Real-World Evaluation and Deployment of Wildlife Crime Prediction Models

by

Benjamin Ford

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(Computer Science)

August 2017

Acknowledgments

I would first like to thank my advisor Milind Tambe. When I first joined Teamcore, I was impressed that all of your students respected you a great deal, but after having been your student for four years, I am even more impressed. Your ability to give thoughtful, constructive criticism is unlike anyone's I've ever seen — never mind that you personally mentor no less than 12 students at a time and are able to be so involved in all of our projects (this will never cease to amaze me). Your commitment to having a real-world impact and ability to successfully navigate new domains has been enlightening and has shaped how I see the role of computer science in the world and is something I hope to emulate in my own career. I am grateful to you for giving me the opportunity to work on the projects I've worked on over the years, especially the wildlife conservation project. I never would have thought this project would have been possible prior to joining, and now I can't think of any other project that I would have had as much pride in as this one. Thank you for pushing me to continue my PhD, even when I had serious doubts about my abilities to finish it. I am humbled that your mentorship has extended long after graduation for all of your students, and it fills me with great pride to be a part of this Teamcore family.

I would like to thank my committee members, Richard John, Eric Rice, and Ning Wang for serving on my committee. Your enthusiasm and great advice has strengthened my research and

further encouraged me to pursue these kinds of real-world topics in my career. I am deeply thankful for all your time and effort in supporting me.

I would also like to thank Lizsl De Leon for all of her program guidance and for being an informal guidance counselor throughout my PhD. Your heartfelt dedication to all CS students amazes me, and we are all super lucky to have you as our advocate. I'd like to also thank Kusum Shori, Lifeng (Mai) Lee, Michael Archuleta, and everyone on the Computer Science department staff. Your hard work and dedication to the USC CS student body is appreciated, and I will miss all of you!

I would like to thank Andrew Plumptre of the Wildlife Conservation Society for providing me with the incredible opportunity to conduct this real-world wildlife crime research and his endless support and guidance throughout this research. I would also like to thank the Wildlife Conservation Society as a whole and all the rangers of Uganda Wildlife Authority. Without your concrete efforts in foot patrolling the Queen Elizabeth Protected Area, sharing data, and field testing my models, none of this work would have been possible, and it has been my deepest honor and privilege to work with you and Andrew Plumptre. Additionally, I thank Andrew Lemieux for his support in the early stages of my PhD — exposing me to the world of conservation and the possibilities for this research to make a positive real-world impact.

Next, I thank Nicole Sintov and Matthew Brown for mentoring me in the early years of my PhD. This was a period of great uncertainty for me, and your encouragement and guidance was invaluable toward me sticking with the program and being where I am today. Nicole: your expertise in human behavior, human subject experiments, and experimental design encouraged me to learn more about these areas, and these have since become my primary areas of research interest. I cannot thank you enough for your collaboration and patience in showing me how these

areas of research can effectively intersect with traditional computer science, and my research and career passions would not be the same without your guidance. Additionally, I thank Christopher Kiekintveld, Francesco Delle Fave, and Biplav Srivastava for their research collaborations on exciting research papers.

Everyone in the Teamcore family, it has been my immense pleasure and privilege to share this four-year journey with all of you: Manish Jain, Jun-young Kwak, Rong Yang, Jason Tsai, Albert Jiang, William Haskell, Eric Shieh, Francesco Delle Fave, Matthew Brown, Arunesh Sinha, Thanh Nguyen, Leandro Marcolino, Fei Fang, Chao Zhang, Yundi Qian, Debarun Kar, Haifeng Xu, Amulya Yadav, Aaron Schlenker, Sara Mc Carthy, Yasaman Abbasi, Shahrzad Gholami, Bryan Wilder, Elizabeth Orrico, Elizabeth Bondi, Subhasree Sengupta, and Aida Rahmattalabi. I think it is amazing that we've all become such great friends that are there to support each other 100% through thick and thin. I will always have fond memories of traveling to conferences around the world and experiencing those new places with you. I will miss seeing you all in the office, and I look forward to hearing about everyone's future successes! A special thanks to Fei, Debarun, and Shahrzad: I will fondly remember all of the time we toiled away on the wildlife poaching problem; it would not have been nearly as enjoyable (or successful) without working with the three of you. Also a special thanks to Donnabell Dmello for her help in analyzing my field test data: thank you so much for helping me out of the blue when I knew you were super busy!

In addition to all of my friends and family back home in Massachusetts and in Los Angeles, whose support has been an appreciated source of encouragement and strength, I would like to take the time to thank some in particular, without whom my time at USC would be very different and perhaps unsuccessful. Elizabeth Staruk: thank you for taking me in at a time when I needed a

friend the most. Your understanding and unblinking compassion got me through one of the most trying experiences in my life, and I will be forever grateful to you. Gordon Bellamy: thank you for believing in me, encouraging my participation in your game development class, and supporting my trip to GDC. All of this was at a time where I didn't know what to do with myself or my future, and your unwavering confidence in my abilities and character helped me rediscover myself and finish my PhD. Benjamin Scanlon: thank you for your love, patience, and unending support during the PhD application process, my move out to Los Angeles, and throughout my PhD. I'm not sure I would've gotten through my first years of adjustment in LA or most of this PhD without you. Arlen Printz: thank you for sharing your life with me as I began to finish my PhD and as you began your own journey into law school. Your constant love, strength, encouragement, and levity has made it possible for me to carry through each day and reach where I am today.

Lastly, I want to thank my parents and sister for their love, support, and patience with me during these four years. I would not be where I'm at today without their never-ending encouragement and belief in me throughout my life.

Contents

List Of Figures

List Of Tables

Abstract

Conservation agencies worldwide must make the most efficient use of their limited resources to protect natural resources from over-harvesting and animals from poaching. Predictive modeling, a tool to increase efficiency, is seeing increased usage in conservation domains such as to protect wildlife from poaching. Many works in this wildlife protection domain, however, fail to train their models on real-world data or test their models in the real world. My thesis proposes novel poacher behavior models that are trained on real-world data and are tested via first-of-their-kind tests in the real world.

First, I proposed a paradigm shift in traditional adversary behavior modeling techniques from Quantal Response-based models to decision tree-based models. Based on this shift, I proposed an ensemble of spatially-aware decision trees, INTERCEPT, that outperformed the prior state-of-the-art and then also presented results from a one-month pilot field test of the ensemble's predictions in Uganda's Queen Elizabeth Protected Area (QEPA). This field test represented the first time that a machine learning-based poacher behavior modeling application was tested in the field.

Second, I proposed a hybrid spatio-temporal model that led to further performance improvements. To validate this model, I designed and conducted a large-scale, eight-month field test of this model's predictions in QEPA. This field test, where rangers patrolled over 450 km in the

largest and longest field test of a machine learning-based poacher behavior model to date in this domain, successfully demonstrated the selectiveness of the model's predictions; the model successfully predicted, with statistical significance, where rangers would find more snaring activity and also where rangers would not find as much snaring activity. I also conducted detailed analysis of the behavior of my predictive model.

Third, beyond wildlife poaching, I also provided novel graph-aware models for modeling human adversary behavior in wildlife or other contraband smuggling networks and tested them against human subjects. Lastly, I examined human considerations of deployment in new domains and the importance of easily-interpretable models and results. While such interpretability has been a recurring theme in all my thesis work, I also created a game-theoretic inspection strategy application that generated randomized factory inspection schedules and also contained visualization and explanation components for users.

Chapter 1

Introduction

Worldwide, conservation agencies are tasked with protecting natural resources from over-harvesting and sustaining ecosystems by protecting key species from poaching. Unfortunately, law enforcement agencies are severely under-resourced, and it is an ongoing challenge for them to adequately protect the vast areas they are tasked to protect. It is thus of utmost importance that law enforcement can identify the most at-risk areas to maximize their efficiency.

Predictive modeling is a paradigm that has seen wide application in the Criminology literature (Perry, 2013; Eck, Chainey, Cameron, & Wilson, 2005; Beck & McCue, 2009) and recently has been gaining momentum in the wildlife protection domain (Yang, Ford, Tambe, & Lemieux, 2014; Haines, Elledge, Wilsing, Grabe, Barske, Burke, & Webb, 2012; Koen, de Villiers, Pavlin, de Waal, de Oude, & Mignet, 2014; Nguyen, Sinha, Gholami, Plumptre, Joppa, Tambe, Driciru, Wanyama, Rwetsiba, Critchlow, et al., 2016; Rashidi, Wang, Skidmore, Vrieling, Darvishzadeh, Toxopeus, Ngene, & Omondi, 2015; Critchlow, Plumptre, Driciru, Rwetsiba, Stokes, Tumwesigye, Wanyama, & Beale, 2015). In these works, the goal is to use collected crime data to predict where future illegal activity will occur. For the wildlife protection domain, crime data typically corresponds to data collected by park rangers while they are patrolling the conservation

area. Once a predictive model is trained on the crime data, predictions can be generated that then enable law enforcement to plan more targeted interventions to more efficiently prevent crime.

The key challenge of predictive modeling lies in its primary component: data. In the wildlife protection domain, the patrolling and data collection process is labor-intensive and thus results in sparse datasets. Additionally, patrolling is an imperfect process due to observability challenges in the real world. When looking for snares, for example, rangers may patrol an area and miss a well-hidden snare. When making their observations about the area, rangers would then mistakenly label the area as having no snares. This results in significant noise in the negative labels (i.e., no observed poaching activity) and presents difficulties for learning accurate adversary models. Finally, compounded by the noise in negative labels, there can be substantial class imbalances in real-world poaching datasets; more often than not, there will be more occurrences of "no poaching observed" than observed poaching activities. This presents even more challenges to models that have to accurately predict positive data points corresponding to a comparatively tiny minority class.

Most of the previous work in the wildlife protection domain focuses solely on developing predictive models (Yang et al., 2014; Haines et al., 2012; Koen et al., 2014; Nguyen et al., 2016; Rashidi et al., 2015; Critchlow et al., 2015). However, some works do not train their models on real-world data (Yang et al., 2014; Koen et al., 2014). While it is non-trivial to gain access to these confidential data sets, it is important to ensure that any developed techniques can handle the sparsity and noise in real-world poaching datasets. Other works do not conduct a sufficient empirical comparison among their proposed model and well-known baselines (Haines et al., 2012; Critchlow et al., 2015). Without such an evaluation, it is difficult to assess the predictive accuracy of proposed models and not possible to compare new models against their work. Finally, even

though real-world field testing is difficult to conduct in this domain, it is important to conduct nonetheless so that models are properly validated and incrementally improved based on feedback. Besides the work in this thesis, a thorough search of the literature has confirmed only one other field test to date of a predictive model in this domain (Critchlow, Plumptre, Andira, Nsubuga, Driciru, Rwetsiba, Wanyama, & Beale, 2016) and no works that field test a machine learning-based predictive model in this domain.

My thesis focuses primarily on the challenges of modeling adversary behavior in the wildlife protection domain and evaluating those models in the real world. In this thesis, I address the challenges associated with learning adversary models from real-world poaching data and then design and conduct field tests of said adversary models, including both a decision tree ensemble and a hybrid model. I also analyze how a decision tree ensemble's predictions change in response to changes in ranger effort (i.e., how often an area is patrolled) to assess the viability of decision tree-based predictive models as input to sophisticated patrol generation frameworks (that modify ranger effort).

Additionally, my thesis presents work that examines important considerations for deploying interventions based off of these models. First, I focus on predictive modeling in trafficking networks and closely examine and challenge the implicit assumption that the best model will also have the best corresponding intervention. Lastly, I consider the environmental compliance problem in the context of preventing river water pollution and the issues that arise in explaining and visualizing game-theoretic interventions.

1.1 Combating the Source of Wildlife Crime: Poacher Behavior Models in the Real World

The first part of my thesis focuses on developing and evaluating poaching prediction models in the real world. Predictive modeling could help conservation and law enforcement agencies make more efficient use of their limited resources to combat poaching.

As discussed earlier, wildlife poaching is a difficult phenomenon to model given the complexities and challenges of real-world poaching data. And even if a model accounts for these complexities, there are no guarantees that the model will perform well in the field especially considering that the dataset itself may not accurately represent reality. As such, it is also necessary to validate any predictive models via field testing.

Given that prior works either did not train their models on real-world data (Yang et al., 2014; Koen et al., 2014) or suffered from some practical limitations that precluded their use in a field test (Nguyen et al., 2016), I proposed INTERCEPT (INTERpretable Classification Ensemble to Protect Threatened species), an ensemble of spatially-aware decision trees that was trained on 13 years of real-world data from Uganda's Queen Elizabeth Protected Area (QEPA). This technique represented a paradigm shift from traditional adversary modeling techniques, such as those based on the Quantal Response model (Yang, Kiekintveld, Ordonez, Tambe, & John, 2011). This shift was necessary due to those types of models being either unable to cope with the dataset challenges previously described or being too computationally expensive to run with rangers' limited computing power. Because decision trees did not take into account the spatial correlations present in the dataset (e.g., crime happens in proximity to crime), I introduced a spatially-aware decision tree algorithm, BoostIT, that significantly improved recall with only modest losses in precision. I

also conducted an extensive empirical evaluation of 41 different models and a total of 193 model variants and demonstrated INTERCEPT's superior performance to the previous state-of-the-art, CAPTURE (Nguyen et al., 2016), and many other baselines.

I also present the results of two field tests I designed. These field tests represented the first times that machine learning-based adversary models were tested in the real world in this domain. The first field test, of INTERCEPT, was conducted over one month in QEPA where I proposed two areas for rangers to patrol based on INTERCEPT's predictions. These areas were chosen because they were previously not patrolled frequently, there were no prior observations of snaring activity in those areas, and INTERCEPT predicted those areas to be attackable. The second field test represented a larger scale deployment of a hybrid (Markov Random Fields and a decision tree ensemble) predictive model that took place in 27 areas over the entirety of QEPA for eight months. In this field test, rangers patrolled approximately 452 kilometers, and from the patrol results, I demonstrated the selectiveness of the hybrid model in that it successfully differentiated between areas of high poaching activity and low poaching activity with statistical significance.

Finally, I present a detailed analysis of how a decision tree ensemble-based predictive detection model would react to changes in the rangers' patrolling strategy; in short, the analysis seeks to answer questions such as "If we increase patrols in this area, will we detect more snaring activity than if we didn't increase patrols?" This analysis is a key step toward ensuring the model's viability as input to sophisticated intervention methodologies (e.g., patrol generation algorithms); if a patrol generation algorithm proposes an increase in patrolling effort in a given area on a given day, the model should not then predict a decrease in the number of detected activities in that area (before poachers have any time to react to that increase in effort).

1.2 Predictive Reliability and the Need for Field Testing of Interventions

Network Security Games (NSGs), a type of Security Game (Tambe, 2011), can be applied to interdict the flow of goods in smuggling networks (e.g., illegal drugs, ivory). In an NSG, the goal is to aid the defender (e.g., law enforcement) in making the most efficient use of her limited resources by allocating those resources to key edges (e.g., roads, ports) in the network. The attacker (e.g., smuggler) conducts long-term surveillance on the defender's actions and is able to infer the probability of which each edge will be protected on a given day. After conducting this surveillance, the attacker chooses a path through the network.

Many works in the Security Games literature model human behavior to increase the potential gains of the defender (Nguyen, Yang, Azaria, Kraus, & Tambe, 2013; Cui & John, 2014; Kar, Fang, Fave, Sintov, & Tambe, 2015; Abbasi, Short, Sinha, Sintov, Zhang, & Tambe, 2015). However, real-world data is not commonly used in these works due to the demonstrated costs associated with obtaining that data (Shieh, An, Yang, Tambe, Baldwin, DiRenzo, Maule, & Meyer, 2012; Fave, Jiang, Yin, Zhang, Tambe, Kraus, & Sullivan, 2014). As such, it is important for models to be able to work with small amounts of data. Additionally, other previous works empirically compare the performance of different human behavior models' ability to predict the actions of humans. However, they do not address how the models' corresponding strategies would perform when played against human subjects that are strategically responding.

In this part of my thesis, I introduce the term **predictive reliability**. Informally defined, **predictive reliability** refers to the percentage of strong correlations between a model's prediction accuracy and the performance of that model's corresponding defender strategy (i.e., the intervention). I then conducted a human subject experiment where players would play the role of the

attacker and attempt to successfully traverse the network without being caught by the defender. In order to simulate the fact that real-world data is limited, the amount of training data provided to the predictive models was intentionally limited. Finally, I conducted an empirical analysis of which models had the highest predictive reliability and also on the various factors that influence predictive reliability. The key takeaway from this portion of my thesis is that predictive reliability is never guaranteed; just because a model has the best predictive performance in a laboratory setting does not mean it will also perform the best in the field. This takeaway underscores the importance of conducting field tests to validate both predictive models and the interventions that are derived from these models.

1.3 Explanation and Visualization of Interventions

Another important problem in this space is enforcing compliance with environmental policies. For this scenario in my thesis, inspection agencies are tasked with deploying their limited resources to inspect leather tanneries that may not be complying with wastewater regulations. By providing randomized inspection schedules, agencies can ensure that they remain unpredictable and can catch as many violators as possible.

While my work builds on previously deployed Security Game solutions for counter-terrorism (Tambe, 2011) and traffic enforcement (Brown, Saisubramanian, Varakantham, & Tambe, 2014b), those works did not provide transparent explanations to users. Because users in this space are typically not familiar with game theory or other randomization techniques, opaque randomizations run the risk of not being adopted by users.

To address both the randomization and user transparency concerns, I introduced NECTAR (**N**irikshana for **E**nforcing **C**ompliance for **T**oxic wastewater **A**batement and **R**eduction), a game-theoretic inspection strategy application. In addition to generating randomized inspection strategies, NECTAR provides both strategy visualizations on Google Earth and a novel Security Game explanation component designed to explain the strategic interactions between their inspection strategy and potentially violating tanneries. To evaluate the model, I conducted an empirical evaluation where I created a real-world network of leather tanneries in Kanpur, India and evaluated the solution quality of NECTAR and other baselines.

1.4 Thesis Overview

My thesis is structured as follows: Chapter 2 discusses related work to provide context for the contributions in this thesis, Chapter 3 discusses the INTERCEPT poaching behavior model and corresponding field test, Chapter 4 introduces a hybrid decision tree model and a large-scale field test, Chapter 5 provides a detailed analysis of how the proposed poaching models would react to patrol generation algorithms, Chapter 6 focuses on predictive reliability and demonstrates the need for real-world field tests, Chapter 7 presents NECTAR, a randomized inspection generation application that also provides transparency to users, and finally Chapter 8 summarizes the contributions of this thesis.

Chapter 2

Related Work

2.1 Wildlife Crime Prevention: Predictive and Prescriptive Analytics

2.1.1 Predictive Modeling

Models inspired by previous work in behavioral game theory (McFadden, 1973; Palfrey & McKelvey, 1995; Costa-Gomes, Crawford, & Broseta, 2001; Stahl & Wilson, 1994) have been extensively used in recent years to predict human behavior in simultaneous-move games (Wright & Leyton-Brown, 2010, 2012, 2014) and also to predict adversary behavior in multiple security game domains including counter-terrorism (Nguyen et al., 2013), wildlife crime (Yang et al., 2014; Kar et al., 2015; Nguyen et al., 2016), fisheries protection (Haskell, Kar, Fang, Tambe, Cheung, & Denicola, 2014; Brown, Haskell, & Tambe, 2014a), and even in urban crime (Zhang, Sinha, & Tambe, 2015; Abbasi et al., 2015; Zhang, Bucarey, Mukhopadhyay, Sinha, Qian, Vorobeychik, & Tambe, 2016). Furthermore, researchers in the conservation community have used two-layered behavioral models similar to CAPTURE, the previous state-of-the-art, to predict future poaching behavior (Critchlow et al., 2016). CAPTURE was only the latest model in a long chain

of behavioral models used for human behavior prediction in game theory and also in the conservation literature. However, as detailed in Chapter 3, CAPTURE suffered from several limitations and performed poorly in predicting attacks in the real-world wildlife crime dataset.

Modeling and predicting other agents' behavior has also been studied in application domains such as RoboCup and military operations (Leottau, Ruiz-del Solar, MacAlpine, & Stone, 2015; Sukthankar, Goldman, Geib, Pynadath, & Bui, 2014), but such predictions are often based on real-time information, which is not available in this particular problem and dataset. There have been other attempts to predict poacher behavior in machine learning research: (Park, Serra, & Subrahmanian, 2015b) use association rule mining to get a single rule that classifies locations with a poaching attack, but the expressiveness of this approach is limited due to the single rule; (Park, Serra, Snitch, & Subrahmanian, 2015a) use standard classification algorithms to predict the attackability of targets and a regression model to predict attack probability. However, this work only reports accuracy, which is not an informative metric given the extreme class imbalance present in real-world wildlife crime datasets (e.g., just predicting no attacks everywhere could lead to high accuracy) and the potentially high cost of false negatives (e.g., an endangered animal may be poached). Moreover, our decision tree-based model can be seen as a generalization of this work since we can view a set of rules (instead of just one) that describe the model in richer terms than a single rule.

Spatio-temporal models have been used for prediction tasks in image and video processing. Markov Random Fields (MRF) were used by (Solberg, Taxt, & Jain, 1996; Yin & Collins, 2007) to capture spatio-temporal dependencies in remotely sensed data and moving object detection, respectively. In that work, each pixel influenced neighboring pixels spatially and temporally in

video sequences. Also, (Zhang, Brady, & Smith, 2001) used hidden MRF models for segmentation of brain magnetic resonance images. They obtained an accurate and robust segmentation by encoding spatial information through modeling the mutual influences of neighboring sites.

Critchlow et al. (Critchlow et al., 2015) analyzed spatio-temporal patterns in illegal activity in Uganda's Queen Elizabeth Protected Area (QEPA) using Bayesian hierarchical models. With real-world data, they demonstrated the importance of considering the spatial and temporal changes that occur in illegal activities. However, in this work and other similar works with spatio-temporal models (Rashidi, Wang, Skidmore, Mehdipoor, Darvishzadeh, Ngene, Vrieling, & Toxopeus, 2016; Rashidi et al., 2015), no standard metrics were provided to evaluate the models' predictive performance (e.g., precision, recall). As such, it is impossible to compare our predictive models' performance to theirs. While (Critchlow et al., 2016) was a field test of (Critchlow et al., 2015)'s work, (Rashidi et al., 2016, 2015) did not conduct field tests to validate their predictions in the real-world.

2.1.2 Prescriptive Analytics and Field Testing

There have been recent efforts on planning effective patrol strategies to combat poaching (Fang, Stone, & Tambe, 2015; Fang, Nguyen, Pickles, Lam, Clements, An, Singh, Tambe, & Lemieux, 2016), which have led to a project, PAWS, being deployed in the field. Previously, the focus of PAWS has been on generating risk-based randomized patrols and not on predicting poacher attacks. INTERCEPT, our contribution, provides predictive analysis that is essential to efficiently allocating limited ranger patrolling resources and can thus be the driving force for further prescriptive analysis (i.e., patrol planning). Additionally, the deployment of our work in the field has

shown a level of success that has not been previously seen in PAWS. As such, INTERCEPT is now part of the overall PAWS project as a predictive analytics module.

It is vital to validate predictive models in the real world, and in addition to our work, (Critchlow et al., 2016) has also conducted field tests in QEPA. (Critchlow et al., 2016) conducted a controlled experiment where their goal, by selecting three areas for rangers to patrol, was to maximize the number of observations sighted per kilometer walked by the rangers. Their test successfully demonstrated a significant increase in illegal activity detection at two of the areas, but they did not provide comparable evaluation metrics for their predictive model. Also, our field test was much larger in scale, involving 27 patrol posts compared to their 9 posts.

Finally, inspection games (Avenhaus, von Stengel, & Zamir, 2002) have been proposed for various inspection-related problems such as arms control compliance and environmental regulation compliance. In these games, an inspector verifies that the inspectee is adhering to a set of legal rules. While this framework has not yet been applied to the wildlife poaching domain, such a framework could be created where the ranger (defender) is the inspector and the poachers (attacker) are the inspectees. In this situation, the objective of predictive models would be to successfully predict where and when poachers will "violate" (e.g., place snares in the park) so that the inspection game framework can successfully devise an equilibrium strategy to minimize the amount of violations.

2.2 Network Security Games

Human bounded rationality has received considerable attention in Security Game research (Nguyen et al., 2013; Cui & John, 2014; Kar et al., 2015; Abbasi et al., 2015). The goal of these works

was to accurately model human decision making such that it could be harnessed to generate defender strategies that lead to higher expected utilities for the defender. For the developed models and corresponding defender mixed strategies, some of these works conducted human subject experiments to validate the quality of their models (Nguyen et al., 2013; Kar et al., 2015; Abbasi et al., 2015). Often in this research, different models' prediction accuracies are tested against human subjects, and the one that is most accurate is then used to generate defender strategies against human subjects (Nguyen et al., 2013; Kar et al., 2015). However, these works do not evaluate whether or not the other models' prediction accuracies correlated with their actual performance (i.e., predictive reliability). In other words, prediction accuracy is used as a proxy for the defender's actual performance, but it has not been well established that this is a reasonable proxy to use. In order to evaluate predictive reliability for SSGs, we obtained the human subject experiment data from Nguyen et al. (Nguyen et al., 2013) and evaluated predictive reliability on this data between the Quantal Response (QR) and Subjective Utility Quantal Response (SUQR) models.

