew Plumptre, Lucas Joppa, Milind Tambe, Margaret Driciru, Fred Wanyama, Aggrey Rwetsiba, Rob Critchlow, et al. Capture: A new predictive anti-poaching tool for wildlife protection. In *Proc. of AAMAS-16*, pages 767–775, 2016.

[Xu *et al.*, 2016] Haifeng Xu, Long Tran-Thanh, and Nicholas R Jennings. Playing repeated security games with no prior knowledge. In *Proc. of AAMAS-16*, pages 104–112, 2016.

[Xu *et al.*, 2017] Haifeng Xu, Benjamin Ford, Fei Fang, Bistra Dilkina, Andrew Plumptre, Milind Tambe, Margaret Driciru, Fred Wanyama, Aggrey Rwetsiba, Mustapha Nsubaga, et al. Optimal patrol planning for green security games with black-box attackers. In *Proc. of GameSec-17*, pages 458–477. Springer, 2017.

[Xu *et al.*, 2020] Lily Xu, Shahrzad Gholami, Sara Mc Carthy, Bistra Dilkina, Andrew Plumptre, Milind Tambe, Rohit Singh, Mustapha Nsubuga, Joshua Mabonga, Margaret Driciru, et al. Stay ahead of poachers: Illegal wildlife poaching prediction and patrol planning under uncertainty with field test evaluations. In *Proc. of ICDE-20*, 2020.