CAPTURE: A New Predictive Anti-Poaching Tool for Wildlife Protection

Thanh H. Nguyen[1], Arunesh Sinha[1], Shahrzad Gholami[1], Andrew Plumptre[2], Lucas Joppa[3], Milind Tambe[1], Margaret Driciru[4], Fred Wanyama[4], Aggrey Rwetsiba[4], Rob Critchlow[5], Colin M. Beale[5]
[1]University of Southern California, USA, {thanhhng,aruneshs,sgholami,tambe}@usc.edu
[2]Wildlife Conservation Society, USA, aplumptre@wcs.org
[3]Microsoft Research, USA, lujoppa@microsoft.com
[4]Uganda Wildlife Authority, Uganda,
{margaret.driciru,fred.wanyama,aggrey.rwetsiba}@ugandawildlife.org
[5]The University of York, UK, {rob.critchlow,colin.beale}@york.ac.uk

ABSTRACT

Wildlife poaching presents a serious extinction threat to many animal species. Agencies ("defenders") focused on protecting such animals need tools that help analyze, model and predict poacher activities, so they can more effectively combat such poaching; such tools could also assist in planning effective defender patrols, building on the previous security games research.

To that end, we have built a new predictive anti-poaching tool, CAPTURE (Comprehensive Anti-Poaching tool with Temporal and observation Uncertainty REasoning). CAPTURE provides four main contributions. First, CAPTURE's modeling of poachers provides significant advances over previous models from behavioral game theory and conservation biology. This accounts for: (i) the defender's imperfect detection of poaching signs; (ii) complex temporal dependencies in the poacher's behaviors; (iii) lack of knowledge of numbers of poachers. Second, we provide two new heuristics: parameter separation and target abstraction to reduce the computational complexity in learning the poacher models. Third, we present a new game-theoretic algorithm for computing the defender's optimal patrolling given the complex poacher model. Finally, we present detailed models and analysis of real-world poaching data collected over 12 years in Queen Elizabeth National Park in Uganda to evaluate our new model's prediction accuracy. This paper thus presents the largest dataset of real-world defender-adversary interactions analyzed in the security games literature. CAPTURE will be tested in Uganda in early 2016.

Keywords

Security Game; Wildlife Protection; Temporal Behavorial Model

1. INTRODUCTION

Wildlife protection is a global concern. Many species such as tigers and rhinos are in danger of extinction as a direct result of illegal harvesting (i.e., poaching) [19, 26]. The removal of these and other species from the landscape threatens the functioning of natural ecosystems, hurts local and national economies, and has become an international security concern due to the unregulated profits of poachers flowing to terrorist organizations [24]. To prevent wildlife poaching, conservation organizations attempt to protect wildlife parks with well-trained park rangers. In each time period (e.g., one month), park rangers conduct patrols within the park area to prevent poachers from capturing animals either by catching the poachers or by removing animals traps laid out by the poachers. During the rangers' patrols, poaching signs are collected and then can be used together with other domain features (e.g., animal density) to predict the poachers' behavior [6, 8]. In essence, learning the poachers' behavior, anticipating where poachers often go for poaching, is critical for the rangers to generate effective patrols.

Motivated by the success of defender-attacker Stackelberg Security Game (SSG) applications for infrastructure security problems [28, 3, 14], previous work has began to apply SSGs for wildlife protection [30, 9, 8]. In particular, an SSG-based patrolling decision-aid called PAWS has been deployed in south-east Asia [8]. PAWS focuses on generating effective patrols for the rangers, taking into account the complex topographic conditions of Asian forests. Despite its successful application, PAWS is known to suffer from several limitations. First, PAWS relies on an existing adversary behavior model known as Subjective Utility Quantal Response (SUQR) [8], which makes several limiting assumptions such as (a) all poaching signs are perfectly observable by the rangers; (b) poachers' activities in one time period are independent of their activities in previous or future time periods; (c) the number of poachers is known. As a result, SUQR's modeling falls short of what is required, as security agencies in some countries are interested in detailed analysis, modeling and prediction of poacher behavior, taking into account all of the detailed domain features. That is they may wish to obtain such information for situational awareness of the area under their protection and for other strategic decisions. Second, since SUQR has traditionally only relied on three or four domain attributes in its modeling, it has not been able to provide a detailed analysis of the impact of environmental and terrain features on poacher behavior, and thus such analysis of real-world data has been lacking in the literature. Third, richer adversary models would also require new patrol generation algorithms that improve upon what is used in PAWS.

In essence, our new CAPTURE tool attempts to address all aforementioned limitations in PAWS while providing the following three key contributions. Our first area of contribution relates to CAPTURE's addressing SUQR's limitations in modeling ad-

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), John Thangarajah, Karl Tuyls, Stacy Marsella, Catholijn Jonker (eds.), May 9–13, 2016, Singapore.*

versary behavior. More specifically, CAPTURE introduces a new behavioral model which takes into account the rangers' imperfect detection of poaching signs. Additionally, we incorporate the dependence of the poachers' behavior on their activities in the past into the component for predicting the poachers' behavior. Moreover, we adopt logistic models to formulate the two components of the new model. This enables capturing the aggregate behavior of attackers without requiring a known number of poachers. Finally, CAPTURE considers a richer set of domain features in addition to the three/four features used in SUQR in analyzing the poachers' behavior. Second, we provide two new heuristics to reduce the computational cost of learning adversary models in CAPTURE, namely *parameter separation* and *target abstraction*. The first heuristic divides the set of model parameters into separate subsets and then iteratively learns these subsets of parameters separately while fixing the values of the other subsets. This heuristic decomposes the learning process into less complex learning components which help in speeding up the learning process with no loss in accuracy. The second heuristic of target abstraction works by leveraging the continuous spatial structure of the wildlife domain, starting the learning process with a coarse discretization of forest area and gradually using finer discretization instead of directly starting with the most detailed representation, leading to improved runtime overall. Our third contribution lies in computing the optimal patrolling strategy of the rangers given the new behavioral model. Specifically, we provide a new game-theoretic algorithm for single/multiple-step patrolling plans wherein the poachers' actions (which follow the CAPTURE model) are recursively explored in multiple time steps.

Finally, we extensively evaluate the prediction accuracy of our new CAPTURE model based on a detailed analysis of the largest dataset of real-world defender-adversary interactions collected by rangers in Queen Elizabeth National Park (QENP) over 12 years. In fact, this is the largest such study in the security games literature. The experimental results show that our model is superior to existing models in predicting the poachers' behaviors, demonstrating the advances of our model over the previous state-of-the-art models. To that end, CAPTURE will be tested in Uganda in early 2016.

2. BACKGROUND & RELATED WORK

Stackelberg Security Games. In Stackelberg security games, there is a defender who attempts to optimally allocate her limited security resources to protect a set of targets against an adversary attempting to attack one of the targets [28]. In SSGs, the defender commits to a *mixed* strategy first while the attacker can observe the defender's strategy and then take an action based on that observation. A pure strategy of the defender is an assignment of her limited resources to a subset of targets and a mixed strategy of the defender refers to a probability distribution over all possible pure strategies. The defender's mixed strategies can be represented as a marginal coverage vector over the targets (i.e., the coverage probabilities with which the defender will protect each target) [13]. We denote by N the number of targets and $0 \leq c_i \leq 1$ the defender's coverage probability at target i for $i = 1 \ldots N$. If the attacker attacks target i and the defender is not protecting that target, the attacker obtains a reward R_i^a while the defender gets a penalty P_i^d. Conversely, if the target is protected, the attacker receives a penalty P_i^a while the defender achieves a reward R_i^d. The expected utilities of the defender, U_i^d, and attacker, U_i^a, are computed as follows:

$$U_i^d = c_i R_i^d + (1 - c_i) P_i^d \tag{1}$$

$$U_i^a = c_i P_i^a + (1 - c_i) R_i^a \tag{2}$$

Behavioral Models of Adversaries. In SSGs, different behavioral models have been proposed to capture the attacker's behavior. The Quantal Response model (QR) is one of the most popular behavioral models which attempts to predict a stochastic distribution of the attacker's responses [16, 17]. In general, QR predicts the probability that the attacker will choose to attack each target with the intuition that the higher expected utility of a target, the more likely that the attacker will choose that target. A more recent model, SUQR, (which is shown to outperform QR) also attempts to predict an attacking distribution over the targets [22]. However, instead of relying on expected utility, SUQR uses the subjective utility function, $\hat{U}_i^a$, which is a linear combination of all features that can influence the attacker's behaviors.

$$\hat{U}_i^a = w_1 c_i + w_2 R_i^a + w_3 P_i^a \tag{3}$$

where (w_1, w_2, w_3) are the key model parameters which measure the importance of the defender's coverage, the attacker's reward and penalty w.r.t the attacker's action. Based on subjective utility, SUQR predicts the attacking probability, q_i, at target i as follows:

$$q_i = \frac{e^{\hat{U}_i^a}}{\sum_j e^{\hat{U}_j^a}} \tag{4}$$

In addition to QR/SUQR, there are other lines of research which focus on building models of criminal behavior in urban crime [7, 20, 23, 32] or opponent behavior in poker [10, 27]. However, these models are specifically designed for these domains, which rely on the complete past crime/game data as well as intrinsic domain characteristics. Another line of research focuses on adversarial plan recognition [1], which can be applied for computer intrusion detection and detection of anomalous activities, etc. This line of work does not learn model parameters as well as do any patrol planning. Here, CAPTURE focuses on modeling the poachers' behavior in wildlife protection which exhibits unique challenges (as shown below) that existing behavioral models cannot handle.

Wildlife Protection. Previous work in security games has modeled the problem of wildlife protection as a SSG in which the rangers play in a role of the defender while the poachers are the attacker [30, 9, 8, 12]. The park area can be divided into a grid where each grid cell represents a target. The rewards and penalties of each target w.r.t the rangers and poachers can be determined based on domain features such as animal density and terrain slope. Previous work focuses on computing the optimal patrolling strategy for the rangers given that poachers' behavior is predicted based on existing adversary behavioral models. However, these models make several limiting assumptions as discussed in Section 1 including (a) all poaching signs (e.g., snares) are perfectly observable by the rangers; (b) poachers' activities in one time period are independent of their activities in previous or future time periods; (c) the number of poachers is known. To understand the limiting nature of these assumptions, consider the issue of observability. The rangers' capability of making observations over a large geographical area is limited. For example, the rangers usually follow certain paths/trails to patrol; they can only observe over the areas around these paths/trails which means that they may not be able to make observations in other further areas. In addition, in areas such as dense forests, it is difficult for the rangers to search for snares. As a result, there may be still poaching activities happening in areas where rangers did not find any poaching sign. Therefore, relying entirely on the rangers' observations would lead to an inaccurate prediction of the poachers' behavior, hindering the rangers' patrol effectiveness. Furthermore, when modeling the poachers' behavior, it is critical to incorporate important aspects that affect the poachers' behavior including time dependency of the poachers' activities and patrolling frequencies of the rangers. Lastly, the rangers are unaware of the total number of attackers in the park.

In ecology research, while previous work mainly focused on estimating the animal density [15], there are a few works which attempt to model the spatial distribution of the economic costs/benefits of illegal hunting activities in the Serengeti national park [11] or the threats to wildlife and how these change over time in QENP [6]. However, these models also have several limitations. First, the proposed models do not consider the time dependency of the poachers' behaviors. These models also do not consider the effect of the rangers' patrols on poaching activities. Furthermore, the prediction accuracy of the proposed models is not measured. Finally, these works do not provide any solution for generating the rangers' patrolling strategies with a behavioral model of the poachers.

3. BEHAVIORAL LEARNING

Security agencies protecting wildlife have a great need for tools that analyze, model and predict behavior of poachers. Such modeling tools help the security agencies gain situational awareness, and decide general strategies; in addition, these agencies also find it useful to have patrol planning tools that are built based on such models. The key here is that in wildlife protection areas around the world, these security agencies have collected large amounts of data related to interactions between defenders (patrollers) and adversaries (poachers). In our work, we focus on QENP [30, 6], where in collaboration with the Wildlife Conservation Society (WCS) and Uganda Wildlife Authority (UWA), we have obtained 12 years of ranger-collected data (that is managed in database MIST/SMART).

In CAPTURE, we introduce a new hierarchical behavioral model to predict the poachers' behavior in the wildlife domain, taking into account the challenge of rangers' imperfect observation. Overall, the new model consists of two layers. One layer models the probability the poachers attack each target wherein the temporal effect on the poachers' behaviors is incorporated. The next layer predicts the conditional probability of the rangers detecting any poaching sign at a target given that the poachers attack that target. These two layers are then integrated to predict the rangers' final observations. In our model, we incorporate the effect of the rangers' patrols on both layers, i.e., how the poachers adapt their behaviors according to rangers' patrols and how the rangers' patrols determine the rangers' detectability of poaching signs. Furthermore, we consider the poachers' past activity in reasoning about future actions of the poachers. We also include different domain features to predict either attacking probabilities or detection probabilities or both.

3.1 Hierarchical Behavioral Model

We denote by T the number of time steps, N the number of targets, and K the number of domain features. At each time step t, each target i is associated with a set of feature values $\mathbf{x_{t,i}} = \{x^k_{t,i}\}$ where $k = 1 \ldots K$ and $x^k_{t,i}$ is the value of the k^{th} feature at (t, i). In addition, $c_{t,i}$ is defined as the coverage probability of the rangers at (t, i). When the rangers patrol target i in time step t, they have observation $o_{t,i}$ which takes an integer value in $\{-1, 0, 1\}$. Specifically, $o_{t,i} = 1$ indicates that the rangers observe a poaching sign at (t, i), $o_{t,i} = 0$ means that the rangers have no observation and $o_{t,i} = -1$ when the rangers did not patrol at (t, i). Furthermore, we define $a_{t,i} \in \{0, 1\}$ as the actual action of poachers at (t, i) which is hidden from the rangers. Specifically, $a_{t,i} = 1$ indicates the poachers attack at (t, i); otherwise, $a_{t,i} = 0$ means the poachers did not attack at (t, i). In this work, we only consider the situation of attacked or not (i.e., $a_{t,i} \in \{0, 1\}$); the case of multiple-level attacks is left for future work. Moreover, we mainly focus on the problem of false negative observations, meaning that there may still exist poaching activity at locations where the rangers found no sign of poaching. We make the reasonable assumption that there is no false positive observation, meaning that if the rangers found any poaching sign at a target, the poachers did attack that target. In other words, we have $p(a_{t,i} = 1|o_{t,i} = 1) = 1$ and $p(o_{t,i} = 1|a_{t,i} = 0) = 0$.