As yet another type of Security Game, Network Security Game (NSG) research covers a wide variety of applications and domains. NSGs have been applied to curbing the illegal smuggling of nuclear material (Morton, Feng, & J., 2007), protecting maritime assets such as ports and ferries (Shieh et al., 2012), studying ways to minimize road network disruptions (Bell, U., D., & A., 2008), deterring fare evasion in public transit systems (Correa, Harks, Kreuzen, & Matuschke, 2014), and the assignment of checkpoints to urban road networks (Tsai, Yin, Kwak, Kempe, Kiekintveld, & Tambe, 2010; Jain, Conitzer, & Tambe, 2013). Although our NSG models most closely resemble the model used by Jain et al. (Jain, Korzhyk, Vanek, Conitzer, Pechoucek, &

Tambe, 2011; Jain et al., 2013), the primary difference is that we are not limited to modeling perfectly rational attackers.

In most NSG research, there is a basic assumption that the attacker is perfectly rational, but as demonstrated in work in behavioral game theory by Camerer et al., humans do not behave with perfect rationality (Camerer, 2003). Gutfraind et al. (Gutfraind, Hagberg, & Pan, 2009) addressed one type of boundedly rational adversary, an unreactive Markovian evader, in their work. Even though the evader (i.e., attacker) is unreactive to the defender's actions, the relaxation of the rational adversary assumption still results in an NP-hard problem. Positing that humans will rely on heuristics due to the complex nature of solving an NSG, Yang et al. (Yang, Fang, Jiang, Rajagopal, Tambe, & Maheswaran, 2012) addressed bounded rationality in a non-zero sum NSG setting by modeling the adversary's stochastic decision making with the Quantal Response (QR) model and various heuristic-based variants of the QR model. While they demonstrated that attacker behavior is better captured with human behavior models, their work is limited to using one defender resource in generating defender strategies and only focused on much smaller networks. In order to adequately defend larger networks, like those modeled in previous work by Jain et al. (Jain et al., 2011) and the ones presented in this work, multiple defender resources are required. For the behavior models we present, multiple defender resources are supported in a zero-sum setting.

2.3 Environmental Policy Compliance: Inspection Games

Several theoretical papers have used game theory to model the impact of environmental policies. Environmental games, such as those in (Tapiero, 2005), can use Stackelberg Games to model

interactions between a regulator and a polluting firm, while (Dong, Li, Li, Wang, & Huang, 2010) used game theory to study the effect of environmental policies in the Chinese electroplating industry.

Inspection games consider the general problem of scheduling inspections and have been extensively studied in the literature. For example, (Filar et al., 1985) modeled cases where an inspector must travel to multiple sites and determine violations as a stochastic game. A general theory of inspection games for problems such as arms control and environmental policy enforcement has been studied in (Avenhaus et al., 2002), including analysis of whether inspectors can benefit from acting first. (von Stengel, 2014) also considered inspection games with sequential inspections, including compact recursive descriptions of these games. However, most of these works did not focus on concrete applications and thus, unlike our work, did not provide executable inspection schedules to inspectors.

Other areas of research have considered various models of patrolling strategies and scheduling constraints. These include patrolling games (Alpern, Morton, & Papadaki, 2011; Bošanskỳ, Lisỳ, Jakob, & Pěchouček, 2011; Basilico, Gatti, & Amigoni, 2012) and security games with varying forms of scheduling constraints on resources (Yin, Jiang, Johnson, Tambe, Kiekintveld, Leyton-Brown, Sandholm, & Sullivan, 2012; Jain et al., 2013; Brown et al., 2014b). There has also been recent work on utilizing MDPs to represent strategies in security games (Shieh, Jiang, Yadav, Varakantham, & Tambe, 2014; Bosansky, Jiang, Tambe, & Kiekintveld, 2015). However, none of these efforts have focused on environmental inspections and have not investigated topics important in this domain, such as the impact of fine structures on adversary behavior (i.e., compliance).

Chapter 3

Cloudy with a Chance of Poaching: Adversary Behavior Modeling and Forecasting with Real-World Poaching Data[1]

Given the magnitude of the wildlife poaching problem and the difficulty of the patrol planning problem, patrol managers can benefit from tools that analyze data and generate forecasts of poacher attacks. In working with real-world wildlife crime data, we illustrated the importance of research driven by data from the field and real-world trials. This work potentially introduces a paradigm shift in showing how adversary modeling ought to be done for deployed security games (Shieh et al., 2012; Delle Fave, Jiang, Yin, Zhang, Tambe, Kraus, & Sullivan, 2014), particularly in domains such as green security games (Fang et al., 2015; Kar et al., 2015; Nguyen, Delle Fave, Kar, Lakshminarayanan, Yadav, Tambe, Agmon, Plumptre, Driciru, Wanyama, et al., 2015; Mc Carthy, Tambe, Kiekintveld, Gore, & Killion, 2016), where data is sparse compared to settings such as urban crime (Zhang, Jiang, Short, Brantingham, & Tambe, 2014; Abbasi et al., 2015). Security games have received significant attention at multiagent systems conferences (Korzhyk, Conitzer, & Parr, 2011; Kiekintveld, Islam, & Kreinovich, 2013; Munoz de Cote, Stranders, Basilico, Gatti, & Jennings, 2013; Basilico & Gatti, 2014; Kar et al., 2015), and past

[1]The work in this chapter was a joint first authorship with Debarun Kar.

work in security games has often focused on behavioral models that are learned from and tested in human subject experiments in the laboratory, which provides a large amount of attacker choice data over a small number of targets (Yang et al., 2011; Nguyen et al., 2013; Kar et al., 2015). The Quantal Response model is one example that models boundedly rational attackers' choices as a probability distribution via a Logit function (Yang et al., 2011). However, the wildlife crime domain introduces a set of real-world challenges (e.g., rangers collect limited, noisy data over a large number of targets with rich target features) that require behavior modeling efforts to not only focus more on real-world data and less on laboratory data, but also not rely on plentiful attack data.

Outperforming previous laboratory-developed models (Yang et al., 2011; Nguyen et al., 2013), CAPTURE (Nguyen et al., 2016) was a two-layered model, developed using real-world wildlife poaching data, that incorporated key insights and addressed the challenges present in wildlife crime data. CAPTURE's top layer attempted to predict the "attackability" of different targets, essentially providing predictions of poacher attacks. The bottom observation layer predicted how likely an attack that has occurred would be observed given the amount of patroller coverage (also known as effort). CAPTURE modeled the attackability layer as a hidden layer and used the Expectation Maximization (EM) algorithm to learn parameters for both layers simultaneously. Moreover, CAPTURE also contained a Dynamic Bayesian Network, allowing it to model attacker behavior as being temporally dependent on past attacks. The CAPTURE model, the previous state-of-the-art in the wildlife crime domain, represented a level of complexity not previously seen in behavior modeling in the security game literature.

While the focus of CAPTURE was on the observation layer's performance (i.e., "Where will patrollers observe past poaching attacks given their patrol effort?"), our focus was on forecasting

where future attacks will happen and thus we were interested in the attackability layer's predictions and performance (e.g., "Where will poachers attack next?"). However, CAPTURE's attackability predictions would sometimes predict too many targets to be attacked with a high probability and would thus have poor performance, as discussed in more detail later in this chapter. Given that CAPTURE embodied the latest in modeling adversary behavior in this domain, our first attempt focused on three different enhancements to CAPTURE: replacement of the observation layer with a simpler layer adapted from (Critchlow et al., 2015) (CAPTURE-LB), modeling attacker behavior as being dependent on the defender's historical coverage in the previous time step (CAPTURE-PCov), and finally, exponentially penalizing inaccessible areas (CAPTURE-DKHO). Unfortunately, all of these attempts ended in failure.

While poor performance was already a significant challenge, there were two additional, important shortcomings of CAPTURE and other complex models in this same family. First, CAPTURE's learning process took hours to complete on a high-performance computing cluster — unacceptable for rangers in Uganda with limited computing power. Second, CAPTURE's learned model was difficult to interpret for domain experts since it made predictions based on a linear combination of decision factors; the values of all its parameters' feature weights (i.e., 10 weights and a free parameter for the attack layer) needed to be simultaneously accounted for in a single interpretation of poacher preferences. These limitations and CAPTURE's poor performance drove us to seek an alternative modeling approach.

This chapter presents INTERCEPT (INTERpretable Classification Ensemble to Protect Threatened species), a new adversary behavior modeling application, and its three major contributions. (1) Given the limitations of traditional approaches in adversary behavior modeling, INTERCEPT

took a fundamentally different modeling approach, decision trees, and delivered a surprising result: although decision trees were simpler and did not take temporal correlations into account, they performed significantly better than CAPTURE (a complex model that considered temporal relationships), its variants, and other popular machine learning models (e.g., Logistic Regression, SVMs, and AdaBoost). Furthermore, decision trees satisfied the fundamental requirement of fast execution; given rangers' limited computing resources, models need to be capable of being quickly fitted using limited computing power. However, decision trees did not take into account the spatial correlations present in this dataset, and we introduced a spatially-aware decision tree algorithm, BoostIT, that significantly improved recall with only modest losses in precision. To further augment INTERCEPT's performance, we constructed an ensemble of the best classifiers which boosted predictive performance to a factor of 3.5 over the existing CAPTURE model. (2) These surprising results raised a fundamental question about the future of complex behavioral models (e.g., Quantal Response-based security game models (Yang et al., 2011; Nguyen et al., 2013, 2016)) in real-world applications. To underline the importance of this question, we conducted the most extensive empirical evaluation, at the time of publication, of the Queen Elizabeth Protected Area (QEPA) dataset with an analysis of 41 different models and a total of 193 model variants (e.g., different cost matrices) and demonstrated INTERCEPT's superior performance to traditional modeling approaches. (3) As a first for adversary behavior modeling applications applied to the wildlife crime domain, we present the results of a *month long* real-world deployment of INTERCEPT: compared to historical observation rates of illegal activity, rangers that used INTERCEPT observed 10 times the number of findings than the average. In addition to many signs of trespassing, rangers found a poached elephant, a roll of elephant snares, and a cache of 10

Figure 3.1: Campfire ashes and snare found by rangers directed by INTERCEPT. Photo credit: Uganda Wildlife Authority ranger

antelope snares before they were deployed (pictures in Figure 3.1). Each confiscated snare potentially represents an animal's life saved; while the rangers' finding of a poached elephant carcass was a grim reminder that poachers were active, these successful snare confiscations demonstrated the importance of real-world data in developing and evaluating adversary behavior models.

3.1 Dataset

The following discussion is on wildlife crime data from Uganda's Queen Elizabeth Protected Area (QEPA) (Figure 3.2), an area containing a wildlife conservation park and two wildlife reserves, which spans about 2,520 square kilometers. There are 37 patrol posts situated across QEPA from which Uganda Wildlife Authority (UWA) rangers conduct patrols to apprehend poachers, remove any snares or traps, monitor wildlife, and record signs

Figure 3.2: Queen Elizabeth Protected Area

of illegal activity. Along with the amount of patrolling effort in each area, the dataset for this work contained 13 years (2003-2015) of the type, location, and date of wildlife crime activities.

3.1.1 Dataset Challenges

Because this is a real-world geospatial crime dataset, it is important to understand the inherent challenges in analyzing its contents, such as nonlinear relationships between features (Kanevski, Pozdnoukhov, & Timonin, 2008). Additionally, data can only be collected in areas that are patrolled, and even in the areas that are patrolled, poaching signs may remain undetected. This occurs because poaching signs (such as snares) are often well-hidden, and rangers may need to conduct a thorough patrol in order to detect any attack – an infeasible task to undertake for all targets all the time due to limited patrolling resources. This real-world constraint not only leads to uncertainty in the negative class labels (i.e., when poaching signs are not observed we are uncertain whether an attack actually happened at the corresponding target or not) but also results in a small number of positive samples being recorded in the dataset thus creating a huge class imbalance. As such, it is necessary to evaluate the attack prediction model's performance with metrics that account for this uncertainty, such as those for Positive and Unlabeled Learning (PU Learning) (Lee & Liu, 2003), and are discussed in more detail in the following sections.

3.1.2 Dataset Composition

The entire QEPA area was discretized into 1 square kilometer grid cells (total 2,522 cells), each as a potential target of poaching. For each target, the ranger patrol effort level (i.e., coverage) and observed illegal human activity signs (e.g., poached animal carcasses, snares) were recorded. In addition, each target is associated with a non-static average ranger patrol effort value and a set

of static features (that are constant throughout the entire time period): terrain features such as habitat (the terrain type and relative ease of travel) and terrain slope; distances to nearby roads, water bodies, patrol posts, and villages; and animal density.

For the following analysis, we examine poaching data from 2003-2015. We aim to find the targets that are liable to be attacked since predicting the attackability of targets can guide future patrols. We assume a target is attackable if an attack is ever observed at that target at any point in time. Therefore, when creating training sets, we combine observations from the entire training period for each target and label it as attackable if any observations were made.

Given the uncertainty in negative labels, there are bound to be training and testing samples that contradict one another. We consider a sample in the training set and a sample in the testing set to be contradictory when they have the same combination of static domain features values (e.g., terrain, distances, animal density) and non-static patrol coverage amount (i.e., low or high coverage) but different class labels (attacked or not attacked). These contradictions introduce additional noise in evaluating the performance of learned models and would thus cause any model to perform poorly on said contradictory data. As such, we remove these contradictions, about 10% of the data, from testing sets.

3.2 CAPTURE and Proposed Variants

The natural first step towards predicting future poaching attacks based on our real-world wildlife crime dataset was to use the best previous model, CAPTURE (Nguyen et al., 2016). CAPTURE was shown to have superior predictive performance to a number of other standard models in the

behavioral game theory literature (e.g., Quantal Response (QR) (Yang et al., 2011), Subjective Utility Quantal Response (SUQR) (Nguyen et al., 2013)).

To make attackability predictions, we discretized the protected area into a set of targets I. Each target $i \in I$ has a set of domain-specific features $x_i \in x$ such as animal density d_i and distance to water. In a given time period t, a target i will be patrolled/covered by rangers with probability $c_{t,i}$.

CAPTURE consists of a two-layered behavior model. CAPTURE's first layer, the attackability layer, computes the probability that a poacher will attack a given target i at time step t. Similar to SUQR, which has been used to describe human players' stochastic choice of actions in security games, CAPTURE predicts attacks based on a linear combination of domain features $x_{t,i}$, ranger coverage probability $c_{t,i}$ at the current time step t, and whether the target was attacked in the previous time step $a_{t-1,i}$. With this last feature, $a_{t-1,i}$, CAPTURE models attacker behavior as being temporally dependent on past attacks.

$$p(a_{t,i} = 1 | a_{t-1,i}, c_{t,i}, x_{t,i}) = \frac{e^{\lambda^\intercal [a_{t-1,i}, c_{t,i}, x_{t,i}, 1]}}{1 + e^{\lambda^\intercal [a_{t-1,i}, c_{t,i}, x_{t,i}, 1]}} \tag{3.1}$$

λ is a parameter vector representing the importance of the features.

CAPTURE's second layer, the observation layer, computes the probability that rangers will observe an attack if poachers did attack that patrolled area based on a subset of domain features (e.g., habitat and slope) $\hat{x}_{t,i}$ and ranger coverage probability $c_{t,i}$.

$$p(o_{t,i} = 1 | a_{t,i} = 1, c_{t,i}, \hat{x}_{t,i}) = c_{t,i} \times \frac{e^{\omega^\intercal [\hat{x}_{t,i}, 1]}}{1 + e^{\omega^\intercal [\hat{x}_{t,i}, 1]}} \tag{3.2}$$

ω is a parameter vector that measures how domain features impact observation probability. The model parameters (λ, ω) that can maximize the likelihood of observations are estimated via the Expectation Maximization (EM) algorithm.

However, CAPTURE has a few limitations that lead to poor predictive performance in its *attackability layer*. First, CAPTURE's attackability predictions would sometimes predict too many targets to be attacked with a high probability (e.g., 80% of the targets will be attacked with almost 100% probability), leading to poor performance (see Section 3.5). One explanation is CAPTURE's parameter learning algorithm focuses on maximizing the performance of the observation layer rather than on the attackability layer. As the observation layer acts as a filter for the attackability layer, CAPTURE's learning process will converge to solutions that obtain decent performance for the observation layer even if the attackability layer's performance is poor.

Therefore, we propose several novel variants of CAPTURE as attempts to improve its predictions. In an attempt to restrict the degrees of freedom in the observation layer, and thus restrict the values the attackability layer can take in the learning process, we propose **CAPTURE-LB** which replaces the observation layer with a simpler observation layer, adapted from (Critchlow et al., 2015), described as follows:

$$p(o_{t,i} = 1 | a_{t,i} = 1, c_{t,i}) = 1 - e^{-\beta \times c_{t,i}} \tag{3.3}$$

where $\beta \in [0, 1]$ is the parameter that estimates the detection efficiency. This not only provides a straightforward way of assessing the performance of patrol effort to observations but also has a smaller chance of overfitting, due to fewer parameters.

For a given attack probability $p(a_{t,i} = 1)$, the unconditional probability of observing an attack at target i at time step t is given by:

$$p(o_{t,i}) = p(a_{t,i} = 1) \times p(o_{t,i} = 1 | a_{t,i} = 1, c_{t,i}) \tag{3.4}$$

Second, CAPTURE's attackability layer assumes that poachers plan attacks based on the patrol coverage in the current time step, which may not be realistic in the real world as the poachers may not get up-to-date information about the current patrol strategy and thus would rely on historical patrol coverage instead (Fang et al., 2015). Therefore, we propose another variant of CAPTURE, **CAPTURE-PCov**, that learns based on the previous time step's patrol coverage instead of the current time step's patrol coverage (Equation 3.5). Similarly, we propose **CAPTURE-PCov-LB**, a model that uses the attackability layer of CAPTURE with previous coverage as a feature but instead uses the LB observation layer defined in Equation 3.3.

$$p(a_{t,i} = 1 | a_{t-1,i}, c_{t-1,i}, x_{t,i}) = \frac{e^{\lambda^\intercal [a_{t-1,i}, c_{t-1,i}, x_{t,i}, 1]}}{1 + e^{\lambda^\intercal [a_{t-1,i}, c_{t-1,i}, x_{t,i}, 1]}} \tag{3.5}$$

Finally, CAPTURE's attackability predictions fail to take into account the domain knowledge that inaccessible and unattractive areas of the park will not be attacked with high probability, and we thus propose another variant **CAPTURE-DKHO**, which is the same as CAPTURE-PCov-LB except that it exponentially penalizes the attractiveness of inaccessible areas (Equation 3.6).

$$p(a_{t,i} = 1 | a_{t-1,i}, c_{t-1,i}, x_{t,i}) = \frac{e^{\lambda^\intercal [a_{t-1,i}, c_{t-1,i}, x'_{t,i}, 1]}}{1 + e^{\lambda^\intercal [a_{t-1,i}, c_{t-1,i}, x'_{t,i}, 1]}} \tag{3.6}$$

x' corresponds to the linear combination of features x but with the modified habitat feature $\sigma_i' = -\sigma_i e^{\sigma_i}$ which heavily penalizes high habitat values (i.e., hard to access areas).

3.3 INTERCEPT

The attempts of using the best previous model CAPTURE and the more complex variants of CAPTURE, proposed to address the limitations of CAPTURE, all suffered from poor attackability prediction performance as shown in Section 3.5. The natural progression then would have been to pursue more complex models in this behavioral game theory family of models with the expectation that they would improve performance on our real-world data. However, as reported in (Nguyen et al., 2016), complex models such as CAPTURE and its variants incur heavy computational costs; it takes approximately 6 hours for these models to complete execution. In addition, these models become more difficult to interpret when the dimensionality of the feature space increases (e.g., more numerical values to simultaneously account for in a single interpretation). We wanted to use models that would address all of these shortcomings by, not only significantly reducing computational costs so as to be usable by rangers with limited computing power in Uganda, but also remain interpretable to domain experts as the feature space dimensionality increases. All of these factors pointed against using more complex behavioral models. Therefore, we break from the current trend in behavior modeling in security games and model adversary behavior in terms of decision tree-based behavior models, even though we were initially skeptical about its predictive capabilities. This allowed us to not only express the nonlinear relationships between the geospatial features but also remain interpretable to domain experts as the feature

space dimensionality increases. Surprisingly, this simpler approach led to significant improvements in performance over the prior state-of-the-art (i.e., CAPTURE).

3.3.1 BoostIT

A binary decision tree D is trained on a set Θ of independent variables x (the domain features), a dependent variable o (attack observations), and outputs a binary classification D_i for each target i: $\{$not attacked ($D_i = 0$), attacked ($D_i = 1$)$\}$. A decision tree's negative predictions for a test set Ψ are denoted by $P^-_\Psi(D)$ and positive predictions by $P^+_\Psi(D)$ (i.e., vectors of binary predictions).

Crime hot spots are part of a well-known theory in Criminology (Eck et al., 2005) that views crime as an uneven distribution; crime is likely to be concentrated in particular areas called hot spots. If a particular geographic area has a high concentration of predicted attacks, it is reasonable to interpret these predictions as a hot spot prediction (i.e., predicting a high concentration of crime). While CAPTURE explicitly models attacks as a probability distribution decided by a linear combination of feature values and thus can implicitly represent the hot spots with soft boundaries in the geographic space, decision trees' rules with hard boundaries in the feature space would lead to fine-grained segmentations in the geographic space and is thus less capable of representing hot spots. As such, we designed the **Boost**ed decision tree with an **It**erative learning algorithm (henceforth referred to as BoostIT) (Algorithm 1), where proximity to a predicted hot spot is encoded as an additional input feature.

D^0 is the initial decision tree learned without the hot spot proximity feature h, and Θ^0 and Ψ^0 correspond to the initial training and test sets, respectively. For each level of iteration m, a feature h^Θ (and h^Ψ) is computed for each target $i \in I$ that corresponds to whether that target is close to a predicted hot spot in the training (and test sets); for example, if a target $i \in P_{\Theta^{m-1}}(D^{m-1})$ is

Algorithm 1 BoostIT

$D^0 \leftarrow$ LEARNDECISIONTREE(Θ^0)
repeat
  $h^\Theta \leftarrow$ CALCHOTSPOTPROXIMITY($P_{\Theta^{m-1}}(D^{m-1}), \alpha$)
  $h^\Psi \leftarrow$ CALCHOTSPOTPROXIMITY($P_{\Psi^{m-1}}(D^{m-1}), \alpha$)
  $\Theta^m \leftarrow$ ADDFEATURE(Θ^0, h_Θ)
  $\Psi^m \leftarrow$ ADDFEATURE(Ψ^0, h_Ψ)
  $D^m \leftarrow$ LEARNDECISIONTREE(Θ^m)
  $m = m + 1$
until iterationStoppingLevelReached
return P

adjacent to α or more targets in $P^+_{\Theta^{m-1}}(D^{m-1})$ (i.e., targets that are predicted to be positive), then $h^\Theta_i = 1$. We then re-learn the decision tree at each iteration m with a feature augmented dataset Θ^m. As an example, BoostIT may add a feature to a target i that i is near a hot spot if there are two adjacent targets that are predicted to be attackable. In the next iteration, this new feature ("near a hot spot") will get used in learning about predicting attacks on i. This continues until an iteration criterion is reached. Note that the test set Ψ is not used while learning new decision trees (only training data Θ is used) and is only used to update the test set prediction P_Ψ. In the rest of this chapter, we will refer to BoostIT with an α as BoostIT-αNearestNeighbors (or BoostIT-αNN). With this algorithm, the final decision tree D^m would generally predict more positive predictions with concentrated areas (i.e., hot spots) compared to D^0, but the set of predictions of D^m is not necessarily a superset of the set of predictions of D^0.

Although we are primarily interested in predicting attackability, we can also predict where patrollers would observe attacks by cascading attackability predictions with the LB observation layer (Equation 3.3). We convert the unconditional observation probability, derived from the cascaded model (Equation 3.4), to binary predictions by classifying samples as observed/not observed based on whether they are above or below the mean respectively.

3.3.2 INTERCEPT: Ensemble of Experts

We investigated the predictions of the traditional decision tree and BoostIT and observed that they are diverse in terms of their predictions. Here, by diversity, we mean that they predict attacks at a variety of targets. Therefore, while one model may fail to correctly classify a particular target as attacked, another model may succeed. This indicates the ability of different models to correctly learn and predict on different regions of the feature space. For example, let us consider the following three models: (i) DecisionTree, (ii) BoostIT-3NN and (iii) BoostIT-2NN. While computing pairwise disagreement between the models' attackability predictions, we observed that: (i) DecisionTree and BoostIT-3NN disagree on 105 out of 2211 target samples; (ii) DecisionTree and BoostIT-2NN disagree on 97 out of 2211 samples; and (iii) BoostIT-3NN and BoostIT-2NN disagree on 118 out of 2211 samples. This observation led us to consider combining the best decision tree and BoostIT-based models, thus forming INTERCEPT–an ensemble of experts.

Because of uncertainty in negative labels, INTERCEPT considers not only decision tree models with the standard false positive (FP) cost of one, but also decision trees with various FP costs. For a decision tree with FP cost of 0.6, during the learning process, the decision tree will not receive the full penalty of 1 but will instead receive a penalty of 0.6 for each false positive prediction it makes.

In INTERCEPT, each expert model voted for the final attack prediction on a particular target. We considered three types of voting rules to determine whether a target should be predicted to be attacked by the ensemble: (a) majority of the experts predict an attack; (b) all experts predict an attack; and (c) any one expert predicts an attack. INTERCEPT uses the best voting rule: majority.

We considered ensembles with three and five experts. Having at most 5 experts makes the ensemble easily interpretable. In other words, the final prediction at a target is due to only 5 decision rules at a maximum, and it is easy to walk the human domain experts through the 5 rules in a way that the logic is easily verified.

3.4 Evaluation Metrics

To evaluate INTERCEPT and other models, we first prepared two separate train/test splits on the dataset. For one dataset, we trained on data from 2003 to 2013 and evaluated our models on data in 2014, and for the other dataset, we trained on data from 2003 to 2014 and evaluated on data from 2015. Prior to discussing the evaluation results, we briefly discuss the metrics we use for computing our performance on predicting attackability and observed attacks.

Any metric to evaluate targets' *attackability* in domains such as wildlife poaching must account for the uncertainty in negative class labels. Therefore, in addition to standard metrics (Precision, Recall, and F1-score) that are used to evaluate models on datasets where there is no uncertainty in the underlying ground truth, we also evaluate our models with a metric that accounts for the uncertainty present in our dataset. The metric introduced in (Lee & Liu, 2003), henceforth referred to as L&L, is an appropriate metric since it is specifically designed for models learned on Positive and Unlabeled (PU) datasets (i.e., datasets with uncertain negative labels).

L&L is defined in Equation 3.7, where r denotes the recall and $Pr[f(Te) = 1]$ denotes the probability of a classifier f making a positive class label prediction. We compute $Pr[f(Te) = 1]$ as the percentage of positive predictions made by our model on a given test set.

$$L\&L(D, Te) = \frac{r^2}{Pr[f(Te) = 1]} \tag{3.7}$$

As we are certain about the positive samples in our dataset, L&L rewards a classifier more for correctly predicting where attacks have occurred (i.e., positive labels). However, it also prevents models from predicting attacks everywhere, via its denominator, and ensures that the model is selective in its positive predictions.

We also evaluate the models in terms of *observation* predictions. Here, we report standard metrics (Precision, Recall, and F1-score). We also compute the area under the Precision-Recall curve (PR-AUC). PR-AUC is a more appropriate metric for evaluating models on datasets with severe class imbalance (Davis & Goadrich, 2006) compared to area under the ROC curve. When there are many more negative points than positive points, the model can make many false positive predictions and the false positive rate would still be low, and thus, the ROC curve becomes less informative. In contrast, precision better captures how well the model is making correct positive predictions given a small number of positive examples. L&L is no longer used to evaluate the observation probability model as there is no uncertainty in terms of the observations, i.e., we either observed or did not observe an attack, and we are measuring the model's ability to predict whether we will observe attacks at already attacked targets.

3.5 Evaluation on Historical Real-World Patrol Data

To compare INTERCEPT with its competitors, we conducted a thorough investigation of the performance of 41 different models and 193 variants. A subset of the best performing models are detailed in the following evaluation.

This is one of the largest evaluation efforts on a real-world dataset in the wildlife crime domain, and we compared INTERCEPT against the previous best model CAPTURE, its variants, and other machine learning approaches such as Support Vector Machines (SVM), AdaBoosted Decision Trees, and Logistic Regression[2]. All the numbers highlighted in **bold** in the tables indicate the results of the best performing models in that table. The best performing INTERCEPT system is an ensemble of five decision trees with majority voting. The five decision trees are: a standard decision tree, two BoostIT decision trees ($m = 1$) with $\alpha = 2$ and $\alpha = 3$ respectively, and two decision trees with modified false positive costs 0.6 and 0.9 respectively. Note that, due to data collection methodology changes in 2015, the distribution of attack data in 2015 is significantly different than all other previous years; 2015 is a difficult dataset to test on when the training dataset of 2003-2014 represents a different distribution of attack data, and we will demonstrate this impact in the following evaluation.

3.5.1 Attackability Prediction Results

In Tables 3.1 and 3.2, we show a comparison of the performance between our best INTERCEPT system (the five decision tree ensemble with majority voting), the current state-of-the-art CAPTURE, its variants, and other baseline models towards accurately predicting the attackability of targets in QEPA for years 2014 and 2015, respectively.

[2]Note that due to data confidentiality agreements, we are unable to show an example decision tree in this chapter.

Classifier Type	F1	L&L	Precision	Recall
PositiveBaseline	0.06	1	0.03	1
UniformRandom	0.05	0.51	0.03	0.50
CAPTURE	0.31	3.52	0.25	0.39
CAPTURE-PCov	0.13	1.29	0.08	0.48
CAPTURE-PCov-LB	0.08	0.87	0.04	0.58
CAPTURE-DKHO	0.10	1.05	0.06	0.67
INTERCEPT	**0.41**	**5.83**	0.37	0.45

Table 3.1: Attackability prediction results on 2014 test data

Classifier Type	F1	L&L	Precision	Recall
PositiveBaseline	0.14	1	0.07	1
UniformRandom	0.19	0.50	0.11	0.50
CAPTURE	0.21	1.08	0.13	0.63
CAPTURE-PCov	0.19	0.87	0.11	0.57
CAPTURE-PCov-LB	0.18	0.69	0.11	0.46
CAPTURE-DKHO	0.20	0.71	0.12	0.5
INTERCEPT	**0.49**	**3.46**	0.63	0.41

Table 3.2: Attackability prediction results on 2015 test data

The PositiveBaseline corresponds to a model that predicts every target to be attacked ($p(a_{t,i}) = 1; \forall i, t$), and the UniformRandom corresponds to the baseline where each target is predicted to be attacked or not attacked with equal probability.

Note that, in this subsection, when evaluating two-layered models such as CAPTURE and its variants, we are examining the performance of just the attackability layer output, and we defer the evaluation of the observation predictions to Section 3.5.2. Since we evaluate the attackability predictions of our models on metrics for binary classification, the real-valued output of the attackability layer of CAPTURE and its variants were converted to a binary classification where probabilities greater than or equal to the mean attack probability were classified as positive.

We make the following observations from these tables: First, INTERCEPT completely outperforms the previous best model CAPTURE and its variants, as well as other baseline models in terms of L&L and F1 scores. For 2014, INTERCEPT outperforms CAPTURE in terms of

precision, recall, F1, and L&L score. For 2015 test data, INTERCEPT represents an even larger performance increase by approximately 3.50 times (L&L score of 3.46 vs 1.08) over CAPTURE and even more so for CAPTURE-PCov (L&L score of 3.46 vs 0.87). CAPTURE-PCov doesn't even outperform the positive baseline. Second, CAPTURE performs better on the 2014 dataset (when the training and testing data were similarly distributed) than on the 2015 dataset. In contrast, INTERCEPT remained flexible enough to perform well on the difficult 2015 testing set. However, CAPTURE-PCov, the more realistic variant of CAPTURE that can actually be used for forecasting, fails to make meaningful predictions about the attackability of targets. Its similar performance to PositiveBaseline demonstrates the need for models to learn the attackability of targets independently of observation probability to avoid learning models that make incorrect inferences about the attackability of the park (e.g., the entire park can be attacked). This is particularly important in the wildlife poaching domain because, due to the limited number of security resources, rangers cannot patrol every target all the time. Therefore, the attack probability model's predictions need to be extremely precise (high precision) while also being useful indicators of poaching activities throughout the park (high recall). Third, CAPTURE-PCov-LB performs even worse than CAPTURE-PCov in terms of L&L score for these attackability predictions, although the only difference between the two models is the observation layer. This occurs because the attackability prediction layer and the observation layer are not independent of one another; with the EM algorithm, the parameters are being learned for both layers simultaneously. In addition, by incorporating domain knowledge and penalizing the unattractive areas, CAPTURE-DKHO unfortunately does not lead to a significant improvement in performance. Fourth, INTERCEPT's

Classifier Type	F1	L&L	Precision	Recall
Weighted DecisionTree	0.11	1.01	0.06	0.48
SVM-BestFPCost-0.3	0.13	1.18	0.46	0.45
Logistic Regression	-	-	-	0
AdaBoostDecisionTree-BestFPCost-0.2	0.13	1.22	0.07	0.48
INTERCEPT	**0.41**	**5.83**	0.37	0.45

Table 3.3: Additional attackability prediction results on 2014 test data

precision values are significantly better compared to CAPTURE-PCov in 2014 and both CAPTURE and CAPTURE-PCov in 2015 with only modest losses of recall, indicating a significant reduction in the number of false positive predictions made throughout the park.

In Tables 3.3 and 3.4, we also compare INTERCEPT with other models including: (i) a decision tree where each sample was weighted based on the patrol intensity for the corresponding target (Weighted Decision Tree); (ii) the best performing SVM; (iii) Logistic Regression (which predicted no attacks and thus metrics could not be computed); and (iv) the best performing AdaBoosted Decision Tree. With respect to SVM, Logistic Regression and AdaBoosted Decision Tree, it is worthwhile to note the following: (i) learning SVMs with false positive costs (FPCost) greater than 0.45 resulted in a model that predicted "no attack" at all targets and so we report the test set performance of the SVM with the FPCost that performs the best on the training data; (ii) Logistic Regression fails to predict any attack on the test data for any FPCost and so it ends up with a recall of 0 (and thus other metrics cannot be computed); and (iii) the AdaBoosted Decision Tree result is reported on the test dataset for the FPCost that performs best on training data. These results show the significance of INTERCEPT in terms of successfully capturing attackability trends even in a highly imbalanced dataset and its corresponding superior performance.

Classifier Type	F1	L&L	Precision	Recall
Weighted DecisionTree	0.25	1.42	0.15	0.69
SVM-BestFPCost-0.25	0.19	0.72	0.12	0.43
Logistic Regression	-	-	-	0
AdaBoost-DT-BestFPCost-0.15	0.21	0.86	0.13	0.49
INTERCEPT	**0.49**	**3.46**	0.63	0.41

Table 3.4: Additional attackability prediction results on 2015 test data

3.5.2 Observation Prediction Results

In order to evaluate the performance of our models in terms of predicting whether we will *observe* attacks in the test set, we compute the probability of observing an attack using Equation 3.4 where the attack probability $p(a_{t,i} = 1)$ varies depending on the model used to learn attackability of targets. While CAPTURE and CAPTURE-PCov use the detection probability model specified in Equation 3.2, other models including a variant of CAPTURE called CAPTURE-PCov-LB use the detection model presented in (Critchlow et al., 2015) (i.e., Equation 3.2 is replaced by Equation 3.3).

Tables 3.5 and 3.6 correspond to how accurately each model predicted the observations in our test datasets. For a fair comparison, we also cascade the attackability predictions of the PositiveBaseline and UniformRandom baselines with an LB observation layer, and convert those unconditional observation probabilities to binary predictions with a mean threshold, as was done for CAPTURE's attackability predictions. We observe the following. First, incorporating the observation model in Equation 3.4 improved the PR-AUC score of CAPTURE in both test datasets (for 2014, 0.36 vs 0.33; for 2015, 0.32 vs 0.29). Second, INTERCEPT outperforms the other models by a large margin, both in terms of F1 and PR-AUC, for both test datasets. Combined with the attackability results, these results demonstrate the benefit of learning more precise attackability models in order to better predict observation probability.

Classifier Type	F1	Precision	Recall	PR-AUC
PositiveBaseline	0.13	0.07	0.79	0.12
UniformRandom	0.09	0.05	0.46	0.07
CAPTURE	0.14	0.08	0.73	0.33
CAPTURE-PCov	0.12	0.07	0.61	0.31
CAPTURE-PCov-LB	0.13	0.08	0.48	0.36
CAPTURE-DKHO	0.16	0.09	0.72	0.33
INTERCEPT	**0.36**	0.32	0.89	**0.45**

Table 3.5: Observation prediction results on 2014 test data

Classifier Type	F1	Precision	Recall	PR-AUC
PositiveBaseline	0.26	0.16	0.66	0.20
UniformRandom	0.19	0.12	0.45	0.14
CAPTURE	0.29	0.18	0.70	0.29
CAPTURE-PCov	0.29	0.18	0.70	0.29
CAPTURE-PCov-LB	0.34	0.21	0.85	0.32
CAPTURE-DKHO	0.36	0.24	0.79	0.32
INTERCEPT	**0.50**	0.65	0.41	**0.49**

Table 3.6: Observation prediction results on 2015 test data

3.5.3 Impact of Ensemble and Voting Rules

INTERCEPT consists of five experts with a majority voting rule. We now investigate the impact of combining different decision trees into an ensemble, and the impact of different voting rules. Tables 3.7 and 3.8 show that constructing an ensemble, INTERCEPT, significantly improves the performance of the system as a whole, compared to the performance of its individual decision tree and BoostIT members. The standard decision tree is more conservative as it predicts less false positives, leading to higher precision, but suffers from low recall.

Table 3.9 shows the impact that a voting rule has on performance on 2015 test data. We evaluate the performances of the best ensemble compositions, with three and five experts for each voting rule. We observe that: (i) Ensembles which predict an attack if any one expert predicts an attack (*Any*) are significantly better in terms of recall (0.68), but do poorly in terms

of precision (0.23). This is because such ensembles are more generous in terms of predicting an attack, and this leads to a significantly higher number of false positives; (ii) Ensembles with a voting rule where all experts have to agree (*All*) perform worse in terms of recall (0.16), but do best in terms of precision (0.89) as it makes less positive predictions (both true positives as well as false positives). This would mean that it would miss a lot of attacks in our domain, however; (iii) The majority voting-based ensembles (*Maj*), used by INTERCEPT, provide an important balance between precision (0.63) and recall (0.41) as they are neither extremely conservative nor generous in terms of their predictions and therefore outperform other voting rules significantly (L&L of 3.46).

This analysis provides important guidance for selecting ensembles depending on the requirements of the domain. For example, if it is extremely crucial to predict as many true positives as possible and a high number of false positives is acceptable, then using an *Any* voting method would be beneficial. However, in our wildlife poaching prediction problem, we have limited security resources and therefore cannot send patrols to every target all the time. Therefore, we not only wish to limit the number of false positives but also increase the number of correct poaching predictions. The majority voting rule provides this important balance in our domain.

Classifier Type	F1	L&L	Precision	Recall
PositiveBaseline	0.06	1	0.03	1
DecisionTree	0.2	1.8	0.14	0.36
BoostIT-1NN	0.19	2.23	0.12	0.55
BoostIT-2NN	0.21	2.13	0.13	0.45
BoostIT-3NN	0.2	2.01	0.13	0.45
INTERCEPT	**0.41**	**5.83**	0.37	0.45

Table 3.7: Attackability prediction results for decision tree models on 2014 test data

Classifier Type	F1	L&L	Precision	Recall
PositiveBaseline	0.14	1	0.07	1
DecisionTree	0.39	2.01	0.39	0.38
BoostIT-1NN	0.39	2.16	0.32	0.50
BoostIT-2NN	0.37	2.00	0.30	0.50
BoostIT-3NN	0.42	2.45	0.35	0.52
INTERCEPT	**0.49**	**3.46**	0.63	0.41

Table 3.8: Attackability prediction results for decision tree models on 2015 test data

Classifier Type	F1	L&L	Precision	Recall
BoostIT-3Experts-Any	0.36	2.11	0.26	0.59
BoostIT-5Experts-Any	0.34	2.13	0.23	0.68
BoostIT-3Experts-All	0.36	2.68	0.88	0.22
BoostIT-5Experts-All	0.28	1.97	0.89	0.16
BoostIT-3Experts-Maj	**0.49**	3.34	0.58	0.43
INTERCEPT	**0.49**	**3.46**	0.63	0.41

Table 3.9: Attackability prediction results for different ensembles on 2015 test data

3.6 Evaluation on Real-World Deployment

INTERCEPT represents a paradigm shift from complex logit-based models such as CAPTURE (Nguyen et al., 2016), and many others, to decision tree-based models. During development, we worked with a domain expert from the Wildlife Conservation Society to improve and validate our decision tree models and their corresponding predictions. Indeed, one advantage of shifting to a decision tree-based approach (as opposed to methods like CAPTURE) is that the underlying rules can be easily expressed to experts in non-AI fields.

After this development and evaluation on historical data was completed, we deployed INTERCEPT to the field. Based on INTERCEPT's predictions, we chose two patrol areas for QEPA rangers to patrol for one month. We selected these areas (approximately 9 square km each) such that they were (1) predicted to have multiple attacks and (2) previously infrequently patrolled as rangers did not previously consider these as important as other areas (and thus are good areas to

test our predictions). After providing the rangers with GPS coordinates of particular points in these areas, they patrolled these areas on foot and utilized their expert knowledge to determine where exactly in these areas they were most likely to find snares and other signs of illegal human activity (e.g., salt licks, watering holes). On each patrol, in addition to their other duties, rangers recorded their observations of animal sightings (i.e., 21 animals were sighted in one month) and illegal human activity.

We now present our key findings in Tables 3.10 and 3.11 and provide a selection of photos in Figures 3.1 and 3.3. The most noteworthy findings of these patrols are those related to elephant poaching; rangers, unfortunately, found one poached elephant with its tusks removed. However, this result demonstrates that poachers find this area, predicted by our model, attractive for poaching. On a more positive note, our model's predictions led rangers to find many snares before they caught any animals: one large roll of elephant snares, one active wire snare, and one cache of ten antelope snares. INTERCEPT's predictions assisted rangers' efforts in potentially saving the lives of *multiple animals including elephants*.

In addition to wildlife signs, which represent areas of interest to poachers and are detailed in Table 3.12, the findings of trespassing (e.g., litter, ashes) are significant as these represent areas of the park where humans were able to enter illegally and leave without being detected; if we can continue to patrol areas where poachers are visiting, rangers will eventually encounter the poachers themselves.

So as to provide additional context for these results, we present a set of base rates in Table 3.11. These base rates, computed in and around our proposed patrol areas, correspond to the average number of observed crimes per month from 2003-2015. Animal commercial (AnimalCom)

Week#	Illegal Activity	Count
2	Trespassing	19
3	Active Snares	1
	Plant Harvesting	1
4	Poached Elephants	1
	Elephant Snare Roll	1
	Antelope Snares	10
	Fish Roasting Racks	2

Table 3.10: Real-world patrol results: illegal activity

Figure 3.3: Elephant snare roll found by rangers directed by INTERCEPT. Photo credit: Uganda Wildlife Authority ranger

crimes correspond to elephant, buffalo, and hippopotamus poaching; animal noncommercial (AnimalNoncom) corresponds to all other poaching and poaching via snares; and plant noncommercial (PlantNoncom) corresponds to illegal harvesting of non-timber forest products (e.g., honey). The percentile rank corresponds to the number of months where our deployed patrols recorded more observations than in the historical data. For animal noncommercial crime, there was an average of 0.73 attacks observed monthly; for our deployed patrols, there were 3 separate observations (such as a roll of elephant snares), and in 91% of the months from 2003-2015, 2 or fewer observations were recorded.

Crime Type	**INTERCEPT**	Average	Percentile
AnimalCom	**1**	0.16	89%
AnimalNoncom	**3**	0.73	91%
Fishing	**1**	0.73	79%
PlantNoncom	**1**	0.46	76%
Trespassing	**19**	0.20	100%
Total	**25**	2.28	

Table 3.11: Base rate comparison: hits per month

Week#	# Wildlife Sighted
1	14
3	7

Table 3.12: Real-world patrol results: animal sightings

3.7 Lessons Learned

After our extensive modifications to the CAPTURE model and our subsequent evaluation, it is important to identify the reasons why we obtained such a surprising result: decision trees outperformed a complex, domain-specific temporal model. (1) The amount of data and its quality need to be taken into consideration when developing a model. The QEPA dataset had significant noise (e.g., imperfect observations) and extreme class imbalance. As such, attempting to develop a complex model for such a dataset can backfire when there does not exist sufficient data to support it. Our decision tree approach, generally regarded as simpler, benefits from being able to express non-linear relationships and can thus work with fewer data points. SVMs, also able to express non-linear relationships, appear to fail due to their complexity and attempt to define very fine-grained divisions of the dataset. (2) Model interpretability is a necessity when working in the real-world. Our decision tree model was deployed because, not only did it have superior performance to CAPTURE, but it was also easy to directly look at the rules the decision tree

had learned and evaluate whether or not those rules were reasonable (according to a domain expert). Thus, (3) the tradeoff between interpretability and performance, studied in domains where interpretability is key (e.g., biopharmaceutical classification) (Johansson, Sönströd, Norinder, & Boström, 2011), may not always exist. Indeed, the most interpretable model, out of all that we evaluated, was also the best performing (by a large margin!); future research should (i) not always forego interpretability in favor of performance under the assumption that there is always a tradeoff but (ii) instead be sure to investigate simpler, interpretable models in case there isn't a tradeoff.

Chapter 4

Taking it for a Test Drive: A Hybrid Spatio-Temporal Model for Wildlife Poaching Prediction Evaluated through a Controlled Field Test[1]

Wildlife poaching continues to be a global problem as key species are hunted toward extinction. For example, the latest African census showed a 30% decline in elephant populations between 2007 and 2014 (Census, 2016; on International Trade in Endangered Species of Wild Fauna & Flora, 2016). Wildlife conservation areas have been established to protect these species from poachers, and these areas are protected by park rangers. These areas are vast, and rangers do not have sufficient resources to patrol everywhere with high intensity and frequency.

At many sites now, rangers patrol and collect data related to snares they confiscate, poachers they arrest, and other observations. Given rangers' resource constraints, patrol managers could benefit from tools that analyze these data and provide future poaching predictions. However, this domain presents unique challenges. First, this domain's real-world data are few, extremely noisy, and incomplete. To illustrate, one of rangers' primary patrol goals is to find wire snares, which are deployed by poachers to catch animals. However, these snares are usually well-hidden

[1]The work in this chapter was a joint first authorship with Shahrzad Gholami.

(e.g., in dense grass), and thus rangers may not find these snares and (incorrectly) label an area as not having any snares. Second, poaching activity changes over time, and predictive models must account for this temporal component. Third, because poaching happens in the real world, there are mutual spatial and neighborhood effects that influence poaching activity. Finally, while field tests are crucial in determining a model's efficacy in the world, the difficulties involved in organizing and executing field tests often precludes them.

Previous works in this domain have modeled poaching behavior with real-world data. Based on data from a Queen Elizabeth Protected Area (QEPA) dataset,(Nguyen et al., 2016) introduced a two-layered temporal graphical model, CAPTURE, while (Kar, Ford, Gholami, Fang, Plumptre, Tambe, Driciru, Wanyama, Rwetsiba, Nsubaga, et al., 2017) constructed an ensemble of decision trees, INTERCEPT, that accounted for spatial relationships (detailed in Chapter 3). However, these works did not (1) account for both spatial and temporal components nor (2) validate their models via extensive field testing.

In this chapter, we provide the following contributions. (1) We introduce a new hybrid model that enhances an ensemble's broad predictive power with a spatio-temporal model's adaptive capabilities. Because spatio-temporal models require a lot of data, this model works in two stages. First, predictions are made with an ensemble of decision trees. Second, in areas where there are sufficient data, the ensemble's prediction is boosted via a spatio-temporal model. (2) In collaboration with the Wildlife Conservation Society and the Uganda Wildlife Authority, we designed and deployed a large, controlled experiment to QEPA. Across 27 areas we designated across QEPA, rangers patrolled approximately 498 kilometers over the course of eight months; to our knowledge, this is the largest controlled experiment and field test of machine learning-based

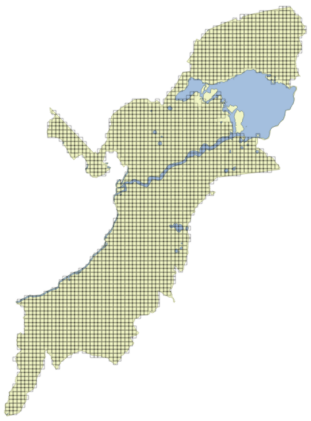
predictive models in this domain. In this experiment, we tested our model's selectiveness: is our model able to differentiate between areas of high and low poaching activity?

In experimental results, (1) we demonstrate our model's superior performance over the prior state-of-the-art (Kar et al., 2017) and thus the importance of spatio-temporal modeling. (2) During our field test, rangers found over three times more snaring activity in areas where we predicted higher poaching activity. When accounting for differences in ranger coverage, rangers found twelve times the number of findings per kilometer walked in those areas. Additionally, the differences between the areas of high predicted activity and low predicted activity are statistically significant ($t(497) = 4.09$, $p < .0001$). These results demonstrate that (i) our model is selective in its predictions and can predict both where poaching will frequently or infrequently occur and (ii) our model's superior predictive performance in the laboratory extends to the real world.

4.1 Dataset

This study's wildlife crime dataset is from Uganda's Queen Elizabeth Protected Area (QEPA), an area containing a wildlife conservation park and two wildlife reserves, which spans about 2,520 square kilometers (Figure 4.1a). There are 37 patrol posts situated across QEPA from which Uganda Wildlife Authority (UWA) rangers conduct patrols to apprehend poachers, remove any snares or traps (Figure 4.1b), monitor wildlife, and record signs of illegal activity. Along with the amount of patrolling effort in each area, the dataset contains 14 years (2003-2016) of the type, location, and date of wildlife crime activities.

Rangers lack the manpower to patrol everywhere all the time, and thus illegal activity may be undetected in unpatrolled areas. Patrolling is an imperfect process, and there is considerable



(a) QEPA grid

(b) Photo of a snare found on patrol. Photo credit: UWA ranger

uncertainty in the dataset's negative data points (i.e., areas being labeled as having no illegal activity); rangers may patrol an area and label it as having no snares when, in fact, a snare was well-hidden and undetected. These factors contribute to the dataset's already large class imbalance; there are many more negative data points than there are positive points (crime detected). It is thus necessary to consider models that estimate hidden variables (e.g., whether an area has been attacked) and also to evaluate predictive models with metrics that account for this uncertainty, such as those in the Positive and Unlabeled Learning (PU Learning) literature (Lee & Liu, 2003). We divide QEPA into 1 square kilometer grid cells (a total of 2,522 cells), and we refer to these cells as targets. Each target is associated with several static geospatial features such as terrain (e.g., slope), distance values (e.g., distance to border), and animal density. Each target is also associated with dynamic features such as how often an area has been patrolled (i.e., coverage) and observed illegal activities (e.g., snares).

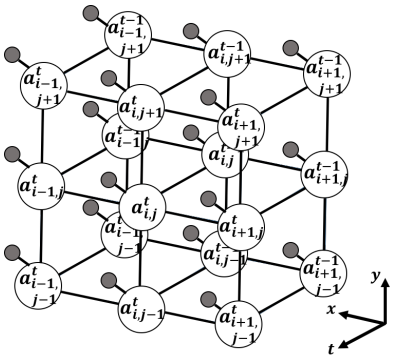


Figure 4.2: Graphical model

4.2 Models and Algorithms

4.2.1 Prediction by Graphical Models

4.2.1.1 Markov Random Field (MRF)

To predict poaching activity, each target, at time step $t \in \{t_1, ..., t_m\}$, is represented by coordinates i and j within the boundary of QEPA. In Figure 4.2, we demonstrate a three-dimensional network for spatio-temporal modeling of poaching events over all targets. Connections between nodes represent the mutual spatial influence of neighboring targets and also the temporal dependence between recurring poaching incidents at a target. $a^t_{i,j}$ represents poaching incidents at time step t and target i, j. Mutual spatial influences are modeled through first-order neighbors (i.e., $a^t_{i,j}$ connects to $a^t_{i\pm1,j}$, $a^t_{i,j\pm1}$ and $a^{t-1}_{i,j}$) and second-order neighbors (i.e., $a^t_{i,j}$ connects to $a^t_{i\pm1,j\pm1}$); for simplicity, the latter is not shown on the model's lattice. Each random variable takes a value in its state space, in this paper, $\mathcal{L} = \{0, 1\}$.

To avoid index overload, henceforth, nodes are indexed by serial numbers, $\mathcal{S} = \{1, 2, ..., N\}$ when we refer to the three-dimensional network. We introduce two random fields, indexed by $\mathcal{S}$, with their configurations: $\mathcal{A} = \{\boldsymbol{a} = (a_1, ..., a_N) | a_i \in \mathcal{L}, i \in \mathcal{S}\}$, which indicates an *actual*

poaching attack occurred at targets over the period of study, and $\mathcal{O} = \{\boldsymbol{o} = (o_1, ..., o_N) | o_i \in \mathcal{L}, i \in \mathcal{S}\}$ indicates a *detected* poaching attack at targets over the period of study. Due to the imperfect detection of poaching activities, the former represents the hidden variables, and the latter is the known observed data collected by rangers, shown by the gray-filled nodes in Figure 4.2. Targets are related to one another via a neighborhood system, $\mathcal{N}_n$, which is the set of nodes neighboring n and $n \notin \mathcal{N}_n$. This neighborhood system considers all spatial and temporal neighbors. We define neighborhood attackability as the fraction of neighbors that the model predicts to be attacked: $u_{\mathcal{N}_n} = \sum_{n \in \mathcal{N}_n} a_n / |\mathcal{N}_n|$.

The probability, $P(a_i | u_{\mathcal{N}_n}, \boldsymbol{\alpha})$, of a poaching incident at each target n at time step t is represented in Equation 4.1, where $\boldsymbol{\alpha}$ is a vector of parameters weighting the most important variables that influence poaching; $\boldsymbol{Z}$ represents the vector of time-invariant ecological covariates associated with each target (e.g., animal density, slope, forest cover, net primary productivity, distance from patrol post, town and rivers (Critchlow et al., 2015; O'Kelly, 2013)). The model's temporal dimension is reflected through not only the backward dependence of each a_n, which influences the computation of $u_{\mathcal{N}_n}$, but also in the past patrol coverage at target n, denoted by c_n^{t-1}, which models the delayed deterrence effect of patrolling efforts.

$$p(a_n = 1 | u_{\mathcal{N}_n}, \boldsymbol{\alpha}) = \frac{e^{-\boldsymbol{\alpha}[\boldsymbol{Z}, u_{\mathcal{N}_n}, c_n^{t-1}, 1]^\intercal}}{1 + e^{-\boldsymbol{\alpha}[\boldsymbol{Z}, u_{\mathcal{N}_n}, c_n^{t-1}, 1]^\intercal}} \tag{4.1}$$

Given a_n, o_n follows a conditional probability distribution proposed in Equation 4.2, which represents the probability of rangers detecting a poaching attack at target n. The first column of the matrix denotes the probability of not detecting or detecting attacks if an attack has not happened, which is constrained to 1 or 0 respectively. In other words, it is impossible to detect

an attack when an attack has not happened. The second column of the matrix represents the probability of not detecting or detecting attacks in the form of a logistic function if an attack has happened. Since it is less rational for poachers to place snares close to patrol posts and more convenient for rangers to detect poaching signs near the patrol posts, we assumed dp_n (distance from patrol post) and c_n^t (patrol coverage devoted to target n at time t) are the major variables influencing rangers' detection capabilities. Detectability at each target is represented in Equation 4.2, where $\boldsymbol{\beta}$ is a vector of parameters that weight these variables.

$$p(o_n|a_n) = \begin{bmatrix} p(o_n = 0|a_n = 0) & p(o_n = 0|a_n = 1, \boldsymbol{\beta}) \\ p(o_n = 1|a_n = 0) & p(o_n = 1|a_n = 1, \boldsymbol{\beta}) \end{bmatrix} = \begin{bmatrix} 1, & \frac{1}{1 + e^{-\boldsymbol{\beta}[dp_n, c_n^t, 1]^\intercal}} \\ 0, & \frac{e^{-\boldsymbol{\beta}[dp_n, c_n^t, 1]^\intercal}}{1 + e^{-\boldsymbol{\beta}[dp_n, c_n^t, 1]^\intercal}} \end{bmatrix} \tag{4.2}$$

We assume that $(\boldsymbol{o}, \boldsymbol{a})$ is pairwise independent, meaning $p(\boldsymbol{o}, \boldsymbol{a}) = \prod_{n \in \mathcal{S}} p(o_n, a_n)$.

4.2.1.2 EM Algorithm to Infer on MRF

We use the Expectation-Maximization (EM) algorithm (Bishop, 2006) to estimate the MRF model's parameters $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$. For completeness, we provide details about how we apply the EM algorithm to our model. Given a joint distribution $p(\boldsymbol{o}, \boldsymbol{a}|\boldsymbol{\theta})$ over observed variables $\boldsymbol{o}$ and hidden variables $\boldsymbol{a}$, governed by parameters $\boldsymbol{\theta}$, EM aims to maximize the likelihood function $p(\boldsymbol{o}|\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. To start the algorithm, an initial setting for the parameters $\boldsymbol{\theta}^{old}$ is chosen. At E-step, $p(\boldsymbol{a}|\boldsymbol{o}, \boldsymbol{\theta}^{old})$ is evaluated, particularly, for each node in MRF model:

$$p(a_n|o_n, \boldsymbol{\theta}^{old}) = \frac{p(o_n|a_n, \boldsymbol{\beta}^{old}).p(a_n|u_{\mathcal{N}_n}^{old}, \boldsymbol{\alpha}^{old})}{p(o_n)} \tag{4.3}$$

M-step calculates $\boldsymbol{\theta}^{new}$, according to the expectation of the complete log likelihood, $\log p(\boldsymbol{o}, \boldsymbol{a}|\boldsymbol{\theta})$, given in Equation 4.4.

$$\boldsymbol{\theta}^{new} = \arg\max_{\boldsymbol{\theta}} \sum_{a_n \in \mathcal{L}} p(\boldsymbol{a}|\boldsymbol{o}, \boldsymbol{\theta}^{old}). \log p(\boldsymbol{o}, \boldsymbol{a}|\boldsymbol{\theta}) \tag{4.4}$$

To facilitate calculation of the log of the joint probability distribution, $\log p(\boldsymbol{o}, \boldsymbol{a}|\boldsymbol{\theta})$, we introduce an approximation that makes use of $u^{old}_{\mathcal{N}_n}$, represented in Equation 4.5.

$$\log p(\boldsymbol{o}, \boldsymbol{a}|\boldsymbol{\theta}) = \sum_{n \in \mathcal{S}} \sum_{a_n \in \mathcal{L}} \log p(o_n|a_n, \boldsymbol{\beta}) + \log p(a_n|u^{old}_{\mathcal{N}_n}, \boldsymbol{\alpha}) \tag{4.5}$$

Then, if convergence of the log likelihood is not satisfied, $\boldsymbol{\theta}^{old} \leftarrow \boldsymbol{\theta}^{new}$, and repeat.

4.2.1.3 Dataset Preparation for MRF

To split the data into training and test sets, we divided the real-world dataset into year-long time steps. We trained the model's parameters $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$ on historical data sampled through time steps $(t_1, ..., t_m)$ for all targets within the boundary. These parameters were used to predict poaching activity at time step t_{m+1}, which represents the test set for evaluation purposes. The trade-off between adding years' data (performance) vs. computational costs led us to use three years ($m = 3$). The model was thus trained over targets that were patrolled throughout the training time period (t_1, t_2, t_3). We examined three training sets: 2011-2013, 2012-2014, and 2013-2015 for which the test sets are from 2014, 2015, and 2016, respectively.

Capturing temporal trends requires a sufficient amount of data to be collected regularly across time steps for each target. Due to the large amount of missing inspections and uncertainty in the collected data, this model focuses on learning poaching activity only over regions that have been

continually monitored in the past, according to Definition 1. We denote this subset of targets as $\mathcal{S}_c$.

Definition 1. ***Continually vs. occasionally monitoring:*** *A target i, j is continually monitored if all elements of the coverage sequence are positive; $c_{i,j}^{t_k} > 0, \forall k = 1, ..., m$ where m is the number of time steps. Otherwise, it is occasionally monitored.*

Experiments with MRF were conducted in various ways on each data set. We refer to a) a *global* model with spatial effects as **GLB-SP**, which consists of a single set of parameters $\boldsymbol{\theta}$ for the whole QEPA, and b) a *global* model without spatial effects (i.e., the parameter that corresponds to $u_{\mathcal{N}_n}$ is set to 0) as **GLB**. The spatio-temporal model is designed to account for temporal and spatial trends in poaching activities. However, since learning those trends and capturing spatial effects are impacted by the variance in local poachers' behaviors, we also examined c) a *geo-clustered* model which consists of multiple sets of local parameters throughout QEPA with spatial effects, referred to as **GCL-SP**, and also d) a *geo-clustered* model without spatial effects (i.e., the parameter that corresponds to $u_{\mathcal{N}_n}$ is set to 0) referred to as **GCL**.

Figure 4.3 shows the geo-clusters generated by Gaussian Mixture Models (GMM), which classifies the targets based on the geo-spatial features, $\boldsymbol{Z}$, along with the targets' coordinates, $(x_{i,j}, y_{i,j})$, into 22 clusters. The number of geo-clusters, 22, are intended to be close to the number of patrol posts in QEPA such that each cluster contains one or two nearby patrol posts. With that being considered, not only are local poachers' behaviors described by a distinct set of parameters, but also the data collection conditions, over the targets within each cluster, are maintained to be nearly uniform.

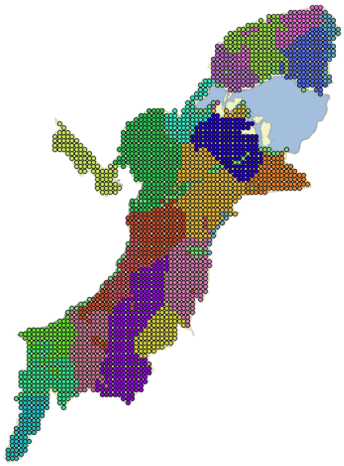


Figure 4.3: Geo-clusters

4.2.2 Prediction by Ensemble Models

A **Bagging ensemble model** or **B**ootstrap **agg**regation technique, called Bagging, is a type of ensemble learning which bags some weak learners, such as decision trees, on a dataset by generating many bootstrap duplicates of the dataset and learning decision trees on them. Each of the bootstrap duplicates are obtained by randomly choosing M observations out of M with replacement, where M denotes the training dataset size. Finally, the predicted response of the ensemble is computed by taking an average over predictions from its individual decision trees. To learn a Bagging ensemble, we used the *fitensemble* function of MATLAB 2017a. **Dataset preparation** for the Bagging ensemble model is designed to find the targets that are liable to be attacked (Kar et al., 2017). A target is assumed to be attackable if it has ever been attacked; if any observations occurred in the entire training period for a given target, that target is labeled as attackable. For this model, the best training period contained 5 years of data based on a performance analysis of using the least amount of training data.

4.2.3 Hybrid of MRF and Bagging Ensemble

Since the amount and regularity of data collected by rangers varies across regions of QEPA, predictive models perform differently in different regions. As such, we propose using different models to predict over them; first, we used a Bagging ensemble model, and then improved the predictions in some regions using the spatio-temporal model. For global models, we used MRF for all continually monitored targets. However, for geo-clustered models, for targets in the continually monitored subset, $\mathcal{S}_c^q$, (where temporally-aware models can be used practically), the MRF model's performance varied widely across geo-clusters according to our experiments. q indicates clusters and $1 \leq q \leq 22$. Thus, for each q, if the average Catch Per Unit Effort (CPUE), outlined by Definition 2, is relatively large, we use the MRF model for $\mathcal{S}_c^q$. In Conservation Biology, CPUE is an indirect measure of poaching activity abundance. A larger average CPUE for each cluster corresponds to more frequent poaching activity and thus more data for that cluster. Consequently, using more complex spatio-temporal models in those clusters becomes more reasonable.

Definition 2. ***Average CPUE*** *is* $\sum_{n \in \mathcal{S}_c^q} o_n / \sum_{n \in \mathcal{S}_c^q} c_n^t$ *in cluster* q*.*

To compute CPUE, effort corresponds to the amount of coverage (i.e., 1 unit = 1 km walked) in a given target, and catch corresponds to the number of observations. Hence, for $1 \leq q \leq 22$, we will boost selectively according to the average CPUE value; some clusters may not be boosted by MRF, and we would only use Bagging ensemble model for making predictions on them. Experiments on historical data show that selecting 15% of the geo-clusters with the highest average CPUE results in the best performance for the entire hybrid model (discussed in the following Evaluation Section).

4.3 Evaluations and Discussions

4.3.1 Evaluation Metrics

The imperfect detection of poaching activities in wildlife conservation areas leads to uncertainty in the negative class labels of data samples (Kar et al., 2017). It is thus vital to evaluate prediction results based on metrics which account for this inherent uncertainty. In addition to standard metrics in machine learning (e.g., precision, recall, F1) which are used to evaluate models on datasets with no uncertainty in the underlying ground truth, we also use the L&L metric introduced in (Lee & Liu, 2003), which is a metric specifically designed for models learned on Positive and Unlabeled datasets. L&L is defined as $L\&L = \frac{r^2}{Pr[f(Te)=1]}$, where r denotes the recall and $Pr[f(Te) = 1]$ denotes the probability of a classifier f making a positive class label prediction and is estimated by the percentage of positive predictions made by the model on a given test set.

4.3.2 Experiments with Real-World Data

Evaluation of models' attack predictions are demonstrated in Tables 4.1, 4.2, 4.3, and 4.4. To compare models' performances, we used several baseline methods, i) Positive Baseline, **PB**; a model that predicts poaching attacks to occur in all targets, ii) Random Baseline, **RB**; a model which flips a coin to decide its prediction, iii) Training Label Baseline, **TL**; a model which predicts a target as attacked if it has been ever attacked in the training data. We also present the results for Support Vector Machines, **SVM**, and AdaBoost methods, **AD**, which are well-known machine learning techniques, along with results for the best performing predictive model on the QEPA dataset, INTERCEPT, **INT**, (Kar et al., 2017). Results for the Bagging ensemble technique, **BG**, and RUSBoost, **RUS**, a hybrid sampling/boosting algorithm for learning from datasets with class

Test set	**2014**					**2015**				
Models	**PB**	**RB**	**TL**	**SVM**	**BG-G***	**PB**	**RB**	**TL**	**SVM**	**BG-G***
Precision	0.06	0.05	0.26	0.24	0.65	0.10	0.08	0.39	0.4	0.69
Recall	1.00	0.46	0.86	0.3	0.54	1.00	0.43	0.78	0.15	0.62
F1	0.10	0.09	0.4	0.27	0.59	0.18	0.14	0.52	0.22	0.65
L&L	1.00	0.43	4.09	1.33	6.44	1.00	0.37	3.05	0.62	4.32
Models	**RUS**	**AD**	**BG**	**INT**	**BG-G***	**RUS**	**AD**	**BG**	**INT**	**BG-G***
Precision	0.12	0.33	0.62	0.37	0.65	0.2	0.52	0.71	0.63	0.69
Recall	0.51	0.47	0.54	0.45	0.54	0.51	0.5	0.53	0.41	0.62
F1	0.19	0.39	0.58	0.41	0.59	0.29	0.51	0.61	0.49	0.65
L&L	1.12	2.86	6.18	5.83	6.44	1.03	2.61	3.83	3.46	4.32

Table 4.1: Comparing all models' performances with the best performing BG-G model (2014 and 2015)

Test set	**2016**				
Models	**PB**	**RB**	**TL**	**SVM**	**BG-G***
Precision	0.10	0.09	0.45	0.45	0.74
Recall	1.00	0.44	0.75	0.23	0.66
F1	0.18	0.14	0.56	0.30	0.69
L&L	1.00	0.38	3.4	1.03	4.88
Models	**RUS**	**AD**	**BG**	**INT**	**BG-G***
Precision	0.19	0.53	0.76	0.40	0.74
Recall	0.65	0.54	0.62	0.66	0.66
F1	0.29	0.53	0.68	0.51	0.69
L&L	1.25	2.84	4.75	2.23	4.88

Table 4.2: Comparing all models' performances with the best performing BG-G model (2016)

imbalance (Seiffert, Khoshgoftaar, Van Hulse, & Napolitano, 2010), are also presented. In all tables, **BG-G*** stands for the best performing model among all variations of the hybrid model, which will be discussed in detail later. Tables 4.1 and 4.2 demonstrate that **BG-G*** outperformed all other existing models in terms of L&L and also F1.

Tables 4.3 and 4.4 provide a detailed comparison of all variations of our hybrid models, **BG-G** (i.e., when different MRF models are used). When **GCL-SP** is used, we get the best performing model in terms of L&L score, which is denoted as **BG-G***. The poor results of

Test set	2014				2015			
MRF models	**GLB**	**GLB-SP**	**GCL**	**GCL-SP**	**GLB**	**GLB-SP**	**GCL**	**GCL-SP**
Precision	0.12	0.12	0.63	0.65	0.19	0.19	0.69	0.69
Recall	0.58	0.65	0.54	0.54	0.52	0.58	0.65	0.62
F1	0.20	0.20	0.58	0.59	0.28	0.29	0.65	0.65
L&L	1.28	1.44	6.31	6.44	0.99	1.14	4.32	4.32

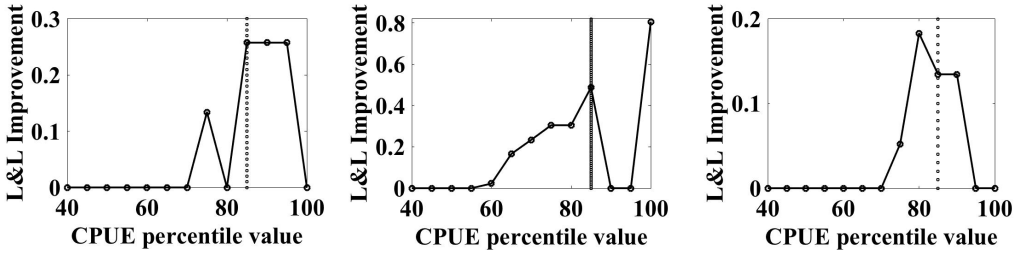
Table 4.3: Performances of hybrid models with variations of MRF (BG-G models), 2014 and 2015

Test set	2016			
MRF models	**GLB**	**GLB-SP**	**GCL**	**GCL-SP**
Precision	0.18	0.19	0.72	0.74
Recall	0.50	0.46	0.66	0.66
F1	0.27	0.27	0.69	0.69
L&L	0.91	0.91	4.79	4.88

Table 4.4: Performances of hybrid models with variations of MRF (BG-G models), 2016

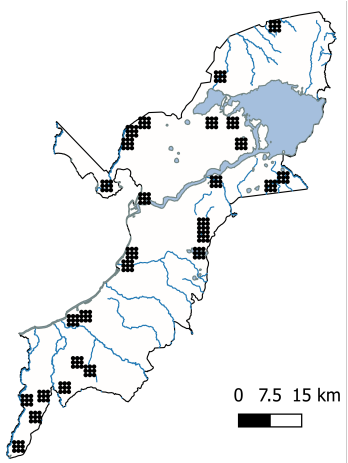
learning a global set of parameters emphasize the fact that poachers' behavior and patterns are not identical throughout QEPA and should be modeled accordingly.

Our experiments demonstrated that the performance of the MRF model within $\mathcal{S}_c^q$ varies across different geo-clusters and is related to the CPUE value for each cluster, q. Figure 4.4a displays an improvement in L&L score for the **BG-G*** model compared to **BG** vs. varying the percentile of geo-clusters used for boosting. Experiments with the 2014 test set show that choosing the 85^{th} percentile of geo-clusters for boosting with MRF, according to CPUE, (i.e., selecting 15% of the geo-clusters, with highest CPUE), results in the best prediction performance. The 85^{th} percentile is shown by vertical lines in Figures where the **BG-G*** model outperformed the **BG** model. We used a similar percentile value for conducting experiments with the MRF model on test sets of 2015 and 2016. Figure 4.4b and 4.4c confirm the efficiency of choosing an 85^{th} percentile value for those test sets, as well. Also, Tables 4.1 and 4.2 demonstrate that for

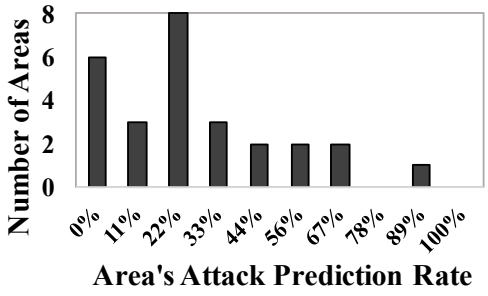


(a) 2014 values (b) 2015 values (c) 2016 values

Figure 4.4: L&L improvement vs. CPUE percentile value; BG-G* compared to BG



(a) Field test areas



(b) Patrol area statistics

Figure 4.5: Field test overview

BG-G* recall increased up to almost 10% for the 2015 test set which would result in marking roughly 10% more vulnerable targets as attacked and thus protecting more endangered animals.

4.4 QEPA Field Test

While our model demonstrated superior predictive performance on historical data, it is important to test these models in the field.

The initial field test we conducted in (Kar et al., 2017), in collaboration with the Wildlife Conservation Society (WCS) and the Uganda Wildlife Authority (UWA), was the first of its kind

in the machine learning (ML) community and showed promising improvements over previous patrolling regimes. Due to the difficulty of organizing such a field test, its implications were limited: only two 9-sq km areas (18 sq km) of QEPA were patrolled by rangers over a month. Because of its success, however, WCS and UWA graciously agreed to a larger scale, controlled experiment: also in 9 sq km areas, but rangers patrolled 27 of these areas (243 sq km, spread across QEPA) over eight months; this is the largest to date field test of ML-based predictive models in this domain. We show the areas in Figure 4.5a. Note that rangers patrolled these areas in addition to other areas of QEPA as part of their normal duties.

This experiment's goal was to determine the selectiveness of our model's snare attack predictions: does our model correctly predict both where there are and are not snare attacks? We define attack prediction rate as the proportion of targets (a 1 km by 1 km cell) in a patrol area (3 by 3 cells) that are predicted to be attacked. We considered two experiment groups that corresponded to our model's attack prediction rates from November 2016 – June 2017: High (group 1) and Low (group 2). Areas that had an attack prediction rate of 50% or greater were considered to be in a high area (group 1); areas with less than a 50% rate were in group 2. For example, if the model predicted five out of nine targets to be attacked in an area, that area was in group 1. Due to the importance of QEPA for elephant conservation, we do not show which areas belong to which experiment group in Figure 4.5a so that we do not provide data to ivory poachers. A three group analysis (i.e., high, medium, low) is presented in the Appendix.

To start, we exhaustively generated all patrol areas such that (1) each patrol area was 3x3 sq km, (2) no point in the patrol area was more than 5 km away from the nearest ranger patrol post, and (3) no patrol area was patrolled too frequently or infrequently in past years (to ensure that the training data associated with all areas was of similar quality); in all, 544 areas were generated

Experiment Group	Exhaustive Group Memberships	Final Group Memberships
High (1)	50 (9%)	5 (19%)
Low (2)	494 (91%)	22 (81%)

Table 4.5: Patrol area group memberships

across QEPA. Then, using the model's attack predictions, each area was assigned to an experiment group. Because we were not able to test all 544 areas, we selected a subset such that no two areas overlapped with each other and no more than two areas were selected for each patrol post (due to manpower constraints). In total, 5 areas in group 1 and 22 areas in group 2 were chosen. Note that this composition arose due to the preponderance of group 2 areas (see Table 4.5). We provide a breakdown of the areas' exact attack prediction rates in Figure 4.5b; areas with rates below 56% (5/9) were in group 2, and for example, there were 8 areas in group 2 with a rate of 22% (2/9). Finally, when we provided patrols to the rangers, *experiment group memberships were hidden to prevent effects where knowledge of predicted poaching activity would influence their patrolling patterns and detection rates*.

4.4.1 Field Test Results and Discussion

The field test data we received was in the same format as the historical data. However, because rangers needed to physically walk to these patrol areas, we received additional data that we have omitted from this analysis; observations made outside of a designated patrol area were not counted. Because we only predicted where snaring activity would occur, we have also omitted other observation types made during the experiment (e.g., illegal cattle grazing). We present results from this eight-month field test in Table 4.6. To provide additional context for these results, we also computed QEPA's park-wide historical CPUE (from November 2015 to June 2016): 0.03.

Experiment Group	Observation Count(%)	Mean Count(std)	Effort(%)	CPUE
High (1)	15 (79%)	3 (5.20)	129.54 (26%)	0.12
Low (2)	4 (21%)	0.18 (0.50)	368.52 (74%)	0.01

Table 4.6: Field test results: observations

Areas with a high attack prediction rate (group 1) had significantly more snare sightings than areas with low attack prediction rates (15 vs 4). This is despite there being far fewer group 1 areas than group 2 areas (5 vs 22); on average, group 1 areas had 3 snare observations whereas group 2 areas had 0.18 observations. It is worth noting the large standard deviation for the mean observation counts; the standard deviation of 5.2, for the mean of 3, signifies that not all areas had snare observations. Indeed, two out of five areas in group 1 had snare observations. However, this also applies to group 2's areas: only 3 out of 22 areas had snare observations.

We also present Catch per Unit Effort (CPUE) results in Table 4.6. When accounting for differences in areas' effort, group 1 areas had a CPUE that was over ten times that of group 2 areas. Moreover, when compared to QEPA's park-wide historical CPUE of 0.03, it is clear that our model successfully differentiated between areas of high and low snaring activity.

We present statistical significance results in Table 4.7 where we computed both a two-sample t-test and Cohen's d to assess our statistical significance and effect size. Each sample corresponded to one experiment group and was encoded to correspond to CPUE. A data point in a sample corresponded to the number of observations made in a single kilometer; sample 1 would have 130 data points and those data points would sum to 15. As can be seen by the results, the differences in group performance are extremely unlikely to be due to random chance, and those differences are also quite sizable. The results of this large-scale field test, the first of its kind for ML models in this domain, demonstrated that our model's superior predictive performance in the laboratory extends to the real world.

High Group Mean(std)	Low Group Mean(std)	t-statistic(df)	p-value	Cohen's d
0.12(0.44)	0.01(0.13)	4.09(497)	p<0.0001	0.42

Table 4.7: Field test results: statistical significance results

4.4.2 Do Rangers Already Differentiate between Areas of High and Low Snaring Activity?

Given that we've demonstrated our model can differentiate between where poaching is and isn't happening with high intensity, it is useful to examine whether rangers are also making this differentiation. If rangers are already able to make this differentiation, our model would not add much value to their operations. We present the following evidence and analyses that demonstrate that effort allocations can be improved such that high snaring activities are patrolled more; our predictive models can provide value to patrol planners' operations.

4.4.2.1 Pilot Field Test in Low Historical Effort Areas Found High Levels of Snaring Activity

In Section 3.6, we discussed the results from a one-month pilot field test where we asked rangers to patrol in areas that were not patrolled frequently in the past and were also predicted to be attacked by our model. While on these directed patrols, rangers found more signs of poaching in that one month than they had in neighboring regions in 91% of months in the history of the dataset. This result indicates that not only did our model correctly predict where poaching was occurring, but rangers may have been able to benefit from this guidance in the past had they known that the area was attractive to poachers.

High Group Mean(std)	Low Group Mean(std)	t-statistic(df)	p-value	Cohen's d
0.08(0.58)	0.02(0.19)	2.15(885)	p<0.02	0.24

Table 4.8: Historical patrolling analysis in field test areas: statistical significance results

Experiment Group	Effort
High (1)	87.46 (10%)
Low (2)	800.06 (90%)

Table 4.9: Historical effort allocation in field test areas

4.4.2.2 Park-Wide Historical Catch per Unit Effort is Low

In Section 4.4.1, we discussed the results for the large-scale field test. As a baseline, we presented a historical park-wide catch per unit effort (CPUE) of 0.03 to give additional context as to whether or not our high-group CPUE of 0.12 was of practical significance. In addition to indicating that our high-group CPUE was practically significant, the low historical CPUE also indicates that rangers may not have differentiated between areas of high and low poaching activity as well as our model. Had rangers been identifying high poaching activity areas as well as our model, we would expect the historical CPUE to have been higher.

4.4.2.3 Historical Effort Allocation in Field Test Areas

Additionally, we examine the historical allocation of effort to all of the areas across the park that were used in the field test. Because we are conducting a historical analysis, from November 2015 – June 2016, we recompute the experiment groups for each of those areas according to the ensemble's predictions for that historical time period. As such, 3 of 27 areas are classified as high (11%) and the remaining 24 of 27 areas are classified as low (89%). We performed a CPUE analysis on these historically reclassified field test areas (see Table 4.8) and present the proportions of allocated effort in Table 4.9.

In Table 4.8, although the ensemble successfully differentiates between areas of high and low poaching activity, ranger patrol effort is nearly uniform across the experiment groups (Table 4.9: rangers allocated 10% of their effort to the 11% of areas which were classified as high. To reiterate, while it is unreasonable (and perhaps unwise due to the risk of worsening any data biases) to expect rangers to allocate 100% of their patrolling effort to the high areas, we believe that had rangers known of these high poaching activity areas, we would see a more disproportionate allocation of patrol effort to those areas. Instead, we see this as an opportunity for our predictive model to provide value to rangers by differentiating between areas of high and low poaching activity.

4.4.3 Real-World Limitations

While the results of this field test validated our model's superior predictive performance in the field and demonstrated the model's selectivity, there are a couple of things to bear in mind. First, in ideal circumstances, experiment groups and patrol areas within each group would be patrolled with equal effort. Given rangers' time, capacity, and logistics constraints, however, this was not feasible to impose. Second, while these are strong and significant results for our model, this was a single field test conducted in a single park. While we are confident that this approach could be applied to other parks and other countries, further field testing will be required before we can be certain of our model's performance in those new settings.

Chapter 5

Analysis of Model Reactivity to Changes in Ranger Effort

As discussed in previous chapters, decision tree-based approaches outperformed other standard machine learning techniques such as SVMs, AdaBoost, and also a domain-specific graphical model (Nguyen et al., 2016). Given the success of the decision tree ensemble approach, our following analysis focuses on a bagging ensemble of decision trees and how the ensemble's predictions change as a function of ranger effort (i.e., number of kilometers patrolled). One goal of this analysis is to answer the question: "How effective will increasing patrolling be in increasing rangers' detections?" By answering this question, we can have more confidence that our model's superior predictive performance translates into meaningful input to patrol generation frameworks. Note that because the best-performing hybrid model in Chapter 4 was a joint work, I instead focus my analysis on how my model, the bagging ensemble component of the hybrid, reacts to changes in ranger effort.

Unlike in previous chapters, where only the previous time step's effort was considered as a feature, now we must consider the current time step's effort as a feature in our model. However, this change in the model's input altars the question that the model is answering from "Which areas are attractive to poachers?" (i.e., attackability) to "Which areas did we detect attacks?"

(i.e., detectability). In the context of understanding where we should send rangers to maximize their attack detections, the question of detectability is the appropriate question to answer.

5.1 Real-World Dataset

This analysis focuses on a real-world wildlife crime dataset from Uganda's Queen Elizabeth Protected Area (QEPA). Consisting of a wildlife conservation park and two wildlife reserves, QEPA spans approximately 2,520 square kilometers and is patrolled by wildlife park rangers. While on patrol, rangers collect data on animal sightings and signs of illegal human activity (e.g., poaching, trespassing). In addition to this observational data, the dataset contains terrain information (e.g., slope, vegetation), distance data (e.g., distance to nearest patrol post), animal density, and the number of kilometers walked by rangers in a given area (i.e., effort).

For this analysis, we divide QEPA into 1 square kilometer grid cells and compute several features based on the dataset's contents (e.g., observations, terrain, effort). Additionally, we group the observations and effort values (i.e., the values that change over time) into a series of month-long time steps. Finally, we compute two effort features, previous effort and current effort, that represent the amount of patrolling effort expended by rangers in the previous time step and current time step, respectively. Because effort is a continuous value (0 to ∞), we discretize the effort values into m effort groups (e.g., $m = 2$: `high` and `low`). For the following analysis, we focus on the case where $m = 2$.

5.2 Ensemble Model

Bagging (also referred to as Bootstrap aggregation technique) is an ensemble method (in this case applied to decision trees) where each tree is trained on a bootstrapped subset of the whole dataset. The subsets are generated by randomly choosing, with replacement, N observations where N is the dataset size. Once all trees in the ensemble are trained, the ensemble's predictions are generated by averaging the predictions from each tree. We trained a bagging ensemble using the *fitcensemble* function in MATLAB 2017a. For this model, the best training period consists of 5 years of data (based on repeated analyses for different train/test splits). The 11 input features consist of terrain information, geospatial features, and two patrol effort features (one for the previous time step's effort and one for the current time step's effort). The label for each data point corresponds to whether an attack was detected at that cell. For the training set, a label will be 1 if at any point in the training period an attack was detected (0 otherwise). For the test set, a label will be 1 if an attack was detected during the current time step.

Additionally, because our patrol generation approach can be used to generate, for example, monthly or annual patrols, we present results for bagging ensembles with four different time scales: one-month, three-month, six-month, and annual. For example, a monthly time scale ensemble would be trained on 60 months of data (i.e., effort values are monthly) and would be used to predict detected attacks in a single test month. Alternatively, an annual time scale ensemble would be trained on 5 years of data (i.e., effort values are annual) and would be used to predict detections for a single test year. Note that for each time scale, the test set contains a different amount of months (e.g., one month for one-month time scale), but all test sets' last month is the same; for example, if the one-month time scale's test set corresponded to November

2016, the three-month time scale's test set would correspond to September through November 2016. Additionally, the training dataset is defined based on a sliding time window such that the last month of the training dataset is the month prior to the first month of the test dataset; for the one-month time scale, if the test set corresponds to November 2016, then its training dataset would contain data for November 2012 to October 2016, whereas the three-month's training dataset would contain data for September 2012 to August 2016. For each time scale (one-month, three-month, six-month, and annual), the training sets contain data for 2,120, 2,129, 2,138, and 2,132 cells respectively. Note that a sample for a cell would not be present in the training data if it was not patrolled in that training period (hence the minor differences in training set compositions based on time scale).

We briefly present prediction performance results as verification that subsequent analyses are done on a realistic model. We also present baseline results from common boosting models – AdaBoost and RUSBoost (Seiffert et al., 2010). Additionally, we present a baseline, TrainBaseline, where if an attack was detected at a cell in the training data, the baseline will predict a detected attack for the test data (for cells that were not patrolled in the training data, and thus there is no training sample for that cell, a uniform random binary prediction is made). Due to the large class imbalance present in the dataset (many more negative labels than positives), we compute the area under a Precision-Recall curve (PR-AUC[1]) instead of the standard Receiver Operating Characteristic curve (which is not as informative for such a dataset) (Davis & Goadrich, 2006). We also present F1, Precision, and Recall scores.

Tables 5.1, 5.2, 5.3, and 5.4 correspond to model performances for one-month, three-month, six-month, and annual time scales respectively.

[1]Because TrainBaseline makes binary predictions and thus does not have continuous prediction values, PR-AUC is not computed for TrainBaseline.

Model	F1	Precision	Recall	PR-AUC
TrainBaseline	0.46	0.30	1	-
RUSBoost	0.19	0.11	0.76	0.19
AdaBoost	0.48	0.36	0.71	0.53
Bagging	**0.61**	**0.48**	**0.82**	**0.82**

Table 5.1: One-month time scale performance

Model	F1	Precision	Recall	PR-AUC
TrainBaseline	0.4	0.25	0.96	-
RUSBoost	0.21	0.12	0.96	0.28
AdaBoost	0.49	0.35	0.82	0.50
Bagging	**0.65**	**0.52**	**0.86**	**0.79**

Table 5.2: Three-month time scale performance

Model	F1	Precision	Recall	PR-AUC
TrainBaseline	0.41	0.26	0.90	-
RUSBoost	0.24	0.14	0.90	0.28
AdaBoost	0.57	0.46	0.77	0.55
Bagging	**0.69**	**0.61**	**0.96**	**0.80**

Table 5.3: Six-month time scale performance

Model	F1	Precision	Recall	PR-AUC
TrainBaseline	0.44	0.30	0.86	-
RUSBoost	0.31	0.19	0.85	0.32
AdaBoost	0.49	0.39	0.64	0.44
Bagging	**0.70**	**0.65**	**0.76**	**0.80**

Table 5.4: Annual time scale performance

As can be seen in all time scales, the Bagging model outperforms all other models in terms of F1, Precision, and PR-AUC. While Bagging does not always score the highest in recall, its precision score greatly outperforms the other models' precision. In practical terms, this means that the Bagging model will predict far less false positives (i.e., detections where there won't be any) and can thus better ensure that any patrol generation algorithm would not needlessly send rangers to areas where they won't detect attacks.

5.3 Effort Function Analysis

The goal of the patrol generation algorithm is to allocate effort such that rangers' crime detections per patrol are maximized. For the following analysis, we examine how the bagging ensemble's predictions change as a function of ranger effort. For example, if we allocate more effort to an area, does the model predict that rangers will detect a previously undetected attack? Alternatively, if we decrease effort in an area, does the model predict that rangers will be unable to detect a previously detected attack? We present results corresponding to the one-month, three-month, six-month, and annual (twelve-month) time scales. For example, for the three-month time scale, if we increase effort in an area over a period of three months, will rangers detect an attack in that area in any of the three months? In the appendix, we discuss results regarding how the attacker adapts to changes in patrol effort.

For this analysis, we present the changes in (1) the model's detected attack predictions and (2) the model's detected attack prediction probabilities when the effort in the current time step is changed. Both values are outputted by MATLAB's *predict* function for our learned ensemble. We refer to effort group 0 as `low` and group 1 as `high`; an increase in allocated effort, for example,

Effort Change	Neg to Pos	Pos to Neg	No Change (Pos)	No Change (Neg)
Low to High	115	16	285	1778
High to Low	0	54	86	188

Table 5.5: One-month time scale prediction changes as function of current effort

Effort Change	Neg to Pos	Pos to Neg	No Change (Pos)	No Change (Neg)
Low to High	119	30	172	1693
High to Low	2	110	122	274

Table 5.6: Three-month time scale prediction changes as function of current effort

would result in the effort group changing from `low` to `high`. Results for changes in predictions are shown in Tables 5.5, 5.6, 5.7, and 5.8, and changes in prediction probabilities are shown in Tables 5.9, 5.10, 5.11, and 5.12.

Because the trends are similar for each time scale, let's look at the three-month time scale as an example. In Table 5.6, for each type of change in effort (`low` to `high` or `high` to `low`), there are three possible outcomes for a prediction change: a negative prediction (no detection) can change to a positive prediction (detected attack), referred to as **Neg to Pos**, positive can change to negative (**Pos to Neg**), and there can be no change in the prediction (for either the positive or negative prediction cases). Given these outcomes, we make the following observations. First, there are a substantial number of cells whose corresponding detection predictions do not change as a result of changes in effort. In the case of the unchanged positive predictions, these are predicted to be high-risk cells where rangers will find poaching activity even if they allocate relatively low effort values to it. For unchanged negative predictions, these correspond to low-risk cells that are

Effort Change	Neg to Pos	Pos to Neg	No Change (Pos)	No Change (Neg)
Low to High	109	38	118	1440
High to Low	4	198	96	519

Table 5.7: Six-month time scale prediction changes as function of current effort

Effort Change	Neg to Pos	Pos to Neg	No Change (Pos)	No Change (Neg)
Low to High	230	18	94	1152
High to Low	16	269	69	674

Table 5.8: Annual time scale prediction changes as function of current effort

Effort Change	Inc	Mean Inc	Dec	Mean Dec	No Change(Pos)	No Change(Neg)
Low to High	1663	0.17	473	0.07	7	51
High to Low	100	0.06	216	0.22	0	12

Table 5.9: One-month time scale prediction probability changes as function of current effort

essentially predicted to not be attacked at all. Second, while there are substantially more instances of predicted detections increasing as a result of increasing effort, there are still some instances of predicted detections decreasing as a result of increasing effort. However, because there is not a rational explanation for this trend, these rare instances are likely due to noise in the model. Finally, we make the same observation regarding the case where detections mostly decrease as a result of decreasing effort while detections increase at only two cells.

Because the trends are similar for each time scale, let's look at the three-month time scale as an example. As for the prediction probability changes in Table 5.10, we examine changes in the prediction probability with increases and decreases referred to as **Inc** and **Dec** respectively, the mean changes in prediction probability for the increase and decrease cases (referred to as **Mean Inc** and **Mean Dec** respectively), and also in the instances where there was no change in the probability for both the positive (i.e., probability ≥ 0.50) and negative (i.e., probability < 0.50) cases. First, when effort is increased, many more cells are predicted to have a substantial

Effort Change	Inc	Mean Inc	Dec	Mean Dec	No Change(Pos)	No Change(Neg)
Low to High	1546	0.16	423	0.09	4	41
High to Low	142	0.09	358	0.22	0	8

Table 5.10: Three-month time scale prediction probability changes as function of current effort

Effort Change	Inc	Mean Inc	Dec	Mean Dec	No Change(Pos)	No Change(Neg)
Low to High	1391	0.21	293	0.10	3	18
High to Low	275	0.07	526	0.25	0	16

Table 5.11: Six-month time scale prediction probability changes as function of current effort

Effort Change	Inc	Mean Inc	Dec	Mean Dec	No Change(Pos)	No Change(Neg)
Low to High	1283	0.25	192	0.10	1	18
High to Low	357	0.09	637	0.26	2	32

Table 5.12: Annual time scale prediction probability changes as function of current effort

increased prediction probability (mean change of 16%). While there are a non-trivial number of cells with a decrease in their prediction probability, the mean decrease is approximately half that of the mean increase, with the difference being statistically significant ($t(1967) = 15.09, p << 0.001$), and is thus interpreted as noise. Second, when effort is decreased, there are many more cells with a decrease in prediction probability than increase. Additionally, the mean decrease in prediction probability is more than twice that of the mean increase (22% vs 9%) and is also statistically significant ($t(498) = 12.90, p << 0.001$). Finally, as with the prediction changes in Table 5.6, a few cells are low-risk and increasing effort will not result in a corresponding increase in predicted detection probability. While changes in predicted probability do not necessarily correspond to changes in actual predictions (0/1), the shifts in probability do provide a concrete indication of the actual impacts that coverage has on the model's predictions.

Chapter 6

Beware the Soothsayer: From Attack Prediction Accuracy to Predictive Reliability in Security Games

By mathematically optimizing and randomizing the allocation of defender resources, Security Games provide a useful tool that has been successfully applied to protect various infrastructures such as ports, airports, and metro lines (Tambe, 2011). Network Security Games (NSGs), a type of Security Game, can be applied to interdict the flow of goods in smuggling networks (e.g., illegal drugs, ivory) or defend road networks from terrorist attacks (e.g., truck bombs). In comparison to previous work in Security Games (Shieh et al., 2012), however, the number of possible actions for both attacker and defender grow exponentially for NSGs; novel scaling techniques have been developed to address this challenge by Jain et al. (Jain et al., 2011) for perfectly rational attackers.

While early work in Security Games relied on the assumption of perfect adversary rationality, more recent work has shifted away towards modeling adversary bounded rationality (Nguyen et al., 2013; Cui & John, 2014; Kar et al., 2015; Abbasi et al., 2015). In the effort to model human decision making, many human behavior models are being developed. As more Security Game applications are being deployed and used by security agencies (Shieh et al., 2012; Fave

et al., 2014), it becomes increasingly important to validate these models against real-world data to better ensure that these and future applications don't cause substantial losses (e.g., loss of property, life) for the defender. In efforts to generate real-world data, previous work (Shieh et al., 2012; Fave et al., 2014) has demonstrated that field experiments are time-consuming and complex to organize for all parties involved; the amount of field experiments that can be feasibly conducted is grossly limited. Thus, in real-world situations, we will have limited field data.

By analyzing the prediction accuracy of many models on an existing large dataset of human subject experiments, previous works (Cui & John, 2014; Abbasi et al., 2015) empirically analyze which models most closely resemble human decision making for Stackelberg (SSG) and Opportunistic Security Games. While these works demonstrate the superiority of some models in terms of prediction accuracy and fitting performance, they do not address the larger, implicit question of how the models' corresponding strategies would perform when played against human subjects (i.e., average defender expected utility). We do not know how well the prediction accuracy of a model will correlate with its actual performance if we were to generate a defender strategy that was based on such a model; informally defined, **predictive reliability** refers to the percentage of strong correlations between a model's prediction accuracy and the model's actual performance. It is also unknown whether the prediction accuracy analysis approach will be suitable, especially for NSGs, in situations where we have limited field data from which to learn the models. As previously discussed, the amount of field experiments that can be conducted (and thus the amount of training data available for learning) is limited; it is important to know whether the model with superior prediction accuracy will actually result in higher defender gains than a model with worse prediction accuracy (especially when training data is limited). This raises the following question

for NSG research: "Without the ability to collect very large amounts of data for training different bounded rationality models and without the ability to conduct very large amounts of tests to compare the performance of these models in action, how do we ensure high predictive reliability and choose the most promising models?"

We first lay the groundwork for determining whether our proposed construct of predictive reliability is valid in SSGs. As such, we first (i) conduct an empirical evaluation of predictive reliability in SSGs in situations where there is a large amount of training data. We then (ii) evaluate predictive reliability for NSGs. In this study, we use NSG human subject data from the lab and train our models on enough data such that prediction accuracies converge [1]. Following this primary analysis, we then examine the various factors that may influence predictive reliability. We propose a metric called Exposed Attack Surface (EAS) which is related to the degree of choice available to the attacker for a given training set. We then (iii) examine the effects of EAS on predictive reliability, and (iv) investigate which graph features influence predictive reliability.

Our primary analysis shows that (i) predictive reliability is strong for an SSG dataset where there is sufficient training data, (ii) even though there is sufficient training data (at least to see our models' prediction accuracies converge), predictive reliability is poor for NSGs. In our analysis to discover which factors have the most influence on predictive reliability, we find that (iii) a training set with a higher EAS score results in better predictive reliability than a training set with a lower EAS score. Note that this finding is independent of the training set's size (both training sets are of the same size). While it won't always be possible to obtain training data with a large exposed attack surface, if we do have it, we can be more confident in the predictive reliability of

[1] In other words, to simulate real-world scenarios, we do not assume the presence of very large amounts of data, but nonetheless, there is a sufficient amount of NSG data included in our study to at least see a stable prediction made by our different behavior models.

$g(V, E)$	General directed graph.
J	Set of paths in graph g.
k	Number of defender resources.
X	Set of defender allocations, $X = \{X_1, X_2, ..., X_n\}$.
X_i	i^{th} defender allocation $X_i = \{X_{ie}\}$ $\forall e$, $X_{ie} \in \{0, 1\}$.
A	Set of attacker paths, $A = \{A_1, A_2, ..., A_m\}$.
A_j	j^{th} attacker path $A_j = \{A_{je}\}$ $\forall e$, $A_{je} \in \{0, 1\}$.
t_j	Target t in the graph g such that the attacker takes path j to attack t.
$\mathcal{T}(t_j)$	The reward obtained for a successful attack on target t by taking path j s.t. $A_j \cap X_i = \emptyset$ where A_j is the attacker's selected path to attack target t and X_i is the selected defender allocation.
x	Defender's mixed strategy over X.
x_i	Probability of choosing defender pure strategy X_i.
$EU_d(x)$	Defender's expected utility from playing x.
z_{ij}	Function that refers to whether a defender allocation X_i intersects with an attacker path A_j. If there is an intersection, returns 1. Else, 0.

Table 6.1: Notations used in this paper

our models. In addition, we find that (iv) there is a strong correlation between poor predictive reliability and whether a graph has both a low to moderate number of intermediate nodes and a low to moderate number of outgoing edges from source nodes.

6.1 Background: Network Security Games

This paper will address zero-sum Network Security Games (NSGs). For a table of notations used in this paper, see table 6.1. In NSGs, there is a network (shown in Figure 6.1) which is a graph g containing a set of nodes/vertices V (the dots/circles in the figure) and a set of edges E

(the arrows in the figure, labelled 1-6). In the network, there is a set of target nodes, denoted by $T \subset V$. While the defender attempts to allocate her limited resources to protect these target nodes, the attacker can observe the defender's patrolling strategy and then attack one of the target nodes based on that observation.

Attacker strategies. The attacker can start at a source node $s \in S$ (where $S \subset V$ is the set of all source nodes in the network) and chooses a sequence of nodes and edges leading to a single target node $t \in T$. The attacker's decision corresponds to a single path $j \in J$ and is referred to as the attacker's path choice $A_j \in A$ where A is the set of all possible paths that the attacker can choose.

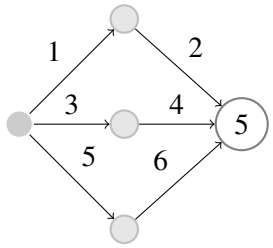


Figure 6.1: Example graph

Defender strategies. The defender can allocate her k resources to any subset of edges in the graph; each allocation is referred to as a pure strategy for the defender, denoted by X_i. There are $\binom{|E|}{k}$ defender pure strategies in total, and we denote this set of pure strategies by X. Then, a defender's *mixed* strategy is defined as a probability distribution over all pure strategies of the defender, denoted by $x = \{x_i\}_{i=1}^{N}$, where x_i is the probability that the defender will follow the pure strategy X_i and $\sum_i x_i = 1$.

Defender and attacker utilities. An attack is successful if the attacker's path choice does not contain any edges in common with the defender's allocation ($X_i \cap A_j = \emptyset$), and the attacker will receive a reward $\mathcal{T}(t_j)$ while the defender receives a penalty of -$\mathcal{T}(t_j)$. Here, t_j is the target node on the path A_j. Conversely, if the attack is unsuccessful (i.e., the attacker's path intersected with the defender's allocation), both attacker and defender receive a payoff of 0.

Finally, the defender's expected utility of executing a mixed strategy x given an attacker path A_j can be computed as shown in Equation 6.1 where the term $p_j(x)$ (defined in Equation 6.2) refers to the probability that the adversary will be caught when choosing path A_j to attack target node t_j. In zero-sum games, the attacker's expected utility for choosing path A_j is equal to the opposite of the defender's expected utility, i.e., $EU_a(x, A_j) = -EU_d(x, A_j)$.

$$EU_d(x, A_j) = -\mathcal{T}(t_j) \cdot (1 - p_j(x)) \tag{6.1}$$

In Equation 6.2, z_{ij} is an integer which indicates if the defender's pure strategy X_i intersects with the attacker path A_j ($z_{ij} = 1$) or not ($z_{ij} = 0$).

$$p_j(x) = \sum_{X_i \in X} z_{ij} x_i \tag{6.2}$$

6.2 Adversary Behavioral Models

We now present an overview of all the adversary behavioral models which are studied in this paper.

6.2.1 The Perfectly Rational Model

In NSG literature, the adversary is often assumed to be perfectly rational and will always maximize his expected utility. In other words, the adversary will choose the optimal attack path that gives him the highest expected utility, i.e., $A_{opt} = \text{argmax}_{A_j} EU_a(x, A_j)$.

6.2.2 The Quantal Response Model

The Quantal Response (QR) model for NSGs was first introduced by Yang et al. (Yang et al., 2012). However, their formulation only works under the assumption that there is one defender resource available, and as a result, we present a revised version of the QR model for a zero-sum NSG with multiple defender resources. In short, QR predicts the probability that the adversary will choose a path A_j, which is presented as the following:

$$q_j(\lambda|x) = \frac{e^{\lambda EU_j^a(x)}}{\sum_{A_k \in A} e^{\lambda EU_k^a(x)}} \tag{6.3}$$

where λ is the parameter that governs the adversary's rationality. For example, $\lambda = 0.0$ indicates that the adversary chooses each path uniformly randomly. On the other hand, $\lambda = \infty$ means that the adversary is perfectly rational. Intuitively, there is a higher probability that the adversary will follow a path with higher expected utility.

6.2.3 The Subjective Utility Quantal Response Model

Unlike QR, the Subjective Utility Quantal Response (SUQR) model (Nguyen et al., 2013) models the attacker's expected utility calculation as a weighted sum of decision factors such as reward and path coverage. As demonstrated by Nguyen et al. (Nguyen et al., 2013) for SSGs and Abbasi et al. (Abbasi et al., 2015) for Opportunistic Security Games (OSGs), SUQR performs better than QR for attack prediction accuracy. As such, we present an NSG adaptation of SUQR as shown in

Equation 6.4. Specifically, SUQR predicts the probability that the adversary chooses a path A_j as the following:

$$q_j(\omega|x) = \frac{e^{\omega_1 p_j(x)+\omega_2 \mathcal{T}(t_j)}}{\sum_{A_k \in A} e^{\omega_1 p_k(x)+\omega_2 \mathcal{T}(t_k)}} \tag{6.4}$$

where (ω_1, ω_2) are parameters corresponding to an attacker's preferences (i.e., weights) on the game features: the probability of capture $p_j(x)$ and the reward for a successful attack $\mathcal{T}(t_j)$.

6.2.4 The SUQR Graph-Aware Model

The previous models, designed for traditional Stackelberg Games, do not account for the unique features of Network Security Games. As such, we present some NSG-specific features that can be incorporated into the existing SUQR model in the form of additional parameters. Each of these features is computed for each path $A_j \in A$.

Path length simply refers to the number of edges in a path A_j, and the corresponding weight is referred to as ω_3 in Equation 6.5. This model will henceforth be referred to as GSUQR1 (i.e., Graph-SUQR w/ 1 parameter). Yang et al. (Yang et al., 2012) also made use of path length as one of the tested QR heuristics.

$$q_j(\omega|x) = \frac{e^{\omega_1 p_j(x)+\omega_2 \mathcal{T}(t_j)+\omega_3|A_j|}}{\sum_{A_k \in A} e^{\omega_1 p_k(x)+\omega_2 \mathcal{T}(t_k)+\omega_3|A_k|}} \tag{6.5}$$

We also compute the maximum total degree (weight ω_4) of a path. This is an aggregate measure (maximum) of the path's nodes' indegrees (i.e., number of edges coming into the node) + outdegrees (i.e., number of edges leaving the node). We refer to this measure as MTO. A low

value for this corresponds to simple paths with little connections to other areas of the graph; a high value corresponds to a path with one or more nodes that are highly connected to other paths. The resultant q_j function is shown in Equation 6.6, and this model is henceforth referred to as GSUQR2.

$$q_j(\omega|x) = \frac{e^{\omega_1 p_j(x)+\omega_2 \mathcal{T}(t_j)+\omega_3|A_j|+\omega_4 MTO_j}}{\sum_{A_k \in A} e^{\omega_1 p_k(x)+\omega_2 \mathcal{T}(t_k)+\omega_3|A_k|+\omega_4 MTO_k}} \tag{6.6}$$

6.3 Defender Strategy Generation

In this section, we present the approach used to generate defender strategies for the boundedly rational adversary models. [2] Because the strategy space for NSGs can grow exponentially large, we address this by adapting a piecewise linear approximation approach, PASAQ, first introduced by Yang et al. (Yang, Ordonez, & Tambe, 2012). Note that while we only show the PASAQ formulation as generating defender strategies for the QR model, we also adapted it for the SUQR, GSUQR1, and GSUQR2 models as well. Whereas the original PASAQ algorithm worked for SSGs involving independent targets and coverages, this paper has adopted PASAQ for NSGs, where non-independent path coverage probabilities ($p_j(x)$) must be taken into account. PASAQ works by performing a binary search to solve a non-linear fractional objective function. Determining whether the current solution is feasible, however, is a non-convex problem, and this feasibility checking problem is expressed as an inequality in Equation 6.7, where r is the current

[2] The algorithm to generate a Maximin strategy can be found in (Jain et al., 2011).

binary search solution, x^* is the optimal defender mixed strategy, and $EU_d(x)$, the defender's expected utility given an adversary following the QR model, is defined in Equation 6.8. [3]

$$r \leq EU_d(x^*) \tag{6.7}$$

$$EU_d(x) = \frac{\sum_{A_j \in A} e^{\lambda EU_a(x,A_j)} EU_d(x, A_j)}{\sum_{A_j \in A} e^{\lambda EU_a(x,A_j)}} \tag{6.8}$$

After rewriting Equation 6.7 as a minimization function and further expansion, we obtain two non-linear functions

$f_(j)^{(1)}(p_j(x))$=$e^{\lambda(1-p_j(x))\mathcal{T}(t_j)}$ and

$f_(j)^{(2)}(p_j(x))$=$(1-p_j(x))e^{\lambda(1-p_j(x))\mathcal{T}(t_j)}$ which are to be approximated. To do so, we divide the range $p_j(x) \in [0, 1]$ into S segments (with endpoints [$\frac{s-1}{S}$,$\frac{s}{S}$, $s = 1 \ldots S$]) and will henceforth refer to each segment that contains a portion of $p_j(x)$ as $\{p_{js}, s = 1 \ldots S\}$. For example, p_{j2} refers to the second segment of $p_j(x)$ which is located in the interval [$\frac{1}{S}$ and $\frac{2}{S}$]. Our piecewise approximation follows the same set of conditions from (Yang et al., 2012): each $p_{js} \in [0, \frac{1}{S}] \forall s = 1 \ldots S$ and $p_j = \sum_{s=1}^{S} p_{js}$. In addition, any $p_{js} > 0$ only if $p_{js'} = \frac{1}{S}, \forall s' < s$; in other words, p_{js} can be non-zero only when all previous partitions are completely filled (i.e., = $\frac{1}{S}$). Enforcing these conditions ensures that each p_{js} is a valid partition of $p_j(x)$. Following the definition from (Yang et al., 2012), the piecewise linear functions are represented using $\{p_{js}\}$. The S+1 segment end points of $f_j^{(1)}(p_j(x))$ can be represented as $\{(\frac{s}{S}, f_j^{(1)}(\frac{s}{S}))$, s=0. . . S$\}$ and the slopes of each

[3] Details on the binary search algorithm can be found in Yang et al.'s original PASAQ formulation (Yang et al., 2012).

segment as $\{\gamma_{js}$,s=1...S$\}$. Starting from $f_j^{(1)}(0)$, we denote the piecewise linear approximation of $f_j^{(1)}(p_j(x))$ as $L_j^{(1)}(p_j(x))$:

$$\begin{aligned} L_j^1(p_j(x)) &= f_j^{(1)}(0) + \sum_{s=1}^{S} \gamma_{js} p_{js} \\ &= e^{\lambda \mathcal{T}(t_j)} + \sum_{s=1}^{S} \gamma_{js} p_{js} \end{aligned} \tag{6.9}$$

The approximation of function $f_j^{(2)}(p_j(x))$ is performed similarly (slopes denoted as $\{\mu_{js}$,s=1...S$\}$) and yields $L_j^{(2)}(p_j(x))$.

$$L_j^2(p_j(x)) = e^{\lambda \mathcal{T}(t_j)} + \sum_{s=1}^{S} \mu_{js} p_{js} \tag{6.10}$$

Given the definition of these two piecewise linear approximations, the following system of equations details the solution feasibility checking function (invoked during the binary search):

$$\min_{x,b} \sum_{A_j \in A} (e^{\lambda \mathcal{T}(t_j)} + \sum_{s=1}^{S} \gamma_{js} p_{js}) r \tag{6.11}$$

$$+ \sum_{A_j \in A} \mathcal{T}(t_j)(e^{\lambda \mathcal{T}(t_j)} + \sum_{s=1}^{S} \mu_{js} p_{js}) \tag{6.12}$$

$$s.t \sum_{X_i \in X} x_i \leq 1 \tag{6.13}$$

$$p_j(x) = \sum_{s=1}^{S} p_{js} \tag{6.14}$$

$$p_j(x) = \sum_{X_i \in X} z_{ij} x_i \tag{6.15}$$

$$b_{js} \frac{1}{S} \leq p_{js}, \forall j, s = 1 \ldots S-1 \tag{6.16}$$

$$p_{j(s+1)} \leq b_{js}, \forall j, s = 1 \ldots S-1 \tag{6.17}$$

$$0 \leq p_{js} \leq \frac{1}{S}, \forall j, s = 1 \ldots S \tag{6.18}$$

$$b_{js} \in \{0, 1\}, \forall j, s = 1 \ldots S-1 \tag{6.19}$$

$$z_{ij} \in \{0, 1\}, \forall i, j \tag{6.20}$$

where b_{js} is an auxiliary integer variable that is equal to 0 only if $p_{js} < \frac{1}{S}$ (Equation 6.16). Equation 6.17 enforces that $p_{j(s+1)}$ is positive only if b_{js} = 1. In other words, b_{js} indicates whether or not $p_{js} = \frac{1}{S}$ and thus enforces our previously described conditions on the piecewise linear approximation (ensuring each p_{js} is a valid partition). As demonstrated in (Yang et al., 2012), given a small enough binary search threshold ϵ and sufficiently large number of segments S, PASAQ is arbitrarily close to the optimal solution.

6.4 Human Subject Experiments

6.4.1 Experimental Overview

In order to test the effectiveness of these algorithms against human adversaries, we ran a series of experiments on Amazon Mechanical Turk (AMT). Even though we run these (effectively speaking) laboratory experiments, our goal is to collect this data in such a way as to simulate field conditions where there is limited data. Due to the nature of human subject experiments, special considerations were made to reduce the effects of bias and noise. Charness et al. (Charness, Gneezy, & Kuhn, 2012) discussed important design choices for between-subject and within-subject experiment designs in order to minimize the harmful effects of bias; we made use of these recommendations in our experimental design, as discussed below.

6.4.1.1 Validation Rounds

We included two validation graphs in each experiment set (for a total of seventeen graphs presented in random order). Validation graphs are special case graphs where all but one path has a coverage probability of 1.0 (i.e., "wrong paths"), and one remaining path (i.e., the only correct solution) has a coverage probability of 0.0. We dropped participants that selected a covered path (i.e., the wrong path) in any of the two validation graphs; we concluded that players who failed this validation test were either playing randomly or didn't understand the instructions and would only confound our analysis.

6.4.1.2 Within-Participant Biases

For each defender algorithm, we computed an optimal defender mixed strategy on every graph. If we presented every combination of defender strategy and graph to each participant, however, we would encounter substantial within-subject bias. For example, if a participant first played on graph "A" with strategy "a" and then played on graph "A" with strategy "b", their first instinct may be to see if their previous solution will work again; upon seeing the same graph again, their decision making would be immediately biased towards the path they chose previously. To address this bias, we split up the experiment into multiple subject pools and randomly assigned participants to each subject pool. Although we conducted experiments for eight strategies on fifteen graphs (for a total of 120 combinations of strategy $\times$ graph), each subject pool was assigned to play against only one strategy across the 15 graphs. Thus, participants in each subject pool played each graph exactly once.

6.4.1.3 Learning Effects

Learning effects were also of concern to our experiments. After playing on one or two graphs, participants would become more familiar with the game itself and therefore may have some reinforced notions or heuristics for finding a path through the graph. Although this cannot be completely avoided, we attempted to minimize this by randomizing the order in which graphs were presented to participants and by withholding the result of each round until the end of the game; participants were not able to use success information from each round to influence their decision-making in future rounds. We also only allowed participants to participate in these experiments once; even if we run another experiment with a different set of graphs, repeat participants will exhibit different behaviors that will confound comparisons with first-time participants.

6.4.1.4 Compensation

Participant motivation is an important aspect of human subject experiments. To ensure that participants were thinking about their decisions and not playing randomly, we rewarded participants with additional money if they performed well in the experiments. Because we could not inform participants of their successes during the experiment due to aforementioned learning effects, we informed participants of the following bonus structure prior to the experiment. For each graph where a participant successfully attacked a target (i.e., without getting caught by the defender on a covered edge), they received bonus points equal to that target's reward value. At the end of the experiment, they received a bonus payment equal to the sum of their bonus points divided by 100 (e.g., an additional 80 cents if they received 80 points throughout the experiment). Note that if they got caught on a graph, they received zero points for that round. In addition to any bonus payment, all participants received a base payment of $1.50.

6.4.2 Experiment Data Composition

6.4.2.1 Participants and Dataset Sizes

In our experiments, all eligible AMT participants satisfied a set of requirements. They must have participated in more than 1000 prior AMT experiments with an approval rate of $\geq 95\%$, and we required that all participants were first-time players in this set of experiments. Out of 551 participants, 157 failed to complete all graphs or did not pass both validation rounds. The remainder, 394, successfully completed all rounds and passed both validation rounds, and we used only their data in the following data analyses.

6.4.2.2 Graph Design and Generation

To ensure our findings were not limited to a single set of homogeneous graphs, we generated three sets of random geometric graphs. Eppstein et al. demonstrated that geometric graphs were a suitable analogue to real-world road networks due to road networks' non-planar connectivity properties (Eppstein & Goodrich, 2008). Each set was assigned a predefined neighborhood radius (r), corresponding to the maximum distance between two nodes for an edge to exist, and a predefined number of intermediate nodes (v_i). Set 1, a set of sparse random geometric graphs, had r = 0.2, v_i = 10, and was required to have at least 15 edges. Set 2, a set of densely connected graphs, had r = 0.6 and v_i = 4. Set 3, a set of intermediately connected graphs, had r = 0.4 and v_i = 7. In addition, all sets were generated with a set of common constraints; each graph was constrained to have no more than 30 edges, exactly two source nodes, and exactly three destination nodes (with reward values 3, 5, and 8).

For each set, we generated 100 unique random geometric graphs. For each graph, we first randomly placed the nodes in a 2-D region (a unit square), and edges were drawn between nodes that were, at most, a 2-norm distance r away from each other. During post-processing, invalid connections, such as edges connecting source nodes to other source nodes, were removed. After the set was generated, we computed a Maximin, QR, and SUQR strategy for each graph and computed a distance score. This distance score measured the 1-norm distance between the probability distributions (i.e., the mixed strategies) for two sets of strategies: QR and SUQR, and Maximin and SUQR; graphs with distinctly different defender strategies (in terms of the coverage probabilities on paths) would receive a high distance score. The five graphs with the highest distance scores were kept for the final set.

6.4.2.3 Model Parameter Learning

The full experiment set consists of eight subject pools. For the purposes of learning the model parameters for the human behavior models, however, we divided the experiment set into three separate experiment sets. The first experiment set consists solely of the Maximin subject pool (no model learning required). The latter two experiment sets are defined by the training dataset used to train the models (e.g., the experiment data from the Maximin subject pool). As was done in previous work on applying human behavior models to Security Games (Nguyen et al., 2013; Kar et al., 2015; Abbasi et al., 2015; Yang et al., 2012), we use Maximum Likelihood Estimation (MLE) to learn the parameter values (i.e., weights) for each behavior model. Because training data may be limited in the real-world, we limit the scope of each training dataset to contain data from only one subject pool. Unlike previous work in NSGs by Yang et al. (Yang et al., 2012), where one set of weights was learned across all graphs (i.e., an aggregate weight), we found that the log-likelihood was highest when weights were learned individually for each graph.

6.4.2.4 Experiment Set Composition

As mentioned previously, the experiments are divided into three separate experiment sets. Each combination of coverage strategy $\times$ graph set was assigned to their own subject pool. Prior to running these experiments, however, we had no training data on which to learn weights for the behavior models. Thus, the first experiment set, experiment set 1, only contains a coverage strategy generated by the Maximin algorithm.

Experiment set 2 contains coverage strategies generated by the corresponding PASAQ algorithms for the QR (Equation 6.3), SUQR (Equation 6.4), GSUQR1 (Equation 6.5), and GSUQR2

(Equation 6.6) models. For the models used to generate these strategies, we used the Maximin dataset as the training dataset to learn each model's weights. To help differentiate from the datasets in experiment set 3, we will refer to the datasets collected in experiment set 2 as QR-M, SUQR-M, GSUQR1-M, and GSUQR2-M.

Experiment set 3 also contains coverage strategies generated for the QR (Equation 6.3), SUQR (Equation 6.4), and GSUQR1 (Equation 6.5) models. Instead of learning on Maximin data, however, we instead learn on GSUQR1-M data (from experiment set 2). As we will demonstrate later, learning from a non-Maximin dataset has a substantial positive impact on predictive reliability. As was done for experiment set 2, we will refer to the datasets collected in experiment set 3 as QR-S, SUQR-S, and GSUQR1-S.

6.4.3 Data Analysis Metrics

The following section discusses the various metrics used throughout our data analysis. First, we will introduce three metrics for computing model prediction accuracy (the degree to which a model correctly predicted attacker behavior). Next, we will introduce our proposed predictive reliability metric, which measures the degree to which models' predictions correspond to their actual performances. Finally, we introduce our last proposed metric, Exposed Attack Surface, which measures the number of unique path choices available to the attacker.

6.4.3.1 Model Prediction Accuracy

In previous empirical analyses (Cui & John, 2014; Abbasi et al., 2015) and in our own analysis, prediction accuracy measures are key to understanding the relative performance of behavior models; accuracy measures seek to answer the question "How well does this model predict human

behavior?" Computed over all paths for each model $\times$ graph $\times$ coverage strategy combination, prediction accuracy quantifies the degree to which a model's predictions of attacker behavior were correct.

Regardless of a graph's size or coverage strategy, however, only a few paths have an actual probability of attack $(q_j) > 6\%$; most paths in most graphs are attacked with very low frequency. When looking at all paths in a graph, the average absolute prediction error (AAE) is 3%, regardless of the behavior model making the prediction. It appears that the error "outliers" are actually the primary values of interest. In other words, because there is no discriminatory power with the average, we instead analyze the maximum absolute prediction error (MAE) (Equation 6.21) for each model, where $g \in G$ is a graph in the experiment set, ϕ is the behavior model (along with its weights) being evaluated, q_j is the behavior model ϕ's predicted attack proportion on path A_j given defender mixed strategy x, and $\hat{q_j}$ is the actual attack proportion on path A_j.

$$MAE(g, x, \phi) = \max_{A_j \in A} |q_j - \hat{q_j}| \tag{6.21}$$

As mentioned previously, only a few paths in a graph have some substantial probability of being attacked. Over all eight datasets, on average (across all graphs), 70% of all attacks occurred on only three paths (per graph). Thus, it is prudent to also analyze a model's prediction accuracy on these so-called "favored" paths.

Definition 3. *A path A_j is defined as a **favored path** A_{fj} if its actual probability of attack (q_j) is $\geq$ 10%.*

Similar to MAE but instead only over the favored paths $A_{fj} \subset A_j$ in a graph, we compute the maximum absolute error over favored paths (referred to as FMAE). Since this subset of paths does not suffer from excessive skewing, it is appropriate to also analyze the average absolute error (FAAE) over the set of favored paths A_{fj}.

6.4.3.2 Predictive Reliability

Now that we've introduced our prediction accuracy metrics, we turn our attention to the primary focus of our paper: predictive reliability — the degree to which models' prediction accuracies correspond with their corresponding strategies' performances in experiments. If predictive reliability is poor, then models chosen on the basis of having the best prediction accuracy may not perform the best when tested against actual humans; when field-deployment resources are limited, those resources should not be wasted on models that end up performing very poorly in the field!

After all human subject experiments have been conducted (we refer to the whole set of attack data as A_d), we can compute predictive reliability. Put simply, predictive reliability is the percentage of strong Pearson correlations. These correlations are computed separately for each combination of graph ($g \in G$), prediction accuracy metric (PAM), and testing dataset ($Te \in A_d$). For a given g, PAM, and Te, we compute the Pearson correlation over all models' (1) prediction accuracy on Te (using PAM), and (2) actual defender utility on the model's corresponding attack data (e.g., for model QR trained on Maximin, compute on the QR-M dataset). Note that if a model was trained on Te or if the model's corresponding attack data is Te, it is omitted from the Pearson correlation for that combination of g, PAM, and Te.

Definition 4. ***Predictive reliability*** *is defined as the percentage of correlations between actual utility values and prediction accuracies that are both (1) strong (magnitude $>$ 0.70), and (2) in the desired direction (negative: as error decreases, actual utility increases). In other words, predictive reliability corresponds to the percentage of strong correlations (correlation $<$ -0.70).*

6.4.3.3 Exposed Attack Surface

We now introduce our second proposed metric, Exposed Attack Surface (EAS). While early discussion of attack surface exposure was done by Manadhata et al. (Manadhata & Wing, 2004), more recently, Kar et al. (Kar et al., 2015) applied this concept to Repeated Stackelberg Security Games to improve the defender's utility against human subjects. EAS measures the number of unique attacker choices (i.e., paths) for a graph $\times$ strategy combination. To phrase this metric as a question, "Given a coverage strategy and graph, how many paths in the graph have a unique combination of path coverage and reward?" Referring to Figure 6.2 as an example, there are three separate paths to target 5. While two of these paths have the same path coverage of $\{0.2, 0.2\}$ (one attack surface), the other path has 0 path coverage (the second attack surface). Finally, the path to target 8 constitutes the last attack surface; the example figure's EAS score is 3. Although there are four paths in Figure 6.2, two of these paths are equivalent to each other (i.e., same reward and coverage) and thus there are only three unique path choices (i.e., the EAS score) for the attacker.

Definition 5. ***Exposed Attack Surface*** *is defined as the number of unique combinations of reward $\mathcal{T}(t_j)$ and path coverage probability $p_j(x)$ over all paths A in a graph g.*

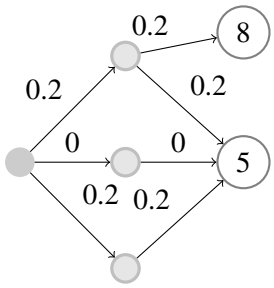


Figure 6.2: Example graph 2

When computing this metric for a dataset $d_{\phi,G} \in D_{\Phi,G}$, we take the sum of EAS scores for each graph $\times$ coverage strategy (corresponding to a model ϕ) combination. To illustrate the simple (but important) intuition behind EAS, we present two extreme cases: (1) consider a training dataset that consists of a single graph $\times$ coverage strategy such that the graph's EAS score is one; all paths to the single target have identical coverage (i.e., one unique path choice). When attempting to learn model parameters, it would be impossible to differentiate between attacker choices; obviously, this training set with a low EAS score is ill-suited for use in model learning. (2) In contrast, a training dataset with a high EAS score implies that there are many distinguishable attacker choices. Attacker choices over these many unique paths provide information about their preferences such that we can more effectively train a model; we hypothesize that a training dataset that contains more information about attacker preferences (i.e., one with high EAS) is superior to one that provides less information (i.e., low EAS).

6.5 Predictive Reliability Analysis

After defining predictive reliability in the previous section (Section 6.4.3.2), we now evaluate predictive reliability in previous work by Nguyen et al. (Nguyen et al., 2013) for SSGs, and then follow up with an evaluation of predictive reliability in our work for NSGs.

6.5.1 SSG Experiment

In this prior work on Stackelberg Security Games (SSGs), participants in human subject experiments were asked to play a game called "The Guards and Treasures". For one experiment, participants in each round (for 11 rounds total) picked one of 24 targets based on its defender coverage probability, reward and penalty to the attacker, and reward and penalty to the defender. For each of these rounds, five coverage strategies were generated: three corresponding to other defender strategy algorithms and two corresponding to the QR and SUQR human behavior models whose weights were learned from a prior dataset consisting of 330 data points. While the previous work demonstrated that SUQR's prediction accuracy was better than QR, and SUQR had the best corresponding strategy performance compared to other algorithms, it was an implicit assumption that the behavior model with the best prediction accuracy would also perform the best in human subject experiments. If predictive reliability was actually poor, then it could have been the case that QR and its strategy would have performed the best in experiments.

6.5.2 SSG Predictive Reliability

For the following analysis, we confirmed that predictive reliability was strong for this SSG experiment; prediction accuracy was reliably correlated with actual performance. In the dataset we obtained from Ngyuen et al. (Nguyen et al., 2013) (which contained human subject attack data), we computed the predictive reliability over the QR and SUQR models. Because there were only two models in this correlation, the correlation output was either -1 (i.e., supports good predictive reliability) or +1 (i.e., supports poor predictive reliability). This analysis was done across 11 different rounds and for each of the three non-QR/SUQR test datasets. In Table 6.2, we show the predictive reliability of the QR and SUQR models in this SSG dataset. When MAE was used as

the error metric for each model, predictive reliability was 91%. In other words, 91% of correlations corresponded to prediction error being strongly inversely related to actual performance.

	MAE	AAE
Predictive Reliability	91%	85%

Table 6.2: Guards and treasures predictive reliability

6.5.3 NSG Predictive Reliability

In the following predictive reliability evaluation analysis for NSGs, we demonstrate that while predictive reliability is strong for SSGs, it is weak for NSGs; in an NSG setting, model prediction accuracy does not consistently correspond to actual performance.

We computed the predictive reliability on the NSG dataset using the three different error metrics: Maximum Absolute Error (MAE), Favored Path Maximum Absolute Error (FMAE), and Favored Path Average Absolute Error (FAAE). Table 6.3 displays the predictive reliability analysis results. While the predictive reliability results for the SSG dataset were strong, it is surprising that predictive reliability is extremely poor for this NSG dataset. This result certainly serves as a cautionary note against relying solely on prediction accuracy (as in previous work (Cui & John, 2014; Abbasi et al., 2015)) to identify the best human behavior models; with weak predictive reliability, even the best model in terms of prediction accuracy may actually perform very poorly when its corresponding strategy is tested against human subjects (either in the lab or in field experiments).

	MAE	FMAE	FAAE
Predictive Reliability	23%	24%	22%

Table 6.3: NSG predictive reliability

6.5.4 Training Set Size

While the predictive reliability for NSGs is poor, an obvious question to ask is "Was there enough training data?" For any learning task, it is important to have sufficient training data. While we do not have nearly as much training data (33 data points) as the prior SSG experiments (330 data points), it is important to ensure that our training set size is sufficiently large for reliable training. In this analysis, we examine the effects of training set size on the Maximum Absolute Error (MAE) rates of each NSG model. While we expect MAE to be unstable when there is very little data in the training set, as we add more data to the training set, we expect the error rates to eventually stabilize. It is at this stabilization point (marked by a training set size) that we can conclude whether we have trained our models on enough data or not. For example, if the stabilization point is at 48 data points, it would indicate that our current training set size (33) is not large enough, and any poor predictive reliability (as was previously demonstrated to be the case) could easily be explained by this deficiency in training set size.

As such, the following analysis illustrates the MAE rates of all six NSG models as a function of changes in the size of the training set. In Figures 6.3, 6.4, and 6.5, we show the results of this analysis on Graphs 7, 9, and 11 (respectively), where MAE is computed on the GSUQR2 testing set. Each line corresponds to a different model (e.g., QR-M refers to QR trained with Maximin data, SUQR-S refers to SUQR trained with GSUQR1 data), the Y-Axis displays the different MAE rates (higher is worse), and the X-Axis displays the change in training set size. While all the models appear to have different error rates and rates of convergence, most of the models appear to converge by the time 33 data points are introduced into the training set. Thus,

we conclude that we have trained our models with a sufficient number of data points, and the poor predictive reliability results cannot be attributed to the size of the training set.

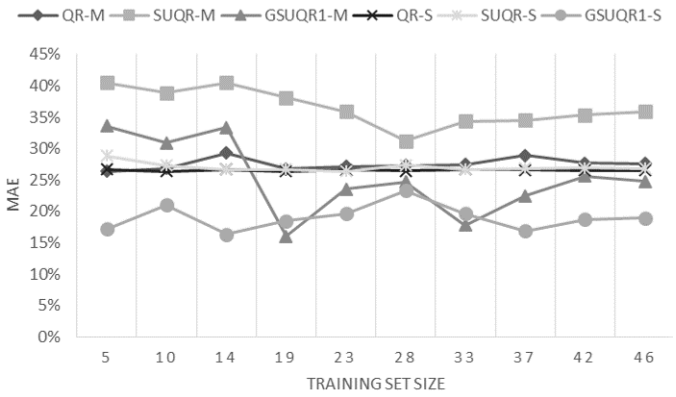


Figure 6.3: MAE as a function of training set size (GSUQR2 testing set, Graph 7)

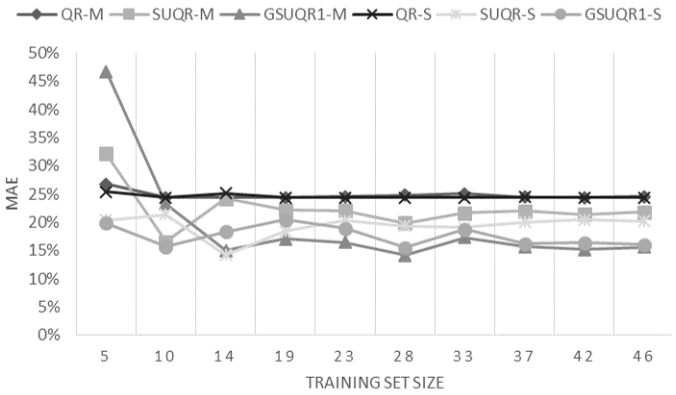


Figure 6.4: MAE as a function of training set size (GSUQR2 testing set, Graph 9)

6.6 Predictive Reliability Factors

6.6.1 Training Set Feature: EAS

In the following analysis for our NSG dataset, we quantify the key difference in our experiment's two training sets: Exposed Attack Surface (EAS), and we demonstrate that having a higher EAS

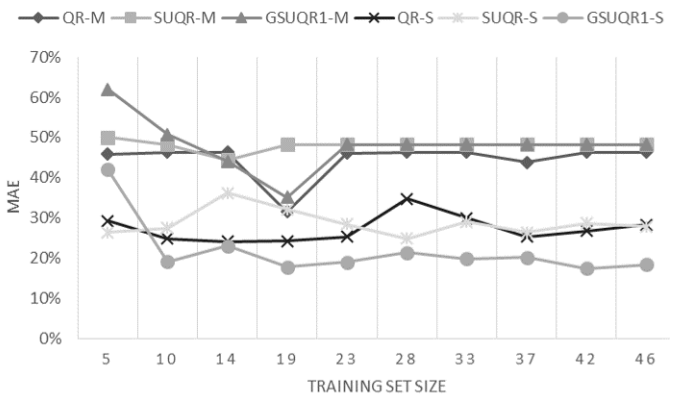


Figure 6.5: MAE as a function of training set size (GSUQR2 testing set, Graph 11)

score can lead to substantial improvements in predictive reliability. Note that both training sets in this analysis are of the same size.

6.6.1.1 Training Set Comparison

As discussed in section 6.4.2.4, the full experiment set is comprised of three separate experiment sets. Experiment set 2 consists of models trained on Maximin data (from experiment set 1), and experiment set 3 consists of models trained on GSUQR1-M data (from experiment set 2). We computed predictive reliability scores as a function of training set (either Maximin or GSUQR1-M) and prediction accuracy metric (Maximum Absolute Error (MAE), Favored Path Maximum Absolute Error (FMAE), and Favored Path Average Absolute Error (FAAE)), and we show those results in Figure 6.6. As is clear, there must be a significant difference in the two training sets; split solely on their training set, the predictive reliability doubles when models are trained on the GSUQR1-M dataset! While their sizes are roughly the same (about 47 participants), we examine one key difference in these datasets: exposed attack surface.

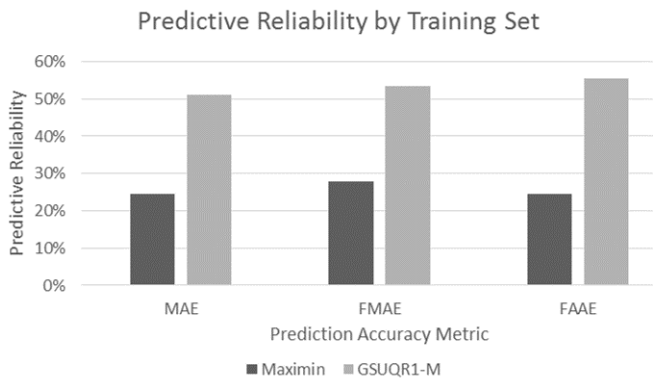


Figure 6.6: Predictive reliability as a function of training set and error metric

6.6.1.2 Exposed Attack Surface Analysis

Exposed Attack Surface (EAS), as defined in section 6.4.3.3, refers to the number of unique combinations of reward $\mathcal{T}(t_j)$ and path coverage probability $p_j(x)$ over all paths A in a graph g. Since we are interested in computing this score for an entire dataset (consisting of 15 graphs $g \in G$), we compute the sum of EAS scores across all graphs. Table 6.4 shows the sum of each training dataset's EAS score. While the Maximin dataset had 50 unique Exposed Attack Surfaces, the GSUQR1-M dataset had 86 unique Exposed Attack Surfaces. This is not surprising, as a Maximin strategy's only goal is to conservatively minimize the attacker expected utility across all paths; for 11 out of 15 graphs in the Maximin dataset, the EAS score is equal to 3 (the minimum given three targets of different reward value). In contrast, an SUQR-based strategy seeks to actively predict which paths an attacker will choose (based on a linear combination of path coverage, reward, and potentially other factors), and as a result, the resultant defender coverage strategy is more varied (and thus only 3 out of 15 graphs have the minimum EAS score of 3).

Based on this line of reasoning, we can view the EAS metric as a measure of dataset diversity. Since a diverse dataset would necessarily give more unique choices for attackers to make, we are

EAS-Sum	Maximin	GSUQR1-M
	50	86

Table 6.4: Training dataset comparison: sum of exposed attack surfaces

able to obtain more information on which choices are favored or not favored by attackers. A higher EAS score could indicate that a dataset is better for training than another dataset; indeed, our current results strongly suggest that when there is a substantial difference in EAS-Sum scores, there will also be a substantial difference in predictive reliability. However, these results do not mean that a high EAS score will result in 100% predictive reliability; if able to train on two datasets of equal size, it will likely improve predictive reliability to train on the dataset with the higher EAS score.

6.7 Graph Features and Their Impacts on Predictive Reliability

In addition to training set features, we also investigated the impacts that a graph's features may have on predictive reliability. For example, some graphs may be inherently more difficult to make predictions on than others, and it would be useful to characterize the factors that add to this complexity. Because this analysis is evaluating how a graph's features impact predictive reliability, the predictive reliability will be computed on a per graph basis. Figure 6.7 shows the predictive reliability scores for each graph, where each bin of three bars corresponds to a single graph, each bar corresponds to a prediction error metric, and the Y-axis corresponds to predictive reliability. As can be seen, the predictive reliability varies greatly as a function of the graph g. As such, it is logical to investigate what graph features could have led to such significant differences in predictive reliability.

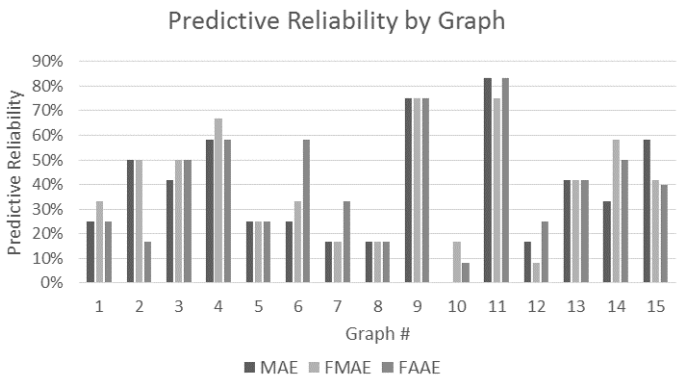


Figure 6.7: Predictive reliability as a function of graph

We analyzed the correlation between a graph's features and the predictive reliability score for that graph. Initially, we tested many different features such as graph size (i.e., the number of paths in the graph), number of edges, number of intermediate nodes, average path length, and the average in-degree (incoming edges) and out-degree (outgoing edges) of source, destination, and intermediate nodes. What we found, however, is that none of these had a strong, direct correlation with predictive reliability. For example, the lack of a strong correlation between graph size and predictive reliability states: "A graph's size does not impact the ability to make reliable predictions".

Upon further investigation, we found one interesting relationship: there is a strong correlation (+0.72) between poor predictive reliability and graphs with both a low to moderate average out-degree for source nodes (< 3) and a low to moderate number of intermediate nodes (≤ 6). While we could not find a correlation among the other features' values and the average out-degree of source nodes, we did find a strong correlation between the number of intermediate nodes and the average in-degree of destination nodes (-0.75). Informally stated, as the number of intermediate nodes increases, the number of edges going into destination nodes decrease. This balance is perhaps due to the edge limit imposed during graph creation. Regardless, when there are less

edges going into destination nodes (due to many intermediate nodes), it is likely easier for the defender to allocate resources which, in turn, reduces the number of good attack options for the attacker. If the attacker does not have many good attack options to choose from, they may act in a way that it is easier to predict by human behavior models.

Chapter 7

Protecting the NECTAR of the Ganga River: Explanation and Visualization of Game-Theoretic Inspection Strategies

The leather industry is a multi-billion dollar industry (Mwinyihija, 2011), and in many developing countries such as India and Bangladesh, the tanning industry is a large source of revenue. Unfortunately, the chemical byproducts of the tanning process are highly toxic, and the wastewater produced by tanneries is sent to nearby rivers and waterways. As a result, the Ganga River (along with many others) has become extremely contaminated, leading to substantial health problems for the large populations that rely on its water for basic needs (e.g., drinking, bathing, crops, livestock) (Institute, 2011). Tanneries are required by law to run wastewater through sewage treatment plants (STPs) prior to discharge into the Ganga. In many cases, however, the tanneries either do not own or run this equipment, and it is up to regulatory bodies to enforce compliance. However, inspection agencies have a severe lack of resources; the combination of the tanneries' unchecked pollution and the inspection agencies' failure to conduct inspections forced India's national environment monitoring agency to ban the operation of 98 tanneries near Kanpur, India with a further threat of closure for approximately 600 remaining tanneries (Jainani, 2015). It is

our goal to provide agencies with randomized inspection plans so tanneries reduce harmful effluents and an important facet of India's economy can operate in a sustainable fashion. However, we recognize that the intended users of these plans (inspectors with backgrounds in Hydrology and the physical sciences) have not used randomized schemes in the past and may not be familiar with game theory or optimization techniques. (Venkatesh & Davis, 2000) observed that user perceptions on ease of use and solution quality have a significant impact on user adoption of information technology; if the randomized solution cannot be understood by users (that are not experts in the randomization process), the solution risks not being adopted.

In this chapter, we introduce a new game-theoretic application, NECTAR (**N**irikshana for **E**nforcing **C**ompliance for **T**oxic wastewater **A**batement and **R**eduction)[1], that incorporates new models and algorithms to support India's inspection agencies by intelligently randomizing inspection schedules. While we build on previous deployed solutions based on Stackelberg Security Games (SSG) for counter-terrorism (Tambe, 2011) and traffic enforcement (Brown et al., 2014b), NECTAR represents the first security game application to directly address user adoption concerns by introducing a novel solution explanation component. Our SSG models are also the first to focus on the problem of pollution prevention by modeling the interaction between an inspection agency (the leader) and leather tanneries (many followers) — an interaction which poses a unique set of challenges. (i) Because there is a large disparity between the number of inspection teams and the number of tanneries, inspection plans must be efficient. (ii) We cannot assume that inspectors can catch 100% of violations. (iii) Inspectors must travel to the tanneries via a road network so solutions must be robust to delays (e.g., traffic). Finally, current fine policies

[1]Nirikshana, the Hindi word for inspect. As many mythological stories and even popular Bollywood songs attest, Ganga water is supposed to be NECTAR (or Amrit, the Hindi antonym of poison) which has inspired our project. The project name is intentionally chosen to fit this international and inter-cultural theme.

may not be sufficient to induce compliance, and (iv) it is important to investigate alternative fine structures.

NECTAR addresses these new challenges of tannery inspections. (i) Our SSG model captures the inspection process and accounts for two types of inspections: thorough inspections and simple (i.e., quick) inspections. While thorough inspections take longer to conduct (and thus less of them can be conducted), they are more likely to detect violations than simple, surface-level inspections which may only be able to check for obvious violations. To model the imperfect nature of these inspections, we (ii) introduce two failure rates: one for thorough inspections and one for simple inspections, with simple inspections failing at a higher rate. (iii) We also address the uncertainty involved with road networks by using a Markov Decision Process (MDP) that will represent and ultimately generate the game solution. In addition, (iv) we also investigate how tannery compliance is affected by two fine structures: fixed fines and variable fines, where the latter will result in larger tanneries receiving larger fines. Finally, (v) we introduce the explanation component framework and demonstrate how it can be applied to explaining NECTAR's solutions. For the evaluation of our model, we apply NECTAR to a real-world network of tanneries in Kanpur, India, and we evaluate the quality of NECTAR's generated solutions. We also piloted a survey among the study team and affiliates in order to receive initial feedback on the explanation component such that we can further refine our explanations and conduct full-scale human subject experiments. We also demonstrate how NECTAR's solutions can be visualized via a Google Earth overlay that we anticipate will improve ease of use and, ultimately, odds of user adoption.

7.1 Model

In this section, we model this pollution prevention problem as a defender-attacker Stackelberg Security Game (SSG). The task of the defender is to send resources to different tannery sites (i.e., the multiple adversaries) on a road network. The defender must devise a patrol strategy to maximize compliance among a number of sites (each site denoted by l), where each site has a number of factories f_l and each site's compliance cost increases with the number of factories. In addition, the defender must take into account the time it takes to travel to and inspect each site. We model the road network as a graph where the nodes represent sites and the edges represent the roads connecting each site. Each edge also has a cost, e_{ab}, associated with it that represents the travel time from a site a to another site b. Using publicly available data regarding tannery locations in Kanpur, we constructed a graph consisting of 50 sites.

The defender has two types of resources: r_1 number of thorough inspection resources and r_2 simple inspection resources. For thorough inspection resources, the inspector conducts a detailed inspection that takes i time units. We model imperfect inspections such that even if a violation exists, the inspectors will fail to detect it with a low probability γ_1. For simple inspection resources, the inspector will conduct a superficial inspection that takes d time units. Since the inspection is not detailed, simple inspection resources will not detect anything but obvious violations. Thus, such resources have a higher probability of failure γ_2. Each of the defender's resources (thorough and simple) have a maximum time budget, t_1 and t_2 respectively, to conduct inspections and travel to sites.

In the SSG framework, the defender will commit to a randomized patrol strategy (a mixed strategy) which is a probability distribution over the executable daily inspection patrols (the pure

strategies for all resources). The adversaries (the sites) can fully observe the defender's mixed strategy and know the probability of being inspected by a thorough inspection team or a simple inspection team on a given day. Formulating the mixed strategy requires enumerating all feasible pure strategies for the defender. However, this approach is impractical for two main reasons: (1) for any realistically-sized patrolling problem, the defender pure strategy space is so large that it cannot fit into memory. For example, with our Kanpur graph of 50 tanneries, only one defender resource, and a time horizon of 10 hours, the pure strategy space size would be too large to enumerate (approximately 50 choose 10). Therefore, we adopt a compact representation (a transition graph) that will allow our approach to scale to large problem sizes. (2) Inspectors must travel to sites via a road network (with potential delays), and the corresponding uncertainty cannot be handled by a standard SSG formulation. Rather than reasoning about mixed strategies, we instead use the compact representation to reason about spatio-temporal flow through a transition graph. To account for stochasticity and uncertainty in the outcome of actions, we use a Markov Decision Process (MDP) to represent the defender's inspection patrolling problem. We can solve the corresponding linear program (LP) to compute the optimal inspection strategy, i.e., the optimal MDP policy.

7.1.1 Compact Game Representation: Transition Graph

Brown et al. also faced the challenge of large state spaces for a traffic enforcement domain (Brown et al., 2014b). Since their game also takes place on a road network, there are sufficient similarities between our approach and theirs to apply their techniques, based on transition graphs, to improve the scalability of our model.

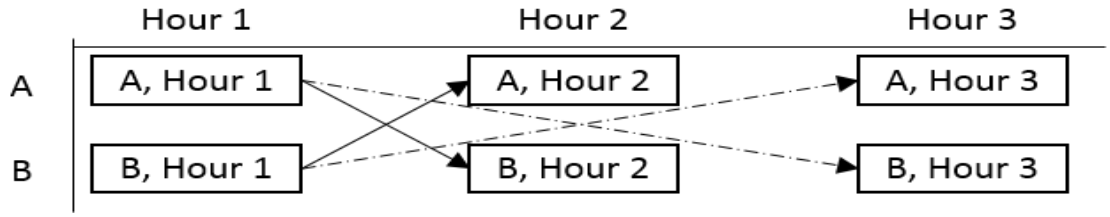


Figure 7.1: Illustrative MDP example

Instead of enumerating an exponential number of pure strategies, we need only enumerate a polynomial number of states and edges in the transition graph. We then compute the optimal probability flow (as seen in the next section), also called a marginal coverage vector, and sample from the vector to create inspection schedules. As the defender resource types (thorough and simple) have different time constraints, each has its own transition graph.

We discretize time into a granularity of h hours. In the thorough inspection resource transition graph, a vertex is added for each site l every h hours until the resource time budget t_1 has been expended. Similarly for the simple resource's transition graph, vertices are added until the time budget t_2 has been expended.

7.1.2 MDP Formulation

We present an MDP $\langle S, A, T, R \rangle$ to incorporate uncertainty into the transition graph. An example MDP is shown in Figure 7.1 to illustrate these definitions.

- S: Finite set of states. Each state $s \in S$ is a tuple (l, τ), where l is the site that the resource is located, and τ is the current time step. For example, an inspector at site A at hour 1 is represented as $s_{A,1}$. Each vertex in the transition graph corresponds to a state s.

- A: Finite set of actions. $A(s)$ corresponds to the set of actions available from state s (i.e., the set of sites reachable from l) that the resource can travel to and inspect. For example,

at site A at hour 1, the only available action is to move to site B (i.e., the solid arrow from A to B in Figure 7.1).

- $T_1(s, a, s')$: Probability of an inspector ending up in state s' after performing action a while in state s. Travel time and inspection time are both represented here. As a simple example, there could be probability 0.7 for transition $T_1(s_{A,1}, a_B, s_{B,2})$: a transition from site A at hour 1 to move to and inspect site B will, with a probability of 0.7, finish at hour 2 (a travel + inspection time of 1 hour). The dashed lines in Figure 7.1 represent the remaining probability (0.3) that the same action will instead finish at hour 3 (due to a delay). Note that the two resource types have separate transition functions due to the difference in action times (i for thorough inspection resources and d for simple inspection resources).

- $R(s, a, s')$: The reward function for ending in state s' after performing action a while in state s. As we are interested in the game-theoretic reward, we define the reward in the LP and define R = 0 $\forall s, a, s'$.

7.2 Inspection Patrol Generation

We provide a linear program (LP) to compute the optimal flow through the MDP (i.e., the transition graph with uncertainty). By normalizing the outgoing flow from each state in the MDP, we obtain the optimal MDP policy from which we can sample to generate dynamic patrol schedules. In the following LP formulation, we make use of the following notation. A site l has a number of factories f_l, and if a site is caught violating during an inspection, they receive a fine, α_l. On the other hand, if a site wants to remain in compliance, they will need to pay a compliance cost β for each factory (total cost = βf_l). We represent the expected cost for each site l as v_l. As defined in

the following LP, the expected cost corresponds to the lowest of either the site's expected fine or the site's full cost of compliance; we assume that these adversaries are rational and that they will choose to pay the lowest of those two values (expected fine or cost of compliance). Finally, we denote as S_l the set of all states that correspond to site l (i.e., all time steps associated with site l).

As discussed in the transition graph definition, the optimal flow through the graph corresponds to the optimal defender strategy, and that flow is represented by a marginal coverage vector. We denote the marginal probability of a resource type i (either thorough or simple inspection team) reaching state s and executing action a as $w_i(s, a)$. We also denote, as $x_i(s, a, s')$, the marginal probability of a resource type i reaching state s, executing action a, and ending in state s'.

$$\max_{w,x} \sum_{l} v_l \tag{7.1}$$

$$s.t. x_i(s,a,s') = w_i(s,a)T_i(s,a,s'), \forall s,a,s',i \tag{7.2}$$

$$\sum_{s',a',i} x_i(s',a',s) = \sum_{a,i} w_i(s,a), \forall s,i \tag{7.3}$$

$$\sum_{a,i} w_i(s_i^+,a) = r_i \tag{7.4}$$

$$\sum_{s,a,i} x_i(s,a,s_i^-) = r_i \tag{7.5}$$

$$w_i(s,a) \geq 0 \tag{7.6}$$

$$v_l \leq \alpha_l(p_{l1} + p_{l2}) \tag{7.7}$$

$$p_{l1} = (1-\gamma_1) \sum_{s \in S_l, a} w_1(s,a) \tag{7.8}$$

$$p_{l2} = (1-\gamma_2) \sum_{s \in S_l, a} w_2(s,a) \tag{7.9}$$

$$p_{l1} + p_{l2} \leq 1 \tag{7.10}$$

$$0 \leq v_l \leq \beta f_l \tag{7.11}$$

The objective function in Equation 1 maximizes the total expected cost over all sites. Constraints 2-5 detail the transition graph flow constraints (for thorough inspections and simple inspections). Constraint 2 defines that x is equal to the probability of reaching a state s and performing action a multiplied by the probability of successfully transitioning to state s'. Constraint 3 ensures that the flow into a state s is equal to the flow out of the state. Constraints 4-5 enforce that the total flow

in the transition graph, corresponding to the number of defender resources r_i, is held constant for both the flow out of the dummy source nodes s_i^+ and into the dummy sink nodes s_i^-.

Constraint 7 constrains the expected cost for site l. Constraints 8-9 define the probability of successfully inspecting a given site l and is the summation of probabilities of reaching any of l's corresponding states (thus triggering an inspection) and taking any action a. Note that the failure probability γ means that even if a violating site is inspected, there may not be a fine issued. Constraint 10 limits the overall probability of a site being inspected. If a site is visited by both thorough and simple inspection resources, the site will only have to pay a fine, at most, once. Constraint 11 defines the bounds for the adversary's expected cost; if the adversary's expected cost is at the upper bound ($v_l = \beta f_l$), we assume that the adversary would prefer to have a positive public perception and choose to comply rather than pay an equivalent amount in expected fines.

7.3 Explaining NECTAR Solutions

For NECTAR to be adopted as an inspection planning tool, the end users must have a high degree of confidence that the solutions computed by the system are feasible and efficient patrolling strategies. In work on the Technology Acceptance Model (TAM), (Venkatesh & Davis, 2000) found that user perceptions of the solution quality and ease of use significantly influenced user acceptance of four different information technology systems. In our context, the end users will likely be inspectors and managers with degrees in the physical sciences. However, it is unlikely that they will also be experts in game theory and optimization; the NECTAR system may seem to function as a black box that generates strategies for opaque reasons. To address this key challenge

for adoption, we have developed an explanation module for NECTAR that is designed to make the solutions more transparent to the users, ultimately building trust in the system.

7.3.1 Simplifying Explanations

The main challenge in explaining the solutions to users is that the optimal policy is very complex: it is the solution to an MDP that specifies inspection probabilities for multiple locations and time steps. In addition, the optimal policy may be the result of complex tradeoffs between many different priorities and constraints. We have designed our explanations to focus on the most important aspect of the solution: *how frequently each site will be inspected*. This allows for simpler explanations that abstract away many of the details of time and real-world uncertainties that are captured in the complete NECTAR model in Section 7.2.

Our simplified model for explanation focuses on the aggregate probability that each site will be inspected: $\hat{x_l}$, which is the sum of incoming flow into the site, $\sum_{s \in S_l, a} w(s, a)$. The defender's expected utility in this case is the sum of the expected fines ($\alpha_l \hat{x_l}$) over all sites, and the optimal solution maximizes this quantity. An additional advantage of this approach is that representing the solution in terms of the coverage probabilities for a set of targets is common to many of the Stackelberg Security Games that have been presented in the literature, even though the details of the resources and scheduling constraints vary depending on the specific domain. As such, our method for generating explanations can be applied with very little modification to other existing decision support systems based on security games.

```
It is not helpful to inspect L3 with more probability (i.e., coverage) because it would not
improve the total expected fine (summed over all sites).
If we did increase coverage by 0.1 on site L3, then its expected fine amount would increase by
6.0, and it would be less prone to violations.
However, we can only conduct 2 inspections; we can only allocate a maximum coverage of 2. Since
we have now allocated a coverage of 2.1, we must remove 0.1 coverage from another site.
If we decreased coverage by 0.1 on site L4, it would have the least negative impact on the total
expected fine.
L4's expected fine amount would decrease by 9.0, and it may be more prone to violations.
However, the best site from which we took coverage, L4, is more valuable to cover because it is a
site with a larger number of factories than site L3 and, with the same amount of coverage,
expected fines are larger for sites with more factories.
(L3 has 2 factories, and L4 has 3 factories.)
If the proposed coverage changes were enacted, the total expected fine amount across all sites
would decrease to 247.0, which is worse than the current optimal solution's value of 250.0.
```

Figure 7.2: Example output from NECTAR's explanation component

7.3.2 Explanation Overview

Our explanations are based on the paradigm of "what-if" analysis. We allow users to ask specific questions about potential modifications to the solution calculated by the system, such as increasing or decreasing the probability of visiting a specific location. The system generates a series of statements that describe the implications of this change and show how it leads to a worse solution overall. We show example output from NECTAR's explanation module in Figure 7.2 in response to a user query: "Why isn't there 10% more coverage on site L3?" The explanation component analyzes this hypothetical scenario, and at key points in its internal evaluation, outputs explanatory statements to the user.

The key ideas that must be explained to the user include (1) there are tradeoffs due to the overall resource limitations, and adding coverage in one location means removing it from another location, (2) even if we assume the best case scenario for the modification (e.g., removing coverage from the least important location), the overall solution quality does not improve, so (3) NECTAR has already generated a solution that optimally balances these tradeoffs within the limitations of the resources.

There are a limited number of different ways for the user to modify the solution, and a few general types of arguments can be used to explain why the modification does not improve solution quality. For each of these possible "what-if" scenarios, we have developed an *explanation template* that has the basic text and structure of the argument. However, the details of the argument are problem-specific so they must be generated by the system each time a user asks for an explanation. The explanation shown in Figure 7.2 is an example of a template that has been instantiated with these details.

7.3.3 Automating Explanations

We now describe how the system automatically generates explanations for questions of the form "Why is target l covered with $\hat{x}_l$ probability?" There are two versions of this question for increasing or decreasing the probability, but they are very similar so we focus on the case of increasing the coverage on l. Consider the scenario of allocating Δ more coverage to some $l \in L$, which is currently assigned coverage $\hat{x}_l$. The system makes this change to generate the modified coverage distribution $\hat{x'}$. However, this coverage change may violate the constraint that the system cannot change the overall number of resources; the sum of the coverage in $\hat{x'}$ should be the same as in $\hat{x}$. The system checks for any violations of these constraints and then attempts to "repair" the solution in the way that is best for the defender. Based on the outcome of this repair operation, the system presents a final explanation comparing the outcomes of the original solution and the modified one to demonstrate that the modification does not result in an improvement for the defender.

The details for how the explanation system repairs violations in coverage overallocation are shown in Algorithm 2. Note that the notation $explain$ refers to filling in a template explanation

with specific details as needed. Here the system needs to both repair the solution and explain to the user why the violations is resolved in this way. It is important that the repaired solution represents the best case for the defender in order for it to be convincing to the user. For example, if the modified solution requires too many resources (e.g., as a result of adding coverage to a location), the way to resolve this is to remove coverage from another target ($l' \in L, l' \neq l$). Logically, this coverage should be removed from the least-harmful target and not a more valuable target. Our system systematically considers each target and picks the one where reducing the coverage is least harmful to the defender. This rationale is communicated to the user in the explanation.

Algorithm 2 Explanation System: Resolve Target Coverage Overallocation

1: **function** RESOLVE-OVERALLOCATION($l, \hat{x'}, L$)
2: $\Delta' \leftarrow$ ComputeOverallocation;
3: $EU_d^* \leftarrow -\infty$;
4: $l^* \leftarrow null$;
5: **for** each $l' \in L, l' \neq l$ **do**
6: Reduce coverage on l' by Δ';
7: Compute adversary best response;
8: Compute $EU_d^{\hat{x''}}$ given adversary best response;
9: **if** $EU_d^{\hat{x''}}$ ¿ EU_d^* **then**
10: $EU_d^* \leftarrow EU_d^{\hat{x''}}$;
11: $l^* \leftarrow l'$;
12: **end if**
13: Revert coverage on l';
14: **end for**
15: $explain$ Coverage on l^* could be reduced with the least harm;
16: Reduce coverage on l^* by Δ';
17: $explain$ State changes in attacker response;

The system first computes the amount of coverage that is overallocated, Δ', that must be removed from another target $l'(l' \neq l)$. The impact on the defender's expected utility for removing this amount of coverage is assessed for each target. This is done by temporarily reducing the

coverage, generating the new coverage distribution $\hat{x''}$, computing the adversary's best response to this coverage, and calculating the expected utility for the defender in this case. In our domain, this corresponds to computing the change in expected fine ($\alpha_{l'}\hat{x_{l'}}$) for each target l'. Once the best case target is found, the explanation is given to the user for why decreasing coverage on l^* is the best case. Finally, the system explains how the attacker's best response changes in this best-case scenario.

7.4 Evaluation

In order to explore the strategic tradeoffs that exist in our model of the tannery domain, we ran a series of experiments on our Kanpur tannery graph. For each experiment, we generated 3 distinct patrolling strategy types. 1. NECTAR's strategy, 2. the Uniform Random (UR) strategy: at each time step, every site has an equal probability of being chosen, and 3. an Ad-Hoc (AH) strategy: a deterministic strategy where sites are visited in numerical order (by ID number).

In order to analyze how different resource types affect performance, for each experiment we generated six defender strategies: the first three (NECTAR, UR, AH) correspond to when the defender had twice as many simple inspection resources as thorough inspection resources, and the last three (again NECTAR, UR, AH) correspond to when the defender had no simple inspection resources.

In addition to running experiments where each site l has the same fine (α), we ran a set of experiments where each site's fine α_l was: $\alpha_l = \alpha f_l$ or, in other words, the fine amount is a constant α multiplied by the number of factories f_l at that site – sites with more factories will

be penalized for violations more harshly than sites with fewer factories. As this type of analysis requires heterogeneous sites, we randomize the number of factories at each site.

Ultimately, we are interested in inducing compliance in sites, and for our performance metric, we compute the number of sites that would be in full compliance given the defender strategy (i.e., how many sites' cost $v_l = \beta f_l$). The maximum number of sites in compliance for each experiment is 50 (i.e., the number of sites on our graph). The default parameter values for each experiment (unless otherwise specified) are listed in Table 7.1.

Variable	Value
Compliance Cost β	10
Fixed Fine Amount α	100
Number of Factories at Each Site f_l	2-5
Number of Simple Inspections r_2	2
Number of Sites	50
Number of Thorough Inspections r_1	1
Patrol duration (hours) t_1, t_2	6
Simple Inspection Failure Rate γ_2	0.6
Thorough Inspection Failure Rate γ_1	0.1
Time granularity (hours) h	1
Time steps to complete simple inspection	1
Time steps to complete thorough inspection	2
Variable Fine Amount α_l	30

Table 7.1: Default experiment values

Fixed Fine Amount In Figure 7.3, we analyze the effects of the fixed fine amount α on the number of complying sites. The x-axis shows the fixed fine amount, and the y-axis shows the number of sites that are complying (i.e., $v_l = \beta f_l$).

From the figure, we observe the following trends: (1) the NECTAR strategy does not achieve any compliance until the fine amount is 350, with all sites in compliance at 400. This is due to the objective function attempting to maximize expected cost over all sites simultaneously with a homogeneous fine. (2) While the UR and AH strategies achieve compliance from some of the

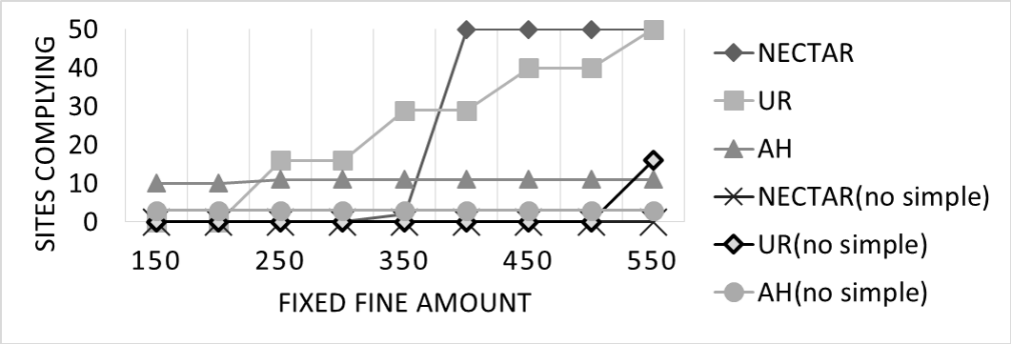


Figure 7.3: Fixed fine: number of sites in compliance

sites for smaller fine amounts, they do not achieve compliance for all of the sites as quickly as the NECTAR strategy. (3) The inclusion of simple inspection resources improve performance for every strategy as expected.

Variable Fine Amount In Figure 7.4, we analyze the effects of the variable fine amount α_l on the number of complying sites. The x-axis shows the variable fine amount, and the y-axis shows the number of sites in compliance (i.e., $v_l = \beta f_l$).

From the figure, we observe the following trends: (1) both the NECTAR and UR strategies achieve compliance from all sites for the same variable fine amount; (2) as the fines are not homogeneous for all sites, it is beneficial for NECTAR to try to maximize expected cost in sites with many factories first (unlike with the fixed fine, there is no "water filling" effect); the NECTAR approach achieves faster compliance from larger sites, and (3) the NECTAR achieves compliance from the most sites at every point.

Number of Resources: Variable Fine In Figure 7.5, we analyze the effect of the number of resources when there is a variable fine amount α_l on the number of complying sites. The x-axis shows the number of thorough inspection resources, r_1 (for the strategies with simple inspection

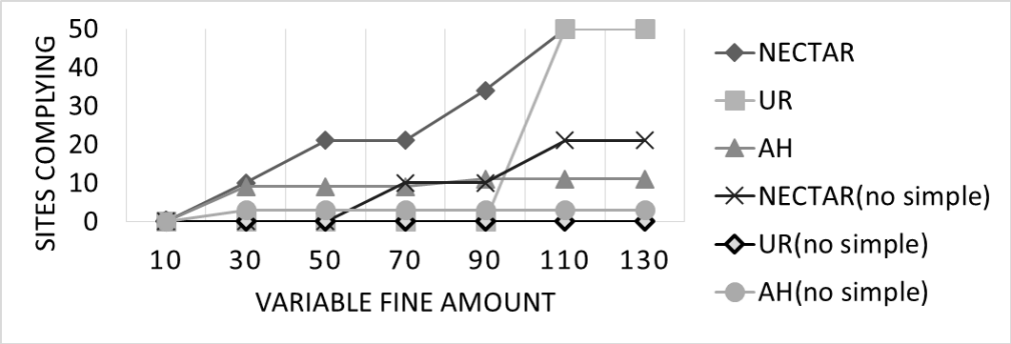


Figure 7.4: Variable fine: number of sites in compliance

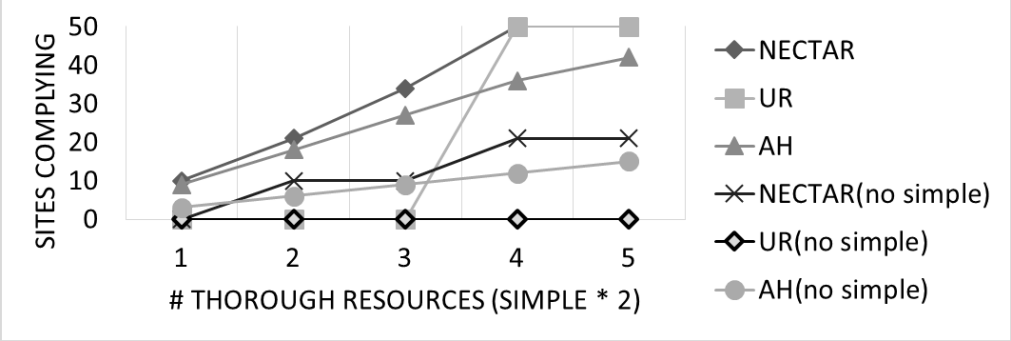


Figure 7.5: Number of resources: variable fine: number of sites in compliance

resources, the number of simple inspection resources is $r_2 = 2 \times r_1$), and the y-axis shows the number of sites that are complying (i.e., $v_l = \beta f_l$).

From the figure, we observe the following trends: (1) the NECTAR and AH strategies achieve compliance from some sites even with few thorough inspection resources, but NECTAR achieves compliance from the most sites at every point, (2) both the NECTAR and UR strategies achieve compliance from all sites for the same number of thorough inspection resources, and (3) even when there are many resources, the AH strategy does not achieve compliance from all sites.

Patrol Duration: Variable Fine In Figure 7.6, we analyze the effects of the patrol duration when there is a variable fine amount α_l on the number of complying sites. The x-axis shows the patrol duration, and the y-axis shows the number of sites that are complying (i.e., $v_l = \beta f_l$).

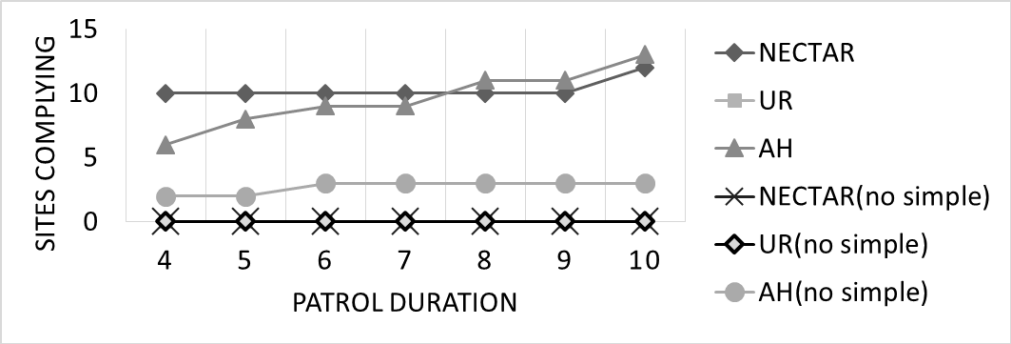


Figure 7.6: Patrol duration: variable fine: number of sites in compliance

From the figure, we observe the following trends: (1) while the NECTAR strategy performs the best for lower values of patrol duration, it is eventually outpaced by the AH strategy, (2) regardless of the strategy, there is not much change in the number of sites in compliance as a function of patrol duration. For this experiment, the default values for the other parameters result in a low compliance rate regardless of the value of the variable of interest, and (3) having simple inspection resources is helpful for the NECTAR and AH strategies, but it is not very helpful for the UR strategy.

7.5 Explanation Pilot Survey

With the comparative explanation component still in its infancy, we piloted a survey among our affiliates. The goal was to acquire a baseline measurement of how explanations could increase trust in security game decision aids such as NECTAR. In order to refine our methodology for future, full-scale human subject experiments, we also wanted to receive feedback on explanations of varying verbosity and on the survey itself. Pilot respondents were randomly assigned to complete one of three different survey versions, where each version contained explanations of a single verbosity (i.e., level of detail) type: low, medium, or high.

In the survey, we presented a simplified NECTAR scenario consisting of the simplified model (as presented in section 7.3.1), a sample problem, and an optimal coverage strategy generated by the NECTAR decision aid. Before any sample explanations were presented, a baseline questionnaire assessed the respondent's level of trust, perceived ease of use, and understanding of the solution. Next, we presented two sets of sample question (e.g., "Why isn't there more coverage on site L4'?"), explanation (e.g., Figure 7.2), and post-explanation questionnaire. Responses were provided on a 5-point likert scale ranging from 1="Strongly disagree" to 5="Strongly agree". As a result of the ordering of these measurements, we would expect increases in the respondent's level of trust to be a result of the explanations. At the end of the survey, we also presented a set of open-ended questions to elicit more detailed feedback.

For this analysis, we evaluated changes in trust as a function of explanation via the following pair of questions: "I trust the decision aid to make the best decisions." and "In the future, if there were explanations provided, I would trust the decision aid to make the best decisions." Out of the 12 respondents, 7 (2 in the low verbosity group, 2 in the medium verbosity group, and 3 in the high verbosity group) expressed an increase in trust in the decision aid, 4 (1 in the low verbosity group, 2 in the medium verbosity group, and 1 in the high verbosity group) already trusted the decision aid and did not express an increase in trust, and only 1 (in the low verbosity group) expressed neither trust nor distrust in the decision aid before and after the explanations.

In the open-ended question section, 75% of respondents in the low verbosity group and 50% in the medium group indicated that more quantitative information would be even more convincing of the solution's optimality. As such, future experiments, focusing on improving user understanding and acceptance, will test explanations containing more quantitative information in an effort to identify the optimal balance between verbosity and cognitive load.

7.6 Discussion and Results Visualization

Based on these simulations, we make the following conclusions: (1) when the number of resources or variable fine amount is the experiment variable, NECTAR makes the most efficient use of its resources, regardless of whether it is using only thorough inspections or a combination of simple and thorough inspections; (2) having more resources (more manpower) is more useful than increasing the duration of patrols (longer work hours). This is intuitive when considering that each resource must spend time traveling to each site; two resources can each cover a separate sub-section of the graph whereas one resource will be forced to spend more time traveling. Finally, (3) using a variable fine (in which sites are fined according to their number of factories) leads to better compliance rates. This observation makes sense when put in the context of our LP's objective function: maximize the sum of the expected costs v_l over all sites.

Since our goal is to assist inspection agencies with patrol planning, it is useful to visualize the proposed inspection patrols. In Figure 7.7, we show a simple graph and strategy visualization in Google Earth (a visualization for the Kanpur area is shown in Figure 7.8). The lines represent edges on the graph (i.e., straight line connections between sites). Each line also has a time step and a coverage probability associated with it, where the probability represents the value of the MDP's transition function, $T(s, a, s')$. In other words, this information answers the question: "If the defender resource starts at site l at the beginning of this edge at time step t (i.e., state s), what is the probability that the defender resource will take action a and arrive at site l', at the end of this edge, in a following time step t' (i.e., state s')?" By clicking on an edge, the user can call up the aforementioned defender strategy information (shown in Figure 7.7).

Figure 7.7: Visualization example

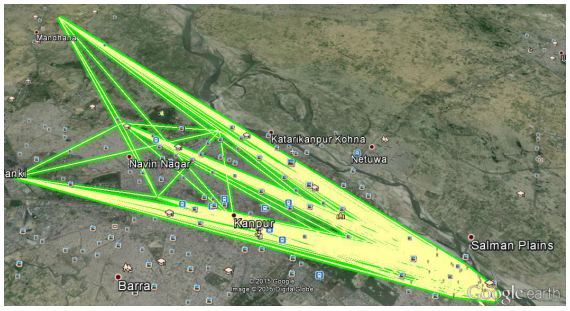


Figure 7.8: A Kanpur inspection patrol plan

Chapter 8

Conclusions and Future Directions

Even though many approaches have been proposed to model poaching behavior, nearly all of these works fail to simultaneously work with real-world data, empirically compare their work to others, and also field test their approaches in the real world. This thesis details novel research conducted in Uganda's Queen Elizabeth Protected Area (QEPA) where we successfully developed poacher behavior models from real-world data, extensively compared them to other baselines, and tested them in the largest and longest field tests of machine learning-based predictive models conducted in this domain to date. Additionally, my work has demonstrated the need for conducting field tests of any intervention and also discussed the need for interpretable interventions.

The first major contribution of this thesis is to present INTERCEPT, a paradigm shift from complex logit-based models to simpler decision tree-based models. While the previous state-of-the-art, CAPTURE, represented the latest in a long line of behavioral game theory research, it suffered from poor performance and other critical limitations that precluded its actual deployment in the field. Indeed, in the process of conducting the most extensive empirical evaluation in the AI literature of one of the largest poaching datasets, I showed a surprising result: INTERCEPT,

based on a simpler model, significantly outperformed the more complex CAPTURE model. Furthermore, decision trees were specifically chosen due to the fundamental requirement of fast execution — a key limitation of previous logit-based models such as CAPTURE. Additionally, as a first for behavior modeling applications applied to this domain, I presented results from a month-long test of the model by rangers in QEPA where rangers found and confiscated an active snare and almost a dozen additional snares, including multiple elephant snares, before they were deployed. Given that the rangers also found a poached elephant, their finding and confiscating of new elephant snares before they were deployed is significant; this research has potentially saved the lives of elephants and other animals in QEPA.

The second major contribution is to present a hybrid spatio-temporal model to predict wildlife poaching threat levels. On real-world historical data from QEPA, