Don’t Put All Your Strategies in One Basket: Playing Green Security Games with Imperfect Prior Knowledge

Shahrzad Gholami[1], Amulya Yadav[2], Long Tran-Thanh[3], Bistra Dilkina[1], Milind Tambe[1],
[1]University of Southern California, {sgholami, dilkina, tambe}@usc.edu,
[2]Penn State University, {amulya}@psu.edu,
[3]University of Southampton, {l.tran-thanh}@soton.ac.uk

ABSTRACT

Security efforts for wildlife monitoring and protection of endangered species (e.g., elephants, rhinos, etc.) are constrained by limited resources available to law enforcement agencies. Recent progress in Green Security Games (GSGs) has led to patrol planning algorithms for strategic allocation of limited patrollers to deter adversaries in environmental settings. Unfortunately, previous approaches to these problems suffer from several limitations. Most notably, (i) previous work in GSG literature relies on exploitation of error-prone machine learning (ML) models of poachers’ behavior trained on (spatially) biased historical data; and (ii) online learning approaches for repeated security games (similar to GSGs) do not account for spatio-temporal scheduling constraints while planning patrols, potentially causing significant shortcomings in the effectiveness of the planned patrols. Thus, this paper makes the following novel contributions: (I) We propose MINION-sm, a novel online learning algorithm for GSGs which does not rely on any prior error-prone model of attacker behavior, instead, it builds an implicit model of the attacker on-the-fly while simultaneously generating scheduling-constraint-aware patrols. MINION-sm achieves a sublinear regret against an optimal hindsight patrol strategy. (II) We also propose MINION, a hybrid approach where our MINION-sm model and an ML model (based on historical data) are considered as two patrol planning experts and we obtain a balance between them based on their observed empirical performance. (III) We show that our online learning algorithms significantly outperform existing state-of-the-art solvers for GSGs.

KEYWORDS

Green Security Games; Game Theory; Online Learning; Adversarial Bandits; Machine Learning; Wildlife Protection

ACM Reference Format:
Shahrzad Gholami, Amulya Yadav, Long Tran-Thanh, Bistra Dilkina, Milind Tambe. 2019. Don’t Put All Your Strategies in One Basket: Playing Green Security Games with Imperfect Prior Knowledge. In Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS’19), Montreal, Canada, May 13-17, 2019, IFAAMAS, 9 pages.

1 INTRODUCTION

Poaching is a serious threat to wildlife conservation around the world and can lead to the extinction of several important species and complete destruction of ecosystems [6]. Specifically, as a result of poaching, tigers are now found in less than 7% of their historical range, and three out of nine tiger subspecies have already been driven to extinction [9, 28]. Not only are the effects of poaching detrimental to animal species, the illegal trade of wildlife also helps fund armed conflict by extremist groups around the world, and it has become a 213 billion dollar industry [1].

As a result, efforts have been made by law enforcement agencies (i.e., park rangers) in many countries to protect endangered animals from poaching. The most direct and commonly used approach is conducting foot patrols [23]. However, given the limited human resources and the vast areas in need of protection, improving the efficiency of the patrols remains a major challenge [14].

Security games are well known to be effective models of protecting valuable targets against an adversary and have been explored extensively at AAMAS [3, 17, 20, 24]. Recently, there has been a lot of progress in the field of Green Security Games (GSGs), which has led to the development of several algorithms which serve as game-theoretic decision aids to optimize the use of limited human patrol resources to combat poaching [11, 12, 15, 16, 27]. The basic premise behind most of this work is that repeated interactions between patrollers and poachers provides the opportunity to gather data which can be used to learn models of poacher behavior [7]. Thus, most previous algorithms design patrol routes assuming poachers attack according to a fixed "*learnable*" model (which could either have a functional form [7, 27], or it could be a black-box model [11, 31]). Most of these algorithms then try to solve a repeated Stackelberg game, where the patrollers (defenders) conduct randomized patrols against poachers (attackers) while balancing the priorities of different locations in the park. Unfortunately, this approach suffers from serious shortcomings, which impedes usability in the real-world.

In particular, the GSG approach can be expected to provide good results only if the collected historical data is a good representation of the actual poaching activities that occurred in the past (and those that will occur in the future), which would allow us to learn an accurate model for attacker behavior. Unfortunately, in the wildlife poaching domain, it is extremely difficult to know ahead of time whether the learned model of attacker behavior is accurate or not (over the entire protected area). Due to logistical issues, several patrollers only conduct patrols either close to their sparsely spread patrol posts, or in areas that are easily accessible by them. This issue is so prevalent that it has a special name in ecological research: the silent victim problem [22]. As a result, the poaching data collected in these domains may be highly biased (in a spatial sense). For example, Figure 1 shows the patrol coverage heatmap in Murchison Falls National Park in Uganda where the color shade indicates the

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

intensity of coverage in the past (darker color correspond to higher patrol levels). Due to such biased data collection, the data sample might not fairly represent the entire space of the problem [21] and the learned model of the attacker behavior might have different prediction accuracy in the park areas that have high vs. low patrol densities in Figure 1. Thus, it may or may not be optimal to rely on learned models of attacker behavior in patrol planning, and there is no straightforward method to determine the optimal course of action prior to deployment, i.e., whether to use the learned model (or not) in patrol planning. Moreover, the sub-optimal choice may lead to arbitrary losses for the defender (as confirmed in our evaluation).

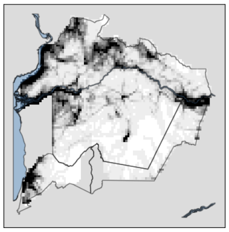


Figure 1: Non-uniform historical patrol coverage in Murchison Falls National Park implies biases in data collection by park rangers

This paper makes three significant contributions to address these shortcomings in the GSG approach. First, we propose a novel online learning algorithm, MINION-sm (a submodule of MultI-expert oNline model for constraIned patrol plaNning), which does not rely on any prior model of attacker behavior, instead it builds an implicit model of the attacker on-the-fly. MINION-sm frames the repeated security game as an adversarial combinatorial bandit problem and trades off exploitation of well-known high-reward patrol routes with exploration of untried patrol routes to provide an online policy for generating randomized patrols. It also takes into account scheduling constraints for defender. We prove that MINION-sm achieves sublinear regret against an optimal hindsight policy, which is the best that an adversarial bandit algorithm can hope for. Second, to model situations where the trained machine learning (ML) models may be a good representation of actual poacher behavior, we propose MINION (MultI-expert oNline model for constraIned patrol plaNning), an online learner which utilizes any benefits that can be achieved from exploitation of the learned ML models. Specifically, MINION considers our MINION-sm model and an ML model (based on historical data) as two patrol planning experts and dynamically combines the recommendations of both these experts to provide even better empirical performance. Finally, we evaluate our online learning algorithms and show that they outperform existing state-of-the-art GSG solvers by ∼100% on a variety of simulated game settings.