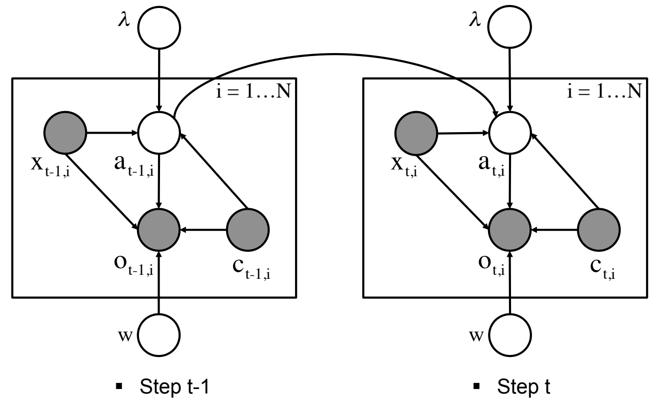


Figure 1: Dependencies among CAPTURE modeling elements

The graphical representation of the new model is shown in Figure 1 wherein the directed edges indicate the dependence between elements of the model. The grey nodes refer to known elements for the rangers such as domain features, the rangers' coverages and observations while the white nodes represent the unknown elements such as the actual actions of poachers. The elements $(\lambda, \mathbf{w})$ are model parameters which we will explain later.

Our new CAPTURE graphical model is a significant advance over previous models from behavioral game theory, such as QR/SUQR, and similarly models from conservation biology [11, 6]. First, unlike SUQR/QR which consider poachers behavior to be independent between different time steps, we assume that the poachers' actions $a_{t,i}$ depends on the poachers' activities in the past $a_{t-1,i}$ and the rangers' patrolling strategies $c_{t,i}$. This is because poachers may tend to come back to the areas they have attacked before. Second, CAPTURE considers a much richer set of domain features $\{x^k_{t,i}\}$ that have not been considered earlier but are relevant to our domain, e.g., slope and habitat. Third, another advance of CAPTURE is modeling the observation uncertainty in this domain. We expect that the rangers' observations $o_{t,i}$ depend on the actual actions of the poachers $a_{t,i}$, the rangers' coverage probabilities $c_{t,i}$ and domain features $\{x^k_{t,i}\}$ Finally, we adopt the logistic model [4] to predict the poachers' behaviors; one advantage of this model compared to SUQR/QR is that it does not assume a known number of attackers and models probability of attack at every target independently. Thus, given the actual action of poachers, $a_{t-1,i}$, at previous time step $(t-1, i)$, the rangers' coverage probability $c_{t,i}$ at (t, i), and the domain features $\mathbf{x_{t,i}} = \{x^k_{t,i}\}$, we aim at predicting the probability that poachers attack (t, i) as follows:

$$p(a_{t,i} = 1|a_{t-1,i}, c_{t,i}, \mathbf{x_{t,i}}) = \frac{e^{\lambda'[a_{t-1,i}, c_{t,i}, \mathbf{x_{t,i}}, 1]}}{1 + e^{\lambda'[a_{t-1,i}, c_{t,i}, \mathbf{x_{t,i}}, 1]}} \quad (5)$$

where $\lambda = \{\lambda_k\}$ is the $(K + 3) \times 1$ parameter vector which measure the importance of all factors towards the poachers' decisions. λ_{K+3} is the free parameter and λ' is the transpose vector of λ. In essence, compared to Equation 4 where SUQR was seen to only use three features, we now have a weighted sum over a much larger number of features as is appropriate in our wildlife domain.

Furthermore, if the poachers attack at (t, i), we predict the probability that the rangers can detect any poaching signs as follows:

$$p(o_{t,i}=1|a_{t,i}=1,c_{t,i},\mathbf{x_{t,i}})=c_{t,i}\times\frac{e^{\mathbf{w}'[\mathbf{x_{t,i}},1]}}{1+e^{\mathbf{w}'[\mathbf{x_{t,i}},1]}} \quad (6)$$

where the first term is the probability that the rangers are present at (t,i) and the second term indicates the probability that the rangers can detect any poaching sign when patrolling at (t,i). Additionally, $\mathbf{w}=\{w_k\}$ is the $(K+1)\times 1$ vector of parameters which indicates the significance of domain features in affecting the rangers' probability of detecting poaching signs. $\mathbf{w}'$ is transpose of $\mathbf{w}$. In QENP specifically, CAPTURE employs seven features: animal density, distances to rivers/roads/villages, net primary productivity (NPP), habitat and slope to predict attacking/detection probabilities.

In the following, we will explain our approach for learning the parameters $(\lambda,\mathbf{w})$ of our hierarchical model. We use $p(a_{t,i}=1|a_{t-1,i},c_{t,i})$ and $p(o_{t,i}=1|a_{t,i}=1,c_{t,i})$ as the abbreviations of the LHSs in Equations 5 and 6. The domain features $\mathbf{x_{t,i}}$ are omitted in all equations for simplification.

3.2 Parameter Estimation

Due to the presence of unobserved variables $\mathbf{a}=\{a_{t,i}\}$, we use the standard Expectation Maximization (EM) method in order to estimate $(\lambda,\mathbf{w})$. In particular, EM attempts to maximize the log-likelihood that the rangers can have observations $\mathbf{o}=\{o_{t,i}\}$ given the rangers' coverage probabilities $\mathbf{c}=\{c_{t,i}\}$ and domain features $\mathbf{x}=\{\mathbf{x_{t,i}}\}$ for all time steps $t=1,\ldots,T$ and targets $i=1,\ldots,N$ which is formulated as follows:

$$\max_{\lambda,\mathbf{w}}\log p(\mathbf{o}|\mathbf{c},\mathbf{x},\lambda,\mathbf{w}) \quad (7)$$

The standard EM procedure [4] is to start with an initial estimate of $(\lambda,\mathbf{w})$ and iteratively update the parameter values until a locally optimal solution of (7) is reached. Many restarts are used with differing initial values of $(\lambda,\mathbf{w})$ to find the global optimum. Each iteration of EM consists of two key steps:

- **E** step: compute $p(\mathbf{a}|\mathbf{o},\mathbf{c},(\lambda,\mathbf{w})^{\mathbf{old}})$
- **M** step: update $(\lambda,\mathbf{w})^{\mathbf{old}})=(\lambda^*,\mathbf{w}^*)$ where $(\lambda^*,\mathbf{w}^*)=\underset{\lambda,\mathbf{w}}{\operatorname{argmax}}\sum_{\mathbf{a}}p(\mathbf{a}|\mathbf{o},\mathbf{c},(\lambda,\mathbf{w})^{\mathbf{old}})\log(p(\mathbf{o},\mathbf{a}|\mathbf{c},\lambda,\mathbf{w}))$.

In our case, the **E** (Expectation) step attempts to compute the probability that the poachers take actions $\mathbf{a}=\{a_{t,i}\}$ given the rangers' observations $\mathbf{o}$, the rangers' patrols $\mathbf{c}$, the domain features $\mathbf{x}=\{\mathbf{x_{t,i}}\}$, and current values of the model parameters $(\lambda,\mathbf{w})^{\mathbf{old}}$. The **M** (Maximization) step tries to maximize the expectation of the logarithm of the complete-data $(\mathbf{o},\mathbf{a})$ likelihood function given the action probabilities computed in the **E** step and updates the value of $(\lambda,\mathbf{w})^{\mathbf{old}}$ with the obtained maximizer.

Although we can decompose the log-likelihood, the EM algorithm is still time-consuming due to the large number of targets and parameters. Therefore, we use two novel ideas to speed up the algorithm: *parameter separation* for accelerating the convergence of EM and *target abstraction* for reducing the number of targets.

Parameter Separation. Observe that the objective in the **M** step can be split into two additive parts as follows:

$$\begin{aligned}&\sum\nolimits_{\mathbf{a}}p(\mathbf{a}|\mathbf{o},\mathbf{c},(\lambda,\mathbf{w})^{\mathbf{old}})\log(p(\mathbf{o},\mathbf{a}|\mathbf{c},\lambda,\mathbf{w})) \quad (8)\\&=\sum_{t,i}\sum_{a_{t,i}}p(a_{t,i}|\mathbf{o},\mathbf{c},(\lambda,\mathbf{w})^{\mathbf{old}})\log p(o_{t,i}|a_{t,i},c_{t,i},\mathbf{w})\\&+\sum_{t,i}\sum_{a_{t,i}}\sum_{a_{t-1,i}}p(a_{t,i},a_{t-1,i}|\mathbf{o},\mathbf{c},(\lambda,\mathbf{w})^{\mathbf{old}})\log p(a_{t,i}|a_{t-1,i},c_{t,i},\lambda)\end{aligned}$$

In (8), the first component is obtained as a result of decomposing w.r.t the detection probabilities of the rangers at every (t,i) (Equation 6). The second one results from decomposing according to the attacking probabilities at every (t,i) (Equation 5). Importantly, the first component is only a function of $\mathbf{w}$ and the second component is only a function of λ. Following this split, for our problem, the **E** step reduces to computing the following two quantities:

$$\textbf{Total probability: } p(a_{t,i}|\mathbf{o},\mathbf{c},(\lambda,\mathbf{w})^{\mathbf{old}}) \quad (9)$$

$$\textbf{2-step probability: } p(a_{t,i},a_{t-1,i}|\mathbf{o},\mathbf{c},(\lambda,\mathbf{w})^{\mathbf{old}}) \quad (10)$$

which can be computed by adapting the Baum-Welch algorithm [4] to account for missing observations, i.e., $o_{t,i}=-1$ when rangers do not patrol at (t,i). This can be done by introducing $p(o_{t,i}=-1|a_{t,i},c_{t,i}=0)=1$ when computing (9) and (10).

More importantly, as shown in (8), the structure of our problem allows for the decomposition of the objective function into two separate functions w.r.t attack parameters λ and detection parameters $\mathbf{w}$: $F^d(\mathbf{w})+F^a(\lambda)$ where the detection function $F^d(\mathbf{w})$ is the first term of the RHS in Equation 8 and the attack function $F^a(\lambda)$ is the second term. Therefore, instead of maximizing $F^d(\mathbf{w})+F^a(\lambda)$ we decompose each iteration of EM into two **E** steps and two **M** steps that enables maximizing F^d and F^a separately as follows:

- **E1** step: compute total probability
- **M1** step: $\mathbf{w}^*=\operatorname{argmax}_{\mathbf{w}}F^d(\mathbf{w})$; update $\mathbf{w}^{\mathbf{old}}=\mathbf{w}^*$
- **E2** step: compute 2-step probability
- **M2** step: $\lambda^*=\operatorname{argmax}_{\lambda}F^a(\lambda)$; update $\lambda^{\mathbf{old}}=\lambda^*$

Note that the detection and attack components are simpler functions compared to the original objective since these components only depend on the detection and attack parameters respectively. Furthermore, at each EM iteration, the parameters get closer to the optimal solution due to the decomposition since the attack parameter is now updated based on the new detection parameters from the **E1/M1** steps instead of the old detection parameters from the previous iteration. Thus, by decomposing each iteration of EM according to attack and detection parameters, EM will converge more quickly without loss of solution quality. The convergence and solution quality of the separation can be analyzed similarly to the analysis of multi-cycle expected conditional maximization [18].

Furthermore, the attack function $F^a(\lambda)$ is shown to be concave by Proposition 1 (its proof is in Online Appendix A[1]), allowing us to easily obtain the global optimal solution of the attacking parameters λ at each iteration of EM.

PROPOSITION 1. *$F^a(\lambda)$ is concave in the attack parameters λ.*

Target Abstraction. Our second idea is to reduce the number of targets via target abstraction. Previous work in network security and poker games has also applied abstraction for reducing the complexity of solving these games by exploring intrinsic properties of the games [2, 25]. In CAPTURE, by exploiting the spatial connectivity between grid cells of the conservation area, we can divide the area into a smaller number of grid cells by merging each cell in the original grid with its neighbors into a single bigger cell. The corresponding domain features are aggregated accordingly. Intuitively, neighboring cells tend to have similar domain features. Therefore, we expect that the parameters learned in both the original and abstracted grid would expose similar characteristics. Hence, the model parameters estimated based on the abstracted grid could be effectively used to derive the parameter values in the original one.

[1]https://www.dropbox.com/s/mngapyvv5112uhb/Appendix.pdf

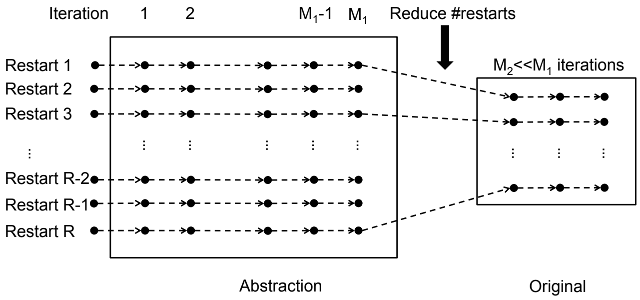


Figure 2: Target Abstraction

In this work, we leverage the values of parameters learned in the abstracted grid in two ways: (i) reduce the number of restarting points (i.e., initial values of parameters) for reaching different local optimal solutions in EM; and (ii) reduce the number of iterations in each round of EM. The idea of target abstraction is outlined in Figure 2 wherein each black dot corresponds to a set of parameter values at a particular iteration given a specific restarting points. At the first stage, we estimate the parameter values in the abstracted grid given a large number of restarting points R, assuming that we can run M_1 EM iterations. At the end of the first stage, we obtain R different sets of parameter values; each corresponds to a local optimal solution of EM in the abstracted grid. Then at the second stage, these sets of parameter values are used to estimate the model parameters in the original grid as the following: (i) only a subset of K resulting parameter sets which refer to the top local optimal solutions in the abstracted grid are selected as initial values of parameters in the original grid; and (ii) instead of running M_1 EM iterations again, we only proceed with $M_2 << M_1$ iterations in EM since we expect that these selected parameter values are already well learned in the abstracted grid and thus could be considered as *warm restarts* in the original grid.

4. PATROL PLANNING

Once the model parameters $(\lambda, \mathbf{w})$ are learned, we can compute the optimal patrolling strategies for the rangers in next time steps taking into account the CAPTURE model. We consider two circumstances: 1) single-step patrol planning in which the rangers only focus on generating the patrolling strategy at the next time step and 2) multiple-step patrol planning for generating strategies for the next $\Delta T > 1$ time steps, given the rangers' patrol and observation history and domain features. While the former provides a one-step patrolling strategy with an immediate but short-term benefit, the latter generates strategies across multiple time steps with a long-term benefit. We leave the choice of which planning option to use for the rangers given the cost/benefit trade-off between the two. The key challenge in designing strategies for the rangers given the CAPTURE model is that we need to take into account new aspects of the modeling of the adversary. These include the rangers' detection uncertainty and the temporal dependency of the poachers' activities. This challenge leads to a complicated non-convex optimization problem to compute the optimal patrolling strategy for the rangers; we provide novel game-theoretic algorithms to solve it.

We suppose that the rangers have an observation history $\mathbf{o} = \{o_{t',i}\}$ for $t' = 1, \ldots, T$ and $i = 1, \ldots, N$. Similar to standard SSGs, we assume that if the poachers successfully attack at (t, i), the rangers receive a penalty $P^d_{t,i}$. Conversely, if the rangers successfully confiscate poaching tools at (t, i), the rangers obtain a reward $R^d_{t,i}$. Therefore, the rangers' expected utility at (t, i) if the poachers attack at (t, i) is computed as follows where $p(o_{t,i} = 1|a_{t,i} = 1, c_{t,i})$ is the rangers' detection probability at (t, i) as shown in Equation 6:

$$U^d_{t,i} = p(o_{t,i} = 1|a_{t,i} = 1, c_{t,i}) \times [R^d_{t,i} - P^d_{t,i}] + P^d_{t,i} \qquad (11)$$

We now explain in detail our new game-theoretic algorithms. The rangers' past patrols at (t', i) for for $t' = 1, \ldots, T$ and $i = 1, \ldots, N$ are already known and thus can be omitted in all following mathematical formulations for simplification.

4.1 Single-step Patrol Planning

Given the rangers' observation history $\mathbf{o}$ and the model parameters $(\lambda, \mathbf{w})$, the problem of computing the optimal strategies at the next time step $T + 1$ can be formulated as follows:

$$\max_{\{c_{T+1,i}\}} \sum_i p(a_{T+1,i} = 1|\mathbf{o}, c_{T+1,i}) \times U^d_{T+1,i} \qquad (12)$$

$$s.t.\ 0 \le c_{T+1,i} \le 1,\ i = 1 \ldots N \qquad (13)$$

$$\sum_i c_{T+1,i} \le B \qquad (14)$$

where B is the maximum number of ranger resources and $p(a_{T+1,i} = 1|\mathbf{o}, c_{T+1,i})$ is the probability that the poachers attack at $(T + 1, i)$ given the rangers' observation history $\mathbf{o}$ and the rangers' coverage probability $c_{T+1,i}$. Since the poachers' behaviors depends on their activities in the past (which is hidden to the rangers), we need to examine all possible actions of the poachers in previous time steps in order to predict the poachers' attacking probability at $(T + 1, i)$. Hence, the attacking probability $p(a_{T+1,i} = 1|\mathbf{o}, c_{T+1,i})$ should be computed by marginalizing over all possible actions of poachers at (T, i) as follows:

$$p(a_{T+1,i} = 1|c_{T+1,i}, \mathbf{o}) = \sum_{a_{T,i}} p(a_{T+1,i} = 1|a_{T,i}, c_{T+1,i}) \times p(a_{T,i}|\mathbf{o}) \qquad (15)$$

where $p(a_{T+1,i}|a_{T,i}, c_{T+1,i})$, which is computed in (5), is the attacking probability at $(T + 1, i)$ given the poachers' action $a_{T,i}$ at (T, i) and the rangers' coverage probability $c_{T+1,i}$. In addition, $p(a_{T,i}|\mathbf{o})$ is the total probability at (T, i) which can be recursively computed based on the Baum-Welch approach as discussed in Section 3. Overall, (12 – 14) is a non-convex optimization problem in the rangers' coverage probabilities $\{c_{T+1,i}\}$. Fortunately, each additive term of the rangers' utility in (12) is a separate sub-utility function of the rangers' coverage, $c_{T+1,i}$, at $(T + 1, i)$:

$$f_i(c_{T+1,i}) = p(a_{T+1,i} = 1|\mathbf{o}, c_{T+1,i}) \times U^d_{T+1,i} \qquad (16)$$

Therefore, we can piecewise linearly approximate $f_i(c_{T+1,i})$ and represent (12 – 14) as a Mixed Integer Program which can be solved by CPLEX. The details of piecewise linear approximation can be found at [31]. Essentially, the piecewise linear approximation method provides an $O(\frac{1}{M})$-optimal solution for (12 – 14) where M is the number of piecewise segments [31].

4.2 Multi-step Patrol Planning

In designing multi-step patrol strategies for the rangers, there are two key challenges in incorporating the CAPTURE model that we need to take into account: 1) the time dependence of the poachers' behavior; and 2) the actual actions of the poachers are hidden (unobserved) from the rangers. These two challenges make the problem of planning multi-step patrols difficult as we show below.

Given that the rangers have an observation history $\mathbf{o} = \{o_{t'i}\}$ for $t' = 1, \ldots, T$ and $i = 1 \ldots N$, the rangers aim at generating patrolling strategies $\{c_{t,i}\}$ in next ΔT time steps where

$t = T+1, \ldots, T+\Delta T$. Then the problem of computing the optimal patrolling strategies for next ΔT time step $T+1, \ldots, T+\Delta T$ can be formulated as follows:

$$\max_{\{c_{t,i}\}} \sum\nolimits_{t,i} p(a_{t,i} = 1|\mathbf{o}, c_{T+1\ldots t,i}) U^d_{t,i} \tag{17}$$

$$s.t.\ 0 \le c_{t,i} \le 1,\ t = T+1 \ldots T+\Delta T,\ i = 1 \ldots N \tag{18}$$

$$\sum\nolimits_i c_{t,i} \le B,\ t = T+1 \ldots T+\Delta T. \tag{19}$$

where $p(a_{t,i} = 1|\mathbf{o}, c_{T+1\ldots t,i})$ is the attacking probability at (t, i) given the rangers' coverages at (t', i) where $t' = T+1, \ldots, t$ and observation history $\mathbf{o} = \{o_{t',i}\}$ where $t' = 1, \ldots, T$. Because of the two aforementioned challenges, we need to examine all possible actions of the poachers in previous time steps in order to compute the attacking probability at (t, i), $p(a_{t,i} = 1|\mathbf{o}, c_{T+1\ldots t,i})$. Our idea is to recursively compute this attacking probability via the attacking probabilities at previous time steps as follows:

$$p(a_{t,i} = 1|\mathbf{o}, c_{T+1\ldots t,i}) = \sum_{a_{t-1,i}} p(a_{t,i}|a_{t-1,i}, c_{t,i}) \times p(a_{t-1,i}|\mathbf{o}, c_{T+1\ldots t-1,i}) \tag{20}$$

where the initial step is to compute the total probability $p(a_{T,i}|\mathbf{o})$ by using the Baum-Welch approach. Here, the objective in (17) can be no longer divided into separate sub-utility functions of a single coverage probability at a particular (t, i) because of the time dependency of the poachers' behaviors. Thus, we can not apply piecewise linear approximation as in the single-step patrol planning for solving (17 – 19) quickly. In this work, we use non-convex solvers (i.e., fmincon in MATLAB) to solve (17 – 19).

In [9], the dependence of the attacker's actions on the defender's patrolling strategies in the past is also considered; they assume that the attacker's responses follow the SUQR model while the attacker perceives the defender's current strategy as a weighted linear function of the defender's strategies in the past. They also assume that these weights are known, thereby making the computational problem easy. In contrast, we make the more realistic assumption that the poachers are influenced by their own past observations and our learning algorithm learns the weights corresponding to such influence from the data. Unfortunately, this makes the problem of planning multistep patrols more difficult as shown before.

5. EXPERIMENTS

In our experiments, we aim to (i) extensively assess the prediction accuracy of the CAPTURE model compared to existing behavioral models based on real-world wildlife/poaching data; (ii) examine the runtime performance of learning the new model; and (iii) evaluate the solution quality of the CAPTURE planning for generating patrols. In the following, we provide a brief description of the real-world wildlife data used.

5.1 Real-world Wildlife/Poaching Data

In learning the poachers' behavior, we use the wildlife data collected by the rangers over 12 years from 2003 to 2014 in QENP (Figure 3 with animal density). This work is accomplished in collaboration with the Wildlife Conservation Society (WCS) and Uganda Wildlife Authority (UWA). While patrolling, the park rangers record information such as locations (latitude/longitude), times, and observations (e.g., signs of human illegal activities). Similar to [6], we also divide collected human signs into six different groups: commercial animal (i.e., human signs such as snares which refer to poaching commercial animals such as buffalo, hippo and elephant), non-commercial animal, fishing, encroachment, commercial plant, and non-commercial plant. In this work, we mainly focus on two types of human illegal activities: commercial animal and non-commercial animal which are major threats to key species of concern such as elephants and hippos.

The poaching data is then divided into the four different groups according to four seasons in Uganda: dry season I (Jun, July, and August), dry season II (December, January, and February), rainy season I (March, April, and May), and rainy season II (September, October, November). We aim at learning behaviors of the poachers w.r.t these four seasons as motivated by the fact that the poachers' activities usually vary seasonally. In the end, we obtain eight different categories of wildlife data given that we have the two poaching types and four seasons. Furthermore, we use seven domain features in learning the poachers' behavior, including animal density, slope, habitat, net primary productivity (NPP), and locations of villages/rivers/roads provided by [6].

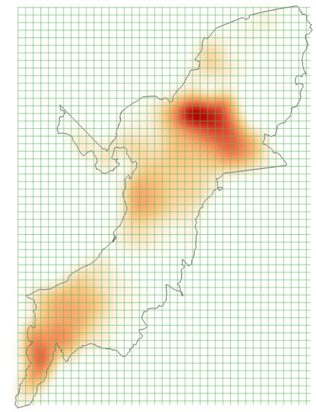


Figure 3: QENP with animal density

We divide the park area into a $1km \times 1km$ grid consisting of more than 2500 grid cells ($\approx 2500km^2$). Domain features and the rangers' patrols and observations are then aggregated into the grid cells. We also refine the poaching data by removing all abnormal data points such as the data points which indicate that the rangers conducted patrols outside the QENP park or the rangers moved too fast, etc. Since we attempt to predict the poachers' actions in the future based on their activities in the past, we apply a time window (i.e., five years) with an 1-year shift to split the poaching data into eight different pairs of training/test sets. For example, for the (commercial animal, rainy season I) category, the oldest training/test sets correspond to four-year data (2003–2006) w.r.t this category for training and one-year (2007) data for testing. In addition, the latest training/test sets refer to the four years (2010–2013) and one year (2014) of data respectively. In total, there are eight different training/test sets for each of our eight data categories.

5.2 Behavioral Learning

Prediction Accuracy. In this work, we compare the prediction accuracy of six models: 1) CAPTURE (CAPTURE with parameter separation); 2) CAP-Abstract (CAPTURE with parameter separation and target abstraction); 3) CAP-NoTime (CAPTURE with parameter separation and without the component of temporal effect); 4) Logit (Logistic Regression); 5) SUQR ; and 6) SVM (Support Vector Machine). We use AUC (Area Under the Curve) to measure the prediction accuracy of these behavioral models. Based on ROC plots of data, AUC is a standard and common statistic in machine learning for model evaluation [5]. Essentially, AUC refers to the probability that a model will weight a random positive poaching sample higher than a random negative poaching sample in labeling these samples as positive (so, higher AUC values are better). For each data category (w.r.t poaching types and poaching seasons), the AUC values of all the models are averaged over the eight test sets as explained in Section 5.1. We also show the average prediction accuracy over all seasons. We use bootstrap-t [29] to measure the

Models	Rainy I	Rainy II	Dry I	Dry II	Average
CAPTURE	0.76	0.76	0.74	0.73	0.7475
CAP-Abstract	0.79	0.76	0.74	0.67	0.74
CAP-NoTime	0.71	0.75	0.67	0.71	0.71
Logit	0.53	0.59	0.57	0.60	0.5725
SUQR	0.53	0.59	0.56	0.62	0.575
SVM	0.61	0.59	0.51	0.66	0.5925

Table 1: AUC: Commercial Animal

Models	Rainy I	Rainy II	Dry I	Dry II	Average
CAPTURE	0.76	0.70	0.78	0.72	0.74
CAP-Abstract	0.76	0.70	0.74	0.70	0.725
CAP-NoTime	0.72	0.68	0.75	0.70	0.7125
Logit	0.52	0.63	0.57	0.52	0.56
SUQR	0.54	0.62	0.58	0.54	0.57
SVM	0.42	0.50	0.55	0.56	0.5075

Table 2: AUC: Non-Commercial Animal

statistical significance of our results.

The results are shown in Tables 1 and 2. We can infer the following key points from these tables. First, and most important, CAPTURE improves performance over the state of the art, which is SUQR and SVM. CAPTURE's average AUC in Table 1 (essentially this is over 32 data points of eight test sets over four seasons) is 0.7475 vs 0.575 for SUQR, and in Table 2 is 0.74 vs 0.57 for SUQR. This clearly shows a statistically significant ($\alpha = 0.05$) advance in our modeling accuracy. This improvement illustrates that all the four advances in CAPTURE mentioned in Section 1 — addressing observation error, time dependence, detailed domain features and not requiring a firm count of poachers beforehand – have indeed led to a significant advance in CAPTURE's performance. We can now attempt to understand the contributions of each of CAPTURE's improvements, leading to the next few insights. Second, comparison of CAPTURE with CAP-NoTime which only addresses the challenge of observation bias demonstrates the importance of considering time dependence. Third, while parameter separation does not cause any loss in solution quality as discussed in Section 3.2, Tables 1 and 2 shows that the prediction accuracy of CAPTURE with target abstraction is good in general except for Dry season II with Commercial Animal. As we show later, parameter separation and target abstraction help in speeding up the runtime performance of learning the CAPTURE model.

Fourth, the results of the model parameter values in the CAPTURE model show that all these domain features substantially impact the poachers' behaviors. For example, one learning result on the model parameters corresponding to the category (non-commercial animal/dry season I) in 2011 is $(0.33, 1.46, -2.96, -1.97, 1.88, -0.78, 0.36)$ for domain features (habitat, NPP, slope, road distance, town distance, water distance, and animal density), -1.40 for the rangers' coverage probability and 4.27 for the poachers' past action. Based on these learned weights, we can interpret how these domain features affect the poachers' behavior. Specifically, the negative weights for road/water distances indicates that the poachers tend to poach at locations near roads/water. In addition, the resulting positive weight for the poachers' past actions indicates that the poachers are more likely to attack the targets which were attacked before. Furthermore, the resulting negative weight for the rangers' patrols also shows that the poachers' activity is influenced by the rangers' patrols, i.e., the poachers are less likely to attack targets with higher coverage probability of the rangers. Lastly, the ranger-poacher interaction changes over time as indicated by different negative weights of the rangers' patrols across different years (Table 3). For example, the patrol weight corresponding the category (non-commercial animal/dry season II) in 2014 is -17.39 while in 2013 is -1.78, showing that rangers' patrols have more impact on the poachers' behavior in 2014 than in 2013. This is the first time there is a real-world evidence which shows the impact of ranger patrols on poacher behavior.

Runtime Performance. We compare the runtime performance of learning the CAPTURE model in three cases: 1) learning without both heuristics of parameter separation and target abstraction; 2) learning with parameter separation only; and 3) learning with both heuristics. In our experiments, for the first two cases, we run 20 restarting points and 50 iterations in EM. In the third case, we first run 20 restarting points and 40 iterations in EM with target abstraction. In particular, in target abstraction, we aggregate or interpolate all domain features as well as the rangers' patrols into $4km \times 4km$ grid cells while the original grid cell size is $1km \times 1km$. Then given the results in the abstracted grid, we only select 5 results of parameter values (which correspond to the top five prediction accuracy results w.r.t the training set). We use these results as restarting points for EM in the original grid and only run 10 iterations to obtain the final learning results in the original grid.

Year	2009	2010	2011	2012	2013	2014
Weight	-10.69	-4.35	-0.7	-2.21	-1.78	-17.39

Table 3: Patrol weights in recent years

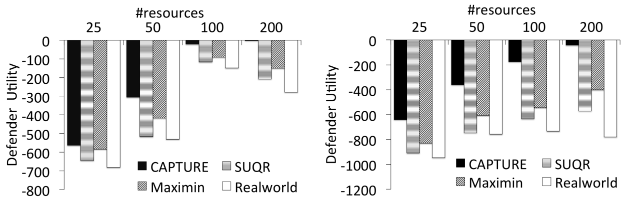
Heuristics	Average Runtime
None	1419.16 mins
Parameter Separation	333.31 mins
Parameter Separation w/ Target Abstraction	222.02 mins

Table 4: CAPTURE Learning: Runtime Performance

The results are shown in Table 4 which are averaged over 64 training sets (statistically significant ($\alpha = 0.05$)). In Table 4, learning CAPTURE model parameters with parameter separation is significantly faster (i.e., 4.25 times faster) than learning CAPTURE without this heuristic. This result clearly shows that reducing the complexity of the learning process (by decomposing it into simpler sub-learning components via parameter separation) significantly speeds up the learning process of CAPTURE. Furthermore, the heuristic of target abstraction helps CAPTURE in learning even faster although the result is not as substantial as with parameter separation, demonstrating the advantage of using this heuristic.

5.3 Patrol Planning

Based on the CAPTURE model, we apply our CAPTURE planning algorithm (Section 4) to compute the optimal patrolling strategies for the rangers. The solution quality of our algorithm is evaluated based on the real-world QENP domain in comparison with SUQR (i.e., optimal strategies of the rangers against SUQR-based poachers), Maximin (maximin strategies of the rangers against worst-case poacher responses), and Real-world patrolling strategies of the rangers. The real-world strategies are derived from the four seasons in years 2007 to 2014. Given that CAPTURE's prediction accuracy is the highest among all the models, in our experiments, we assume that the poachers' responses follow our model. Given the QENP experimental settings, the reward of the rangers at each target are set to be zero while the penalty is the opposite of the animal density (i.e., zero-sum games). We assess the solution quality of all algorithms according to different number of the rangers' resources (i.e., number of targets the rangers can cover during a patrol). The real-world patrolling strategies are normalized accordingly. Moreover, we also consider different number of time steps for generating patrols.



(a) 2 time steps (b) 4 time steps

Figure 4: Solution quality of CAPTURE-based planning

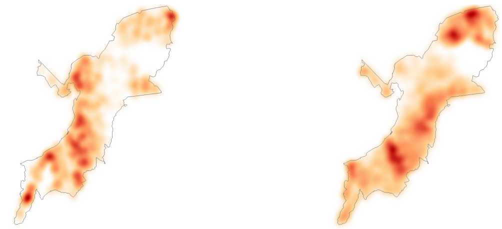
The experimental results are shown in Figure 4 which are averaged over all years and seasons. In Figure 4, the x-axis is the number of the rangers' resources and the y-axis is the aggregated utility the rangers receive over two and four time steps (seasons) for playing CAPTURE, SUQR, Maximin, and Real-world patrolling strategies respectively. As shown in Figure 4, our CAPTURE planning algorithm provides the highest utility for the rangers (with statistical significance ($\alpha = 0.05$)). Especially when the number of the rangers' resources increases, the CAPTURE planning algorithm significantly improves the quality of the rangers' patrolling strategies. Furthermore, our CAPTURE algorithm provides patrolling strategies which take into account the temporal effect on the poachers' behaviors. As a result, when the number of time steps increases (Figure 4(b)), our algorithm enhances its solution quality compared to the others.

6. CAPTURE-BASED APPLICATION

CAPTURE tool is available for the rangers to predict the poachers' behavior and design optimal patrol schedules. Not all the regions are equally attractive to the poachers, so it is beneficial to detect the hotspots and favorite regions for poachers and protect those areas with higher probability. The general work-flow for this software could be itemized as: 1) Aggregating previously gathered data from the park to create a database that includes domain features, poaching signs and rangers' effort to protect the area; 2) Preprocessing of the data points; 3) Running the CAPTURE tool to predict the attacking probability, rangers' observation over the area and generate the optimal patrol strategy; and 4) Post-processing of the results and generating the related heatmaps.

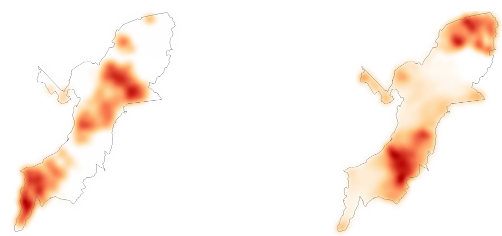
To compare the optimal strategy generated by the single-step patrol planning algorithm provided by CAPTURE and current real strategy deploying over the area, we plotted the related heatmaps according to the defender coverage, shown in Figure 5(a) and Figure 6(a). The darker the area, the greater chance to be covered by the rangers. Also, we used CAPTURE to predict the probability of the attack based on these patrol strategies. These heatmaps are shown in Figure 5(b) and Figure 6(b). The darker regions on the map demonstrate the more attractive regions to the poachers.

We can see the following key points based on the heatmaps: (i) The optimal patrol strategy covers more of the regions with higher animal density (for instance south-west and middle parts of the park as shown in Figure 3). So the deployment of the optimal strategy would result in more protection to areas with higher animal density, as shown in Figure 6(a) and 6(b). (ii) The poaching heatmap shows significantly higher predicted activity of attackers against human generated patrols in regions with higher animal density, as shown in Figure 5(a) and 5(b).



(a) Patrol strategy (b) Attack probability

Figure 5: Heatmaps by CAPTURE (based on the real patrol strategy)



(a) Patrol strategy (b) Attack probability

Figure 6: Heatmaps by CAPTURE (based on the optimal strategy)

7. CONCLUSION

We propose a new predictive anti-poaching tool, CAPTURE. Essentially, CAPTURE introduces a novel hierarchical model to predict the poachers' behaviors. The CAPTURE model provides a significant advance over the state-of-the-art in modeling poachers in security games [8] and in conservation biology [11, 6] via 1) addressing the challenge of imperfect observations of the rangers; 2) incorporating the temporal effect on the poachers' behaviors; and 3) not requiring a known number of attackers. We provide two new heuristics: parameter separation and target abstraction to reduce the computational complexity in learning the model parameters. Furthermore, CAPTURE incorporates a new planning algorithm to generate optimal patrolling strategies for the rangers, taking into account the new complex poacher model. Finally, this application presents an evaluation of the largest sample of real-world data in the security games literature, i.e., over 12-years of data of attacker defender interactions in QENP. The experimental results demonstrate the superiority of our model compared to other existing models. CAPTURE will be tested in QENP in early 2016.

Acknowledgements: This research was supported by MURI Grant W911NF-11-1-0332 and by CREATE under grant number 2010-ST-061-RE0001. We are grateful to the Wildlife Conservation Society, Uganda Wildlife Authority, MacArthur Foundation, USAID, US Fish and Wildlife Service, and Daniel K. Thorne Foundation for supporting the data collection in Queen Elizabeth National Park. We wish to acknowledge the support of the University of York. We would like to thank Bazil Andira and all the rangers and wardens in Queen Elizabeth National Park for their contribution in collecting the law enforcement monitoring data in MIST.

REFERENCES

[1] D. Avrahami-Zilberbrand and G. A. Kaminka. Keyhole adversarial plan recognition for recognition of suspicious and anomalous behavior. In *Plan, Activity, and Intent Recognition*, pages 87–121. 2014.

[2] N. Basilico and N. Gatti. Automated abstractions for patrolling security games. In *AAAI*, 2011.

[3] N. Basilico, N. Gatti, and F. Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *AAMAS*, 2009.

[4] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

[5] A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[6] R. Critchlow, A. Plumptre, M. Driciru, A. Rwetsiba, E. Stokes, C. Tumwesigye, F. Wanyama, and C. Beale. Spatiotemporal trends of illegal activities from ranger-collected data in a ugandan national park. *Conservation Biology*, 2015.

[7] J. S. De Bruin, T. K. Cocx, W. Kosters, J. F. Laros, J. N. Kok, et al. Data mining approaches to criminal career analysis. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 171–177. IEEE, 2006.

[8] F. Fang, T. H. Nguyen, R. Pickles, W. Y. Lam, G. R. Clements, B. An, A. Singh, M. Tambe, and A. Lemieux. Deploying paws: Field optimization of the protection assistant forwildlife security. In *IAAI*, 2016.

[9] F. Fang, P. Stone, and M. Tambe. When security games go green: Designing defender strategies to prevent poaching and illegal fishing. In *IJCAI*, 2015.

[10] S. Ganzfried and T. Sandholm. Game theory-based opponent modeling in large imperfect-information games. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 533–540. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[11] H. HOFER, K. L. CAMPBELL, M. L. EAST, and S. A. HUISH. Modeling the spatial distribution of the economic costs and benefits of illegal game meat hunting in the serengeti. *Natural Resource Modeling*, 13(1):151–177, 2000.

[12] D. Kar, F. Fang, F. D. Fave, N. Sintov, and M. Tambe. A game of thrones: When human behavior models compete in repeated stackelberg security games. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*, 2015.

[13] D. Korzhyk, V. Conitzer, and R. Parr. Complexity of computing optimal stackelberg strategies in security resource allocation games. In *AAAI*, 2010.

[14] J. Letchford and Y. Vorobeychik. Computing randomized security strategies in networked domains. In *AARM*, 2011.

[15] D. I. MacKenzie, J. D. Nichols, G. B. Lachman, S. Droege, J. Andrew Royle, and C. A. Langtimm. Estimating site occupancy rates when detection probabilities are less than one. *Ecology*, 83(8):2248–2255, 2002.

[16] D. McFadden. Conditional logit analysis of qualitative choice behavior. Technical report, 1972.

[17] R. McKelvey and T. Palfrey. Quantal response equilibria for normal form games. *Games and economic behavior*, 10(1):6–38, 1995.

[18] X.-L. Meng and D. B. Rubin. Maximum likelihood estimation via the ecm algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993.

[19] M. Montesh. Rhino poaching: A new form of organised crime1. Technical report, University of South Africa, 2013.

[20] S. V. Nath. Crime pattern detection using data mining. In *Web Intelligence and Intelligent Agent Technology Workshops, 2006. WI-IAT 2006 Workshops. 2006 IEEE/WIC/ACM International Conference on*, pages 41–44. IEEE, 2006.

[21] T. H. Nguyen, F. M. Delle Fave, D. Kar, A. S. Lakshminarayanan, A. Yadav, M. Tambe, N. Agmon, A. J. Plumptre, M. Driciru, F. Wanyama, et al. Making the most of our regrets: Regret-based solutions to handle payoff uncertainty and elicitation in green security games. In *GameSec*, 2015.

[22] T. H. Nguyen, R. Yang, A. Azaria, S. Kraus, and M. Tambe. Analyzing the effectiveness of adversary modeling in security games. In *AAAI*, 2013.

[23] G. Oatley, B. Ewart, and J. Zeleznikow. Decision support systems for police: Lessons from the application of data mining techniques to "soft" forensic evidence. *Artificial Intelligence and Law*, 14(1-2):35–100, 2006.

[24] U. H. C. on Foreign Affairs. Poaching and terrorism: A national security challenge (serial no. 114-25). Washington:US Government Publishing Office, April 2015.

[25] T. Sandholm and S. Singh. Lossy stochastic game abstraction with bounds. In *EC*, pages 880–897. ACM, 2012.

[26] G. Secretariat. Global tiger recovery program implementation plan: 2013-14. *Report, The World Bank, Washington, DC*, 2013.

[27] F. Southey, M. P. Bowling, B. Larson, C. Piccione, N. Burch, D. Billings, and C. Rayner. Bayes' bluff: Opponent modelling in poker. *arXiv preprint arXiv:1207.1411*, 2012.

[28] M. Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2011.

[29] R. Wilcox. *Applying contemporary statistical techniques*. Academic Press, 2002.

[30] R. Yang, B. Ford, M. Tambe, and A. Lemieux. Adaptive resource allocation for wildlife protection against illegal poachers. In *AAMAS*, 2014.

[31] R. Yang, F. Ordonez, and M. Tambe. Computing optimal strategy against quantal response in security games. AAMAS, 2012.

[32] C. Zhang, A. Sinha, and M. Tambe. Keeping pace with criminals: Designing patrol allocation against adaptive opportunistic criminals. In *AAMAS*, 2015.