2 RELATED WORK

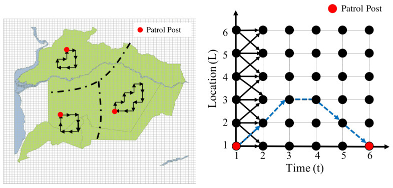
We now elaborate on how our work compares to prior literature on Green Security Games (GSGs) and repeated Stackelberg Security Games. There has been a lot of effort in GSGs at learning models of attacker behavior from historical patrolling data, which has then been used inside Stackelberg game solvers [30]. A lot of initial effort in this direction assumed attackers behaved according to parametric models, e.g., Quantal Response [26], Subjective Utility Quantal Response [7, 8, 33], SHARP model[15], etc., and tried to learn model parameters which best fit the historical data. Unfortunately, the assumption of having a fixed model of attacker behavior is quite restrictive and is not robust to any errors in our knowledge about the model type. As a result, there has also been recent effort at learning black-box machine learning models of attacker behavior from past patrolling data which can be used to plan patrols [11, 13, 31]. Sinha et al. [29] proved sample complexity results for learning in Stackelberg Security Games which showed that a huge amount of prior knowledge (historical data) is required to achieve good performance in the GSG approach. Moreover, as mentioned in the introduction, the poaching data collected in these domains is highly biased (in a spatial sense) and as a result, planning patrols based on this data may lead to arbitrary losses. Moreover, in our work, we propose online learning approaches which do not rely on past data to learn attacker models (or at least trade off between (i) relying on past data; and (ii) online learning approaches), and as we show in our evaluation section, this may lead to significant improvements in solution quality.

In the field of repeated Stackelberg Security Games, Klima et al. [18, 19] solved the problem of patrol planning for repeated border patrols with online learning algorithms. They provided an experimental analysis of the performance of several well-known online learning algorithms. However, they emphasized empirical results and they do not provide theoretical analysis. Balcan et al. [2] solved repeated Stackelberg Security Games with varying attacker types captured with different payoff matrices and proposed an online learning approach, but they assumed perfect rationality of attackers and complete knowledge of the payoff matrices, which is unrealistic to expect in the wildlife poaching domain. Blum et al. [4] optimizes defender strategy with no prior knowledge in repeated Stackelberg Security Games but they consider a query based model, where they try to learn good approximations of the payoff matrices with the least amount of queries, which is an orthogonal setting compared to our work.

In another closely related work, Xu et al. [32] proposed an online learning approach to solving repeated Stackelberg Security Games under no assumptions on the adversary's behavior. While the problem that we are solving in this paper is similar to the one considered in [32], their work do not take into account spatio-temporal scheduling constraints while planning patrols. As a result, the generated patrols are un-implementable in the real-world, and thus, their approach is not easily usable in the real-world. In our work, we ensure that our proposed algorithms generates patrols which take into account several important scheduling constraints. Moreover, Xu et al. [32] do not take into consideration any prior knowledge and learn models from scratch, whereas our approach learns whether models based on prior knowledge are better (or worse) than models learned on-the-fly and takes decisions accordingly.

3 PROBLEM FORMULATION

Game Description We now describe the patrol route planning problem considered in this paper. The entire wildlife park area is planned to be protected in within the patrol plan horizon of T and is divided into L distinct locations (grid cells). One of these locations is designated as the patrol post, w.l.o.g., we treat location 1 as the patrol post throughout the paper. We cast the patrol route planning problem as a repeated game between a defender (having a single



(a) Patrol Route Schema (b) Time-unrolled Graph

Figure 2: Patrol Planning in Green Security Games

patrol team) and an attacker (having M poachers) on $L \times T$ targets (as we explain below). The game proceeds in D sequential rounds. We assume that both the defender and the attacker move simultaneously in each round of the game. However, consistent with the literature on repeated games, the defender (and the attacker) may use their opponent's actions in prior rounds to optimize the defender (and the attacker) strategy in the current round. In each round, the defender plans a patrol route $\{(l_1, t_1), \ldots (l_T, t_T)\}$ for her patrol team, where l and t denote the location and time, respectively. On the other hand, the attacker chooses a set of M distinct $< l, t >$ pairs (or targets), i.e., a location and time pair for each of his M poachers to attack. As a result of choosing these actions, the defender gets a payoff U_i^c for each covered (patrolled) target which was attacked by the attacker, and a payoff of U_i^u ($U_i^u \leq U_i^c$) for each target which was uncovered (unpatrolled) but was attacked by a poacher. We assume that both $U_i^c, U_i^u \in [-0.5, 0.5]$ and their exact value is unknown to the defender. The goal of the defender is to design "good" patrol routes (we formalize our exact objective later) against an adaptive attacker.

Note that our problem setup is slightly different from standard GSGs, where the primary goal of the defender is to uncover snares left by the poachers [7, 8]. As a result, a lot of emphasis is placed in prior GSG work on imperfect detection of snares by the defender when she patrols a location [26]. While this is an important real-world issue, we abstract away this complication by assuming that our defender can detect snares perfectly. In the real-world, this can be achieved by dividing the wildlife park into smaller-sized areas.

Defender's Spatio-Temporal Constraints Due to real-world challenges, the patrol route (or pure strategy) chosen by the defender must satisfy certain spatio-temporal constraints. First, locations patrolled in consecutive time steps in the patrol route must correspond to geographically neighboring locations, otherwise it is not physically possible for the patrol team to implement that patrol. Second, any patrol route must originate from and return to the patrol post (i.e., location 1), as shown in Figure 2(a). We further assume that the patrol team can traverse at most T locations in each round of the game ($T << L$), and thus, the length of every patrol route must be exactly T. To simplify exposition, we model this problem using a time-unrolled graph $G(u, e)$, with LT nodes, as demonstrated in Figure 2(b). Each node u represents a pair $< l, t >$, i.e., location $l \in [L]$ at time $t \in [T]$ and each directed edge, e, connects a location at time t to another accessible location at time $t + 1$. A defender pure strategy in this time-unrolled graph is a "feasible" path (i.e., path which satisfies spatio-temporal constraints) of length T, e.g., the blue dashed line in Figure 2(b) denotes a possible pure strategy for the defender. Similarly, an attacker pure strategy in this time-unrolled graph is a set of M graph nodes (note that each graph node is an $< l, t >$ pair).

Defender's Objective Note that the defender's payoffs in a given round depend only on whether her chosen pure strategy (patrol route) covers (time-unrolled) graph nodes which were chosen for attack by the attacker (this is by definition of the terms U_l^c and U_l^u). On the other hand, they do not depend on the exact ordering in which the graph nodes were patrolled. The time-indexed ordering of graph nodes (as required by the spatio-temporal constraints) in defender patrols is important only to ensure implementability of those patrols.

Thus, to formally define the defender's objective, we represent the defender's patrol route (pure strategy) as a binary vector $v \in \{0, 1\}^{LT}$ s.t. $\|v\|_1 = T$, where each entry v_l is 1 if defender protects graph node l in that patrol route, and 0 otherwise. We reiterate that all such possible binary vectors v may not correspond to implementable patrol routes. However, corresponding to every feasible patrol route, there is exactly one binary vector v. We use $\mathcal{V}$ to denote the set of all such valid pure strategies for the defender. Similarly, we use $a \in \{0, 1\}^{LT}$ s.t. $\|a\|_1 \leq M$ to denote an attacker pure strategy, and $\mathcal{A}$ to denote the set of all attacker pure strategies.

Given the defender and attacker pure strategies at round d, v^d and a^d, the defender's utility in round d is defined as $u(v_d, a_d) = \sum_{i \in [LT]} v_{d,i} a_{d,i} U_i^c + \sum_{i \in [LT]} (1 - v_{d,i}) a_{d,i} U_i^u$, which can be rewritten as $\sum_{i \in [LT]} v_{d,i} a_{d,i} [U_i^c - U_i^u] + \sum_{i \in [LT]} a_{d,i} U_i^u = v_d \cdot r_d(a_d) + C(a_d)$. Consistent with prior work, this utility equation indicates that defender needs to increase his utility by choosing strategy v_d at each round of the game. $r_d(a_d)$ denotes the reward that depends on the adversary actions.

We aim to maximize defender's expected utility over D rounds of the game $\mathbb{E}\big[\sum_{d=1}^{D} u(v_d, a_d)\big]$; where expectation is taken over randomness of the strategy. Alternatively, we want to minimize the defender's regret as computed in equation 1. The first term in equation 1 is the static optimal hindsight strategy, and is the benchmark that we compare our algorithm against. Specifically, it shows the utility of the best fixed hindsight strategy assuming all the $r_d(a_d)$ values chosen by the adversary are apriori known. This is the standard notion of regret computation used within adversarial bandit problems. This is because there are well known results that show that it is impossible to achieve sub-linear regret against a hindsight strategy which dynamically changes in every round [5]. Thus, the static optimal hindsight strategy is used as the benchmark, as it allows for greater computational tractability.

$$\begin{aligned} R_D &= \max_{v \in \mathcal{V}} \sum_{d=1}^{D} u(v, a_d) - \mathbb{E}\Big[\sum_{d=1}^{D} u(v_d, a_d)\Big] \\ &= \max_{v \in \mathcal{V}} \sum_{d=1}^{D} v \cdot r_d - \mathbb{E}\Big[\sum_{d=1}^{D} v_d \cdot r_d\Big] \end{aligned} \tag{1}$$

4 PATROL ROUTE PLANNING WITH IMPERFECT PRIOR KNOWLEDGE

In GSG settings, attackers' behavior is usually represented by explicit models determined by machine learning methods that consume real-world historical data on illegal activities. These explicit models provide predictions on the likelihood of attacks on different targets based on the past adversarial actions detected by defenders who conduct patrols repeatedly to protect the targets. Consequently, if these historical data on illegal activities (collected by the defenders) are not a representative sample from the entire space, ML models might be inaccurate in estimation of attackers' behavior and pure exploitation of such attackers' model in patrol planning models can potentially result in underestimation of attacks in unexplored portions of the space [21] and be detrimental to the defender. Although there are settings that ML models could be beneficial for patrol planning, it is extremely difficult to guarantee the accuracy of the ML models for future deployments prior to the deployment. So to minimize the risk of undesirable exploitation of inaccurate (or insufficiently accurate) ML models, we propose a meta-learning approach that incorporates an online-learner along with an ML-based patrol planning model. Note that in this paper, *prior knowledge* refers to the historical data about adversarial actions before the initial round of the game. In this section, (i) we propose an online learning approach for patrol planning when defender's strategy is constrained, no prior knowledge about past attacks are available and an implicit model of the attacker has to be learned on-the-fly, (ii) we discuss an ML-based patrol planning method where (potentially) imperfect prior knowledge is available, (iii) we outline our meta-learner approach which obtains the best patrol planning expert between the two previous methods based on their empirical performance.

4.1 Expert I: Patrol planning via online learning

To generate defender strategy based on an implicit model of the attackers, we propose an online patrol planning algorithm without any prior knowledge (i.e., historical data before the first round of the game) for constrained defender which builds upon the FPL-UE algorithm for repeated security games.

FPL-UE Algorithm The FPL-UE algorithm (follow-the-perturbed-leader with uniform exploration) proposed in [32] provides the best strategy in each round of the repeated security games by balancing exploration and exploitation. This algorithm assumes no scheduling constraints for defender and no prior knowledge about adversaries, i.e., reward $\tilde{r}_{1,i}$ in the initial round is 0 for all $i \in [N]$, where N is the number of the targets. In each round d of the experiments, a random coin is flipped to choose between exploration (with γ probability) and exploitation (with 1-γ probability) and then the defender strategy v_d is found as follows. They pick a predefined set of exploration strategies $E_{expl} = \{v_1, \ldots, v_N\}$ such that target i is protected in pure strategy v_i. If the exploration phase is selected, the algorithm assures that a strategy is chosen uniformly random from set E_{expl} and each target is covered by $\frac{\gamma}{N}$ probability. If the exploitation phase is selected, v_d is the optimized strategy based on the current estimation of the rewards, $\tilde{r}_d$ and also a perturbation element that models the noise on the reward estimations. This noise is basically a random vector $z = (z_1, \ldots, z_N)$, $z_i \sim \exp(\eta)$, independently drawn from the exponential distribution with parameter η. After the proposed strategy in each round is deployed, the reward estimation, $\tilde{r}_d$, is updated. The FPL-UE algorithm does not consider any constraints on the defender actions which makes the strategies impracticable for deployment in GSGs.

Algorithm 1: The MINION-sm Algorithm

```
   parameters: $\eta \in \mathbb{R}^+, W \in \mathbb{Z}^+, \gamma \in [0,1], st \in [LT]$ ,
               $ds \in [LT]$;
1  Initialize the estimated reward $\tilde{r}_d = 0 \in \mathbb{R}^{LT}$;
2  for $d = 1, \ldots, D$ do
3      sample $flag \in \{0,1\}$ such that $flag = 0$ with prob. $\gamma$;
4      if $flag == 0$ then
5          Let $j \in [LT]$ be a uniform randomly sampled target;
6          Draw $z_{d,i} \sim exp(\eta)$ independently for all $i \in [LT]$
             and let $z = (z_1, \ldots, z_{LT})$;
7          Let $\alpha = 0$;
8          Let $v_d$ be $[\mathcal{P}(a = st, b = j), \mathcal{P}(a = j, b = ds)]$;
9      else
10         Draw $z_{d,i} \sim exp(\eta)$ independently for all $i \in [LT]$
             and let $z = (z_1, \ldots, z_{LT})$;
11         Let $\alpha = 1$;
12         Let $v_d$ be $\mathcal{P}(a = st, b = ds)$ computed from the
             mathematical program 2;
13     end
14     Adversary picks $r_{d,i} \in [0,1]^{LT}$ and defender plays $v_d$;
15     Run $GR(\eta, w, \tilde{r}, d)$: estimate $\frac{1}{p_{d,i}}$ as $K_{d,i}$;
16     Update $\tilde{r}_{d,i} \leftarrow \tilde{r}_{d,i} + K_{d,i} r_{d,i} \mathbb{I}_{d,i}$; where $\mathbb{I}_{d,i} = 1$ for
         $v_{d,i} = 1$; $\mathbb{I}_{d,i} = 0$ otherwise;
17 end
```

MINION-sm Algorithm To overcome the limitation of the FPL-UE algorithm, we propose MINION-sm which recommends the best defender strategy in the repeated security games with scheduling constraints. Our MINION-sm algorithm outlined in Algorithm 1 assume no prior knowledge and initializes the estimation of the reward as 0 (line 1). At each round d of the game, MINION-sm conducts an exploration step with probability γ or plays an exploitative strategy with probability $1 - \gamma$ (lines 4-13).

In the random exploration phase, we suggest a target-level sampling. In other words, we select target $i \in [LT]$ uniformly random and then we choose one route from a set of crossing routes at target i by solving two instances of mathematical program 2, $[\mathcal{P}(a = st, b = i), \mathcal{P}(a = i, b = ds)]$ in linear time (lines 5-8). The mathematical program $\mathcal{P}(a, b)$ in equation 2 gives the optimal path for the time-unrolled graph shown in Figure 2(b), from the starting node a to the destination node b (see the third constraint for the starting and the destination nodes). In our patrol route planning problem, st and ds denote the patrol post locations at the beginning and end of the patrol route. The weights in this graph are the estimated reward values. We add a random noise vector z to prevent the algorithm to choose a fixed route for all the times that a specific node j is selected in exploration phase (line 6). The mathematical program $\mathcal{P}(a, b)$ is equivalent to the problem of finding the longest path in a weighted directed acyclic graph, which can be solved in linear time. E in equation 2 represents the set of the edges in the time-unrolled graph $G(u, e)$ introduced in section 3. $\sigma^+(v_{d,i})$ denotes the in-going edges to the node $v_{d,i}$ and $\sigma^-(v_{d,i})$ denotes the out-going edges from the node $v_{d,i}$, in graph G. To find the longest path (the optimal defender strategy), we used a network flow approach. Thus, $f(e)$ represents the flow on each edge of the graph G. If a node is covered by defender, $f(e)$ will be 1 for one of the in-going edges and one of the out-going edges.

$$v_d = \arg\max_{v \in \mathcal{V}} \sum_{i=1}^{LT} v_{d,i}(\alpha \tilde{r}_{d,i} + z)$$

$$\begin{array}{ll}
\text{subject to} & \\
v_{d,i} = \sum_{e \in \sigma^+(v_{d,i})} f(e) & \forall e \in E, \forall i \in [LT] \\
\sum_{e \in \sigma^+(v_{d,i})} f(e) = \sum_{e \in \sigma^-(v_{d,i})} f(e) & \forall e \in E, \forall i \in [LT] \\
\sum_{e \in \sigma^-(v_{1,a})} f(e) = \sum_{e \in \sigma^+(v_{D,b})} f(e) = 1 & \forall e \in E \\
f(e), v_{d,i} \in \{0,1\} & \forall i \in [LT], \forall e \in E
\end{array} \tag{2}$$

In our game, a pure strategy is defined as a feasible patrol route (i.e., a route in graph G) and the set of all possible strategies (all routes) are $O(L^T)$. Such set is computationally expensive to be generated for large-size graphs. Additionally, even if we generate such a large set for the exploration step, the algorithm would suffer from a slower convergence. So our target-level random sampling does not require generation of $O(L^T)$ routes and assures that each target i is covered by $p_i \geq \frac{\gamma}{LT}$, as opposed to the strategy-level uniform sampling which assures $p_i \geq \frac{\gamma}{L^T}$. Hence, this approach facilitates scalability of the algorithm and demonstrates similar performance guarantee as FPL-UE without scheduling constraints.

In the exploitation phase, we choose an optimized patrol route computed by mathematical program 2 according to the current estimation of the rewards on all targets up to the current round (lines 10-12).

Once the defender strategy v_d is computed and deployed at round d, reward $r_{d,i}$ is observed for the targets visited by the defender (line 14). Then the probability $p_{d,i}$ that target i is chosen at round d by our algorithm is computed based on the algorithm 2 (line 15) and the reward estimations are adjusted and updated for visited targets as $\tilde{r}_{d+1,i} = \tilde{r}_{d,i} + \frac{r_{d,i}}{p_{d,i}} \mathbb{I}_{d,i}$ (line 16). $\mathbb{I}_{d,i}$ is the indicator function that indicates whether target i was chosen by the defender at round d. The term $\frac{r_{d,i}}{p_{d,i}} \mathbb{I}_{d,i}$ is an unbiased estimator of $r_{d,i}$ (i.e., $\mathbb{E}(\frac{r_{d,i}}{p_{d,i}}) = r_{d,i}$). This choice of the reward adjustment is for convenience of theoretical analysis. Since $p_{d,i}$ cannot be computed efficiently, we use the Geometric Resampling technique proposed by [25], outlined in Algorithm 2, where $K_{d,i} = \frac{1}{p_{d,i}}$ denotes the mean of the geometric distribution with success probability of $p_{d,i}$ for the first trial. W denotes number of the iteration that the algorithm 2 is run and is an input to the algorithm. The MINION-sm algorithm continues for D rounds.

Theorem 4.1. *The performance of MINION-sm follows the same theoretical properties as FPL-UE where the regret (i.e., the difference between the performance of MINION-sm and that of the best fixed patrolling strategy in hindsight) is upper bounded by:*

$$R_D \quad \leq \quad \gamma MD + 2DTe^{-W\frac{\gamma}{LT}} + \frac{T(\log LT + 1)}{\eta} + \eta MD \min(M,T)$$

By taking $\eta = \sqrt{\frac{T(\log LT+1)}{MD\min\{M,T\}}}$, $\gamma = \frac{\sqrt{T}}{\sqrt{MD}}$, $W = L\sqrt{TMD}\log(DT)$, we obtain the upper bound $O\left(\sqrt{TMD\min\{M,T\}\log LT}\right)$.

Due to space limitations, the full proof of this theorem is omitted. However, it can be sketched as follows:

Proof sketch. A key step in the proof of is to bound below the probability that the chosen path will contain a particular node. By construction of MINION-sm, this value can be bounded below with γ/LT. By combining this bound with some ideas from the proof of Theorem 1 from [32] (tailored to our setting) and some further technical algebra, we can achieve the required regret bound. □

Algorithm 2: The GR Algorithm

```
input  : $\eta \in \mathbb{R}^+, W \in \mathbb{Z}^+, \tilde{r} \in \mathbb{R}^{LT}, d \in \mathbb{N}$
output : $K_d \in \mathbb{Z}^{LT}$
Initialize $\forall i \in [LT] : K_{d,i} = 0, k = 1$;
for $k = 1, \ldots, W$ do
    Execute steps $3 - 13$ in Algorithm 1 once just to
      produce $\tilde{v}$ as a simulation of $v_d$;
    for all $i \in [LT]$ do
        if $k < W$ and $\tilde{v}_i = 1$ and $K_{d,i} = 0$ then
            $K_{d,i} = k$
        else if $k = W$ and $K_{d,i} = 0$ then
            $K_{d,i} = W$
    end
    if $K_{d,i} > 0$ for all $i \in [LT]$ then
        break
end
```

4.2 Expert II: Patrol planning via machine learning model

In green security games, the wildlife crime datasets are used for development of explicit attackers' model based on machine learning techniques. Since the ML modeling based on the real-world data is not the focus of this paper, we skip the modeling details and we just briefly provide an overview of the inputs/outputs for such ML models and then we show how the outputs of such ML models are used for patrol planning purposes [10, 11].

ML Model Inputs In wildlife protection domain, the park rangers begin to conduct patrols from patrol posts located across the vast national parks and return to the same patrol posts every day as shown in Figure 2(a). So the wildlife crime datasets consist of several years of type, location, and date of the wildlife crime records detected by park rangers during the repeated patrols which is used for supervised ML modeling of attackers' behavior. Along with these historical observations, several environmental features such as terrain (e.g., slope), distance values (e.g., distance to the border, patrol posts, roads, and towns, rivers), and animal density along with past patrol coverage are considered as predictor features that influence the decision making process by adversaries. Such historical records are transformed into spatio-temporal data points to train a machine learning model as follows. The protected area is divided into grid cells $\tilde{l}$ (e.g., cells of size 1 sq. km) and the entire time span of the crime records, $\tilde{T}$, is divided into small time steps $\tilde{t}$ (e.g., 3 month or 12 month long due to sparsity of the data). Thus the dataset $\mathcal{D} = (\mathbf{X}, \mathbf{y})$, contains $\tilde{T}\tilde{L}$ of such spatio-temporal slices (usually tens of thousands) from all around the park over several years where $\mathbf{X} \in \mathbb{R}^{\tilde{T}\tilde{L}\times f}$ is a matrix of f predictor features and $\mathbf{y} \in \{0,1\}^{\tilde{T}\tilde{L}}$ denotes the observation vector which represents the presence or absence of the attack.

ML Model Outputs Training a machine learning model based on $\mathcal{D} = (\mathbf{X}, \mathbf{y})$ gives predictions about probability scores (i.e., attack risk) $p(i) = h(\mathbf{x}_i)$ at each target i. Such predictions are used to generate optimized patrol strategies as shown by the following mathematical model $Q(a, b)$, where a and b are starting and ending targets for patrolling.

$$v_d = \arg\max_{v\in\mathcal{V}} \sum_{i=1}^{LT} v_{d,i}.p_i$$

$$\begin{array}{ll}
\text{subject to} & \\
v_{d,i} = \sum_{e\in\sigma^+(v_{d,i})} f(e) & \forall e \in E, \forall i \in [LT] \\
\sum_{e\in\sigma^+(v_{d,i})} f(e) = \sum_{e\in\sigma^-(v_{d,i})} f(e) & \forall e \in E, \forall i \in [LT] \\
\sum_{e\in\sigma^-(v_{1,a})} f(e) = \sum_{e\in\sigma^+(v_{D,b})} f(e) = 1 & \forall e \in E \\
f(e), v_{d,i} \in \{0,1\} & \forall i \in [LT], \forall e \in E
\end{array} \qquad (3)$$

Due to the sparsity of the datasets, $\tilde{t}$, the smallest time resolution for ML model predictions is much larger than the smallest time horizon T required for fine-tuned patrol planning, i.e., $\tilde{t} \gg T$. Consequently, machine learning predictions for each location does not get updated real-time and remain nearly similar across time period T (i.e., stationary predictions) shown in time-unrolled graph in Figure 2(b).

4.3 Patrol planning via expert I and II

Algorithm 3: The MINION Algorithm

```
   parameters: $\eta \in \mathbb{R}^+, \beta \in \mathbb{R}^+, W \in \mathbb{Z}^+, \gamma \in [0,1], e \in \mathbb{N}$,
               $st \in [LT]$ , $ds \in [LT]$;
1  Initialize the estimated reward $\tilde{r}_d = 0 \in \mathbb{R}^{LT}$, $r_{ml} = 0$, $r_{ol} = 0$,
     $n_{ml} = 0$, $n_{ol} = 0$;
2  Pick $e$ exploration strategies such that two experts ml
     (outlined in line 7) and ol (outlined in lines 10-16) are explored
     uniformly and $r_{ml}$ and $r_{ol}$ are initialized ;
3  for $d = 1, \ldots, D$ do
4      Draw $c_{d,1} \sim exp(\beta)$ and $c_{d,2} \sim exp(\beta)$
5      if $\frac{r_{ml}}{n_{ml}} + c_{d,1} \geq \frac{r_{ol}}{n_{ol}} + c_{d,2}$ then
6          $n_{ml} \leftarrow n_{ml} + 1$;
7          $f = 0$;
8          Let $v_d$ be computed from the mathematical program
             $Q(a = st, b = ds)$ in 3;
9      else
10         $n_{ol} \leftarrow n_{ol} + 1$;
11         $f = 1$;
12         Let $v_d$ be computed by following steps 3-13 in
             algorithm 1;
13     end
14     Adversary picks $r_{d,i} \in [0,1]^{LT}$ and defender plays $v_d$;
15     $r_{ml} \leftarrow r_{ml} + f v_d r_d$;
16     $r_{ol} \leftarrow r_{ol} + f v_d r_d$;
17     Run $GR(\eta, w, \tilde{r}, d)$: estimate $\frac{1}{p_{d,i}}$ as $K_{d,i}$;
18     Update $\tilde{r}_{d,i} \leftarrow \tilde{r}_{d,i} + K_{d,i} r_{d,i} \mathbb{I}_{d,i}$; where $\mathbb{I}_{d,i} = 1$ for
         $v_{d,i} = 1$; $\mathbb{I}_{d,i} = 0$ otherwise;
19 end
```

Algorithm 3 outlines our meta-learning approach to balance between two experts, i.e., (I) MINION-sm online learning algorithm with no prior knowledge and (II) an ML-based patrol planning model with potentially imperfect prior knowledge. This algorithm initializes the estimation of the reward as 0 (line 1) and then picks a set of exploration strategies to obtain an initial assessment about the performance of the experts; thus r_{ml} and r_{ol} are initialized for both patrol planning experts (line 2). At each round d of the game, the current collected rewards for each expert are perturbed (line 4) by drawing random noise for each expert from the exponential distribution with parameter β to model the noise on the current estimation of the rewards and then the best expert is chosen by the algorithm (line 5). If ML model is selected as the best expert, v_d is computed based on the mathematical program $Q(a = st, b = ds)$ presented by equations 3 (lines 6-8). Otherwise, the MINION-sm online learning approach is used (lines 10-22). Then the adversary picks the rewards r_d for the defender (line 24) and the collected rewards for each expert will be updated accordingly (lines 25-28). The MINION algorithm continues for D rounds.

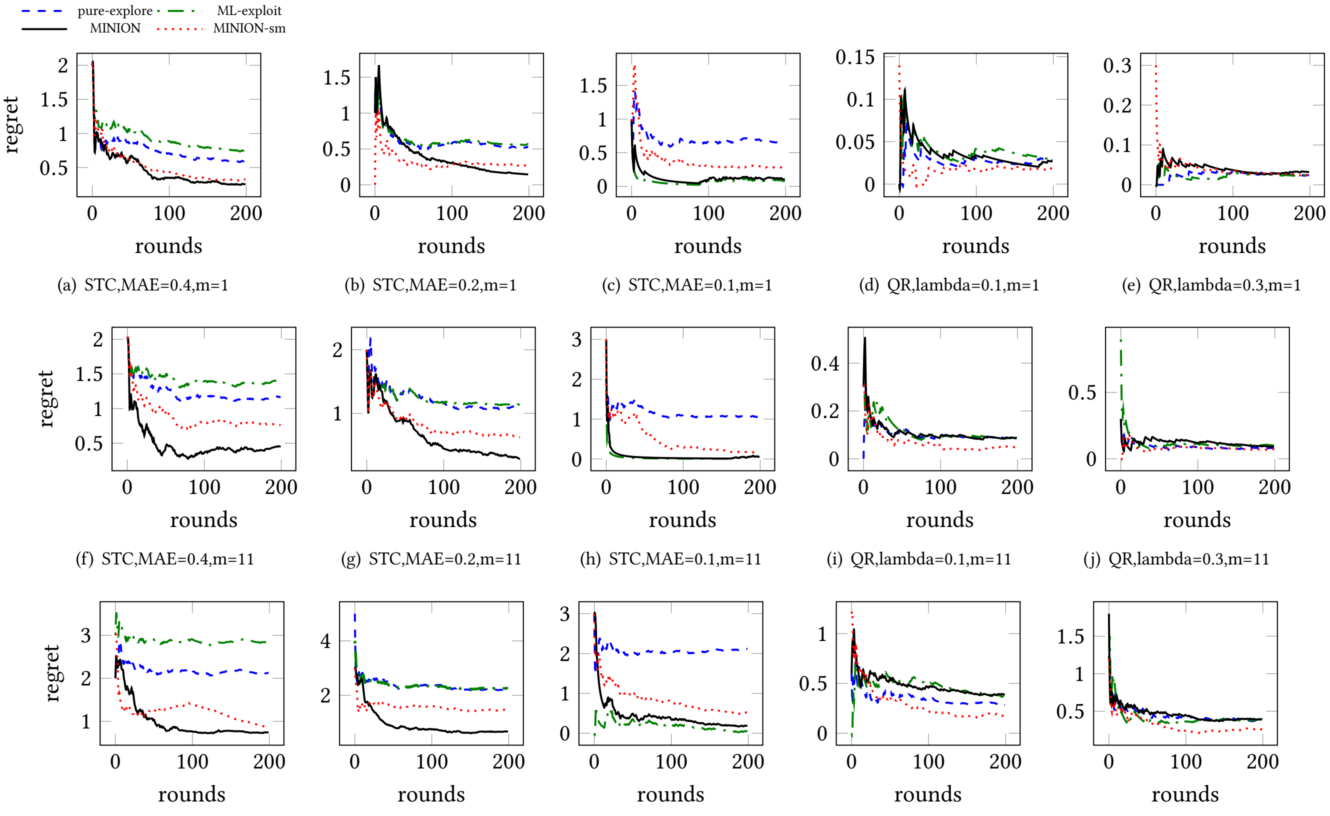
The intuition behind MINION is that the algorithm will learn whether it is useful to rely on historical data. If yes, then it will use the ML model to predict the future payoffs, otherwise it will use MINION-sm to plan the patrolling strategy. In particular, we provide the following guarantee on the performance of MINION:

Theorem 4.2. *Let P_{ML} and P_{fixed} denote the expected performance of the ML model and the best fixed patrolling strategy in hindsight. The expected performance of MINION is at least as good as* $\max\{P_{\text{ML}}, P_{\text{fixed}}\} - O\left(\sqrt{TMD \min\{M,T\} \log LT}\right)$.

Proof sketch. If $P_{\text{ML}} > P_{\text{fixed}}$ then the meta-learner in MINION will learn this with $O(\sqrt{D})$ regret (as the meta-learner a two-expert learning problem). Otherwise, it will converge to MINION-sm. This yields regret of $O(\sqrt{D})$ +
$O\left(\sqrt{TMD \min\{M,T\} \log LT}\right) = O\left(\sqrt{TMD \min\{M,T\} \log LT}\right)$. □

5 NUMERICAL EVALUATION

In this section, we evaluate the numerical performance of the MINION-sm and MINION against an ML-based patrol planning model (ML-exploit) and absolute exploratory defender strategies (pure-explore). We first evaluate our algorithms on a game with 25 locations ($L = 25$) and a patrol horizon of 6 time steps ($T = 6$) and then we show the average defender reward for all techniques by varying the patrol horizon (i.e., different time-unrolled graph sizes). The MINION-sm and pure-explore algorithm do not incorporate any explicit model for attackers' behavior. However, the MINION algorithm and also ML-exploit baseline algorithm require access



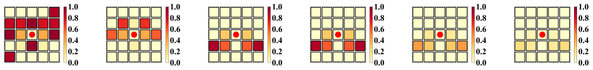
(a) STC,MAE=0.4,m=1 (b) STC,MAE=0.2,m=1 (c) STC,MAE=0.1,m=1 (d) QR,lambda=0.1,m=1 (e) QR,lambda=0.3,m=1

(f) STC,MAE=0.4,m=11 (g) STC,MAE=0.2,m=11 (h) STC,MAE=0.1,m=11 (i) QR,lambda=0.1,m=11 (j) QR,lambda=0.3,m=11

(k) STC,MAE=0.4,m=22 (l) STC,MAE=0.2,m=22 (m) STC,MAE=0.1,m=22 (n) QR,lambda=0.1,m=22 (o) QR,lambda=0.3,m=22

Figure 3: Regret for adversaries with stochastic (stationary) and Quantal response (non-stationary) behavior - $L = 25$, $T = 6$

to an ML model for attackers' behavior to solve the mathematical model 3 for patrol planning. We simulate the ML model predictions (ML outputs), $p(i) = h(\mathbf{x}_i)$, with three different scenarios shown in Figures 4(a) to 4(c). These predictions are stationary for all locations across the patrol horizon. We assume two types of adversarial behavior: (i) STC- a Stochastic adversarial behavior where the likelihood of attack at each location can be defined by probability scores; Figures 4(d) to 4(f) shows our simulated cases for three different m values, where m indicates the expected number of the attackers. These probability scores represent the ground truth for adversarial behavior and are stationary across $T = 6$ time steps for all locations. We used them to pick rewards for the defender play in the game for all of the patrol planning methods. (ii) QR- a Quantal Response adversary where the attackers' behavior is non-stationary across the game rounds and the attackers respond to the empirical defender mixed strategy by a QR model [27].

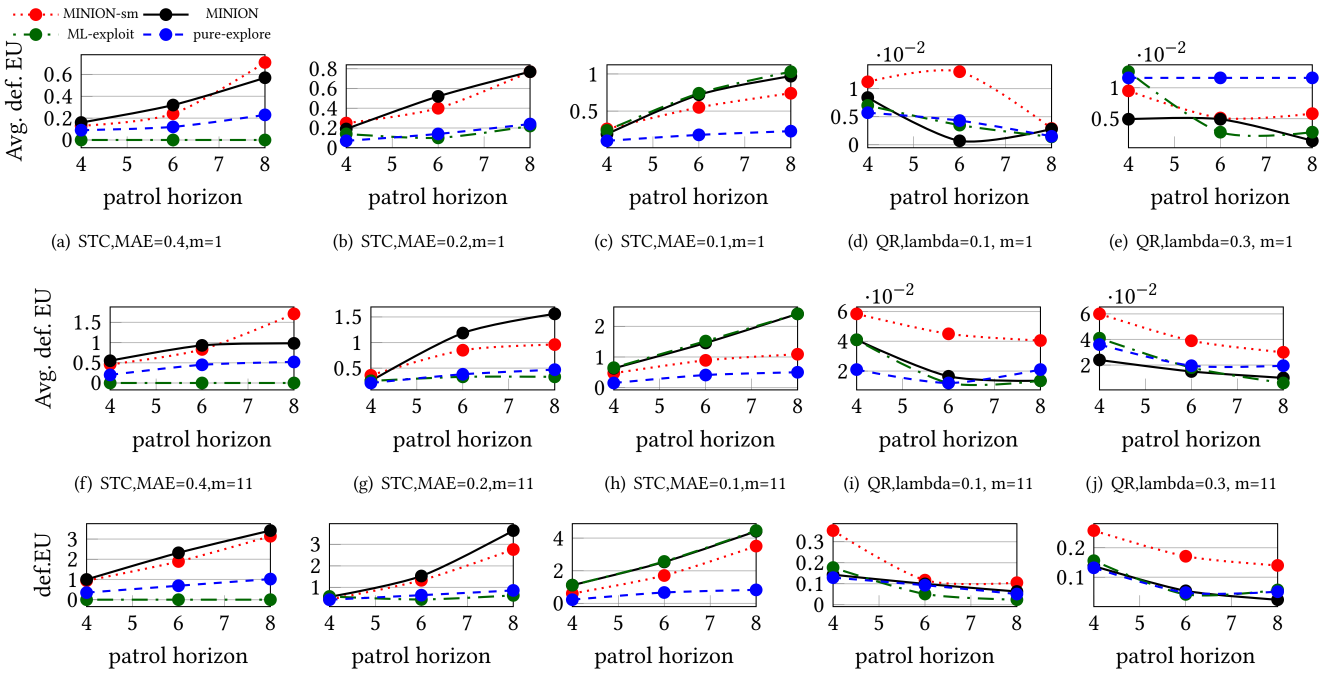


(a) MAE=0.4 (b) MAE=0.2 (c) MAE=0.1 (d) GT,m=22 (e) GT,m=11 (f) GT,m=1

Figure 4: left three heat maps for attack probability predicted by different ML models, right three heat maps for attack probability ground truth, red dot is patrol post location

For each STC adversary represented by the ground truth (GT) probabilities shown in Figure 4(d) to 4(f), the ML simulations have different levels of inaccuracy. We quantify this difference via $MAE = \sum_{l=1}^{L} |p_{gt}(l) - p_{ml}(l)|/L$ which is the mean absolute error in predictions. In our simulated cases shown in Figure 4, MAE varies from 0.1 to 0.4. For QR adversaries, we do not have a fixed GT and adversaries' responses are updated according to the updated mixed strategy of the defender. We examine two λ values, i.e., 0.1 and 0.3 as the rationality parameters of the adversaries where the smaller values indicate more non-rational adversaries in QR model.

The regret values for all 9 scenarios for STC and 6 scenarios for QR are shown in Figures 3. The blue dashed lines and the green lines in the figures show the results for pure-explore and for ML-exploit baseline methods, respectively. The red dotted lines and black solid lines illustrate the results for MINION-sm and MINION algorithms proposed in this study. The regret values are shown along the y-axis and the game rounds are shown along the x-axis. In the left three columns, for the STC adversary, we show the performance loss when the accuracy of the ML model predictions (used



(a) STC,MAE=0.4,m=1 (b) STC,MAE=0.2,m=1 (c) STC,MAE=0.1,m=1 (d) QR,lambda=0.1, m=1 (e) QR,lambda=0.3, m=1

(f) STC,MAE=0.4,m=11 (g) STC,MAE=0.2,m=11 (h) STC,MAE=0.1,m=11 (i) QR,lambda=0.1, m=11 (j) QR,lambda=0.3, m=11

(k) STC,MAE=0.4,m=22 (l) STC,MAE=0.2,m=22 (m) STC,MAE=0.1, m=22 (n) QR,lambda=0.1, m=22 (o) QR,lambda=0.3, m=22

Figure 5: Average defender utility over 200 rounds for adversaries with stochastic (stationary) and Quantal response (non-stationary) behavior for different graph sizes and different number of attackers m

in ML-exploit and MINION) increases from left to the right. From top to the bottom, we show the change in performance loss as the expected number of the adversary increases. In the first and the second column in Figure 3 where ML model is relatively inaccurate, MINION-sm (the technique that uses no prior knowledge and balances exploration and exploitation), outperforms ML-exploit whereas in the third column the trend is reversed since the ML model is sufficiently accurate and informative for the patrol planning task. On the other hand, MINION outperforms all other methods in all cases since it obtains a balance between MINION-sm and ML-exploit and finds the best expert based on their empirical performance. In the right two columns, for QR adversaries, MINION-sm algorithm outperforms other techniques. When the relative number of the adversaries to the number of the defender resources is larger, this difference is more significant. The MINION is outperformed by MINION-sm against QR adversary, since it partially relies on an ML-based patrol planning expert for which the predictions are not updated accordingly over the game rounds and thus suffers from biases in prior knowledge.

Figure 5 shows the average defender utility over 200 rounds of the game on the y-axis vs. different patrol horizons. The game settings with different number of attackers are outlined across the different rows. For STC adversary (shown in the left three columns), MINION outperforms all other methods and for QR adversaries (shown in the right two columns), the MINION-sm algorithm outperforms other methods for all graph sizes. The key reason behind the poor performance of MINION vs. MINION-sm in QR scenario is that MINION incorporates an ML-based planner with stationary predictions about the attackers' behavior as an expert planner against the non-stationary (responsive and strategic) adversaries which is detrimental to the defender.

6 CONCLUSION

This paper focuses on the important problem of game-theoretic patrol route selection for preventing poaching activities in wildlife parks. The main intellectual contribution of the paper is that it shows that over-reliance on historical patrolling data (or "prior knowledge") in the patrol route generation process may lead to highly sub-optimal patrols, and that the optimal amount of reliance on prior knowledge can be learned effectively (by techniques put forth in the paper). Specifically, this paper makes the following methodological contributions: (I) we propose MINION-sm, a scalable online learning algorithm that learns an implicit model of the attacker when defender is spatio-temporally constrained, (II) we propose MINION, which is a scalable multi-expert patrol planning algorithm with spatio-temporal constraints for the defender that obtains a balance between the ML-based planners and MINION-sm based on their empirical performance. We showed that our algorithms outperformed other techniques in different game settings.

Acknowledgment: This research was supported by NSF grant with Cornell University 72954-10598 and MURI W911NF1810208.

REFERENCES

[1] The Atlantic. 2014. UN Warns That Growing $213 Billion Poaching Industry Funds Armed Conflicts. https://www.theatlantic.com/international/archive/2014/06/un-warns-that-growing-213-billion-poaching-industry-funds-armed-conflicts/373324/. (2014).

[2] Maria-Florina Balcan, Avrim Blum, Nika Haghtalab, and Ariel D Procaccia. 2015. Commitment without regrets: Online learning in stackelberg security games. In *Proceedings of the sixteenth ACM conference on economics and computation*. ACM, 61–78.

[3] Nicola Basilico and Nicola Gatti. 2014. Strategic guard placement for optimal response toalarms in security games. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1481–1482.

[4] Avrim Blum, Nika Haghtalab, and Ariel D Procaccia. 2014. Learning optimal commitment to overcome insecurity. In *Advances in Neural Information Processing Systems*. 1826–1834.

[5] Avrim Blum and Yishay Monsour. 2007. Learning, regret minimization, and equilibria. (2007).

[6] Guillaume Chapron, Dale G Miquelle, Amaury Lambert, John M Goodrich, Stéphane Legendre, and Jean Clobert. 2008. The impact on tigers of poaching versus prey depletion. *Journal of Applied Ecology* 45, 6 (2008), 1667–1674.

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

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

[9] International Union for Conservation of Nature. 2015. IUCN red list of threatened species. https://www.iucnredlist.org/. (2015).

[10] Shahrzad Gholami, Benjamin Ford, Fei Fang, Andrew Plumptre, Milind Tambe, Margaret Driciru, Fred Wanyama, Aggrey Rwetsiba, Mustapha Nsubaga, and Joshua Mabonga. 2017. Taking it for a test drive: a hybrid spatio-temporal model for wildlife poaching prediction evaluated through a controlled field test. In *Proceedings of the European Conference on Machine Learning & Principles and Practice of Knowledge Discovery in Databases, ECML PKDD*.

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

[12] Shahrzad Gholami, Bryan Wilder, Matthew Brown, Dana Thomas, Nicole Sintov, and Milind Tambe. 2016. Divide to Defend: Collusive Security Games. In *GameSec*. Springer, 272–293.

[13] Nika Haghtalab, Fei Fang, Thanh Hong Nguyen, Arunesh Sinha, Ariel D Procaccia, and Milind Tambe. 2016. Three Strategies to Success: Learning Adversary Models in Security Games.. In *IJCAI*, Vol. 16. 308–314.

[14] Public Radio International. 2013. Rangers in Kenya are outgunned in the new poaching arms race. https://www.pri.org/stories/2013-10-08/rangers-kenya-are-outgunned-new-poaching-arms-race. (Oct. 2013).

[15] Debarun Kar, Fei Fang, Francesco Delle Fave, Nicole Sintov, and Milind Tambe. 2015. A game of thrones: when human behavior models compete in repeated Stackelberg security games. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1381–1390.

[16] Debarun Kar, Benjamin Ford, Shahrzad Gholami, Fei Fang, Andrew Plumptre, Milind Tambe, Margaret Driciru, Fred Wanyama, Aggrey Rwetsiba, Mustapha Nsubaga, et al. 2017. Cloudy with a Chance of Poaching: Adversary Behavior Modeling and Forecasting with Real-World Poaching Data. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 159–167.

[17] Christopher Kiekintveld, Towhidul Islam, and Vladik Kreinovich. 2013. Security games with interval uncertainty. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 231–238.

[18] Richard Klíma, Christopher Kiekintveld, and Viliam Lisỳ. 2014. Online learning methods for border patrol resource allocation. In *International Conference on Decision and Game Theory for Security*. Springer, 340–349.

[19] Richard Klíma, Viliam Lisỳ, and Christopher Kiekintveld. 2015. Combining online learning and equilibrium computation in security games. In *International Conference on Decision and Game Theory for Security*. Springer, 130–149.

[20] Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. 2011. Solving Stackelberg games with uncertain observability. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*. International Foundation for Autonomous Agents and Multiagent Systems, 1013–1020.

[21] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Eric Horvitz. 2017. Identifying Unknown Unknowns in the Open World: Representations and Policies for Guided Exploration.. In *AAAI*, Vol. 1. 2.

[22] Andrew M Lemieux. 2014. *Situational prevention of poaching*. Routledge.

[23] Jennifer F Moore, Felix Mulindahabi, Michel K Masozera, James D Nichols, James E Hines, Ezechiel Turikunkiko, and Madan K Oli. 2018. Are ranger patrols effective in reducing poaching-related threats within protected areas? *Journal of Applied Ecology* 55, 1 (2018), 99–107.

[24] Enrique Munoz de Cote, Ruben Stranders, Nicola Basilico, Nicola Gatti, and Nick Jennings. 2013. Introducing alarms in adversarial patrolling games. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 1275–1276.

[25] Gergely Neu and Gábor Bartók. 2013. An efficient algorithm for learning with semi-bandit feedback. In *International Conference on Algorithmic Learning Theory*. Springer, 234–248.

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

[27] Thanh Hong Nguyen, Rong Yang, Amos Azaria, Sarit Kraus, and Milind Tambe. 2013. Analyzing the Effectiveness of Adversary Modeling in Security Games.. In *AAAI*.

[28] Eric W Sanderson, Jessica Forrest, Colby Loucks, Joshua Ginsberg, Eric Dinerstein, John Seidensticker, Peter Leimgruber, Melissa Songer, Andrea Heydlauff, Timothy OâĂŹBrien, et al. 2010. Setting priorities for tiger conservation: 2005–2015. In *Tigers of the World (Second Edition)*. Elsevier, 143–161.

[29] Arunesh Sinha, Debarun Kar, and Milind Tambe. 2016. Learning adversary behavior in security games: A PAC model perspective. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 214–222.

[30] Milind Tambe. 2011. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge University Press.

[31] Haifeng Xu, Benjamin Ford, Fei Fang, Bistra Dilkina, Andrew Plumptre, Milind Tambe, Margaret Driciru, Fred Wanyama, Aggrey Rwetsiba, Mustapha Nsubaga, et al. 2017. Optimal Patrol Planning for Green Security Games with Black-Box Attackers. In *International Conference on Decision and Game Theory for Security*. Springer, 458–477.

[32] Haifeng Xu, Long Tran-Thanh, and Nicholas R Jennings. 2016. Playing repeated security games with no prior knowledge. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 104–112.

[33] Rong Yang, Christopher Kiekintveld, Fernando Ordonez, Milind Tambe, and Richard John. 2011. Improving resource allocation strategy against human adversaries in security games. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, Vol. 22. 458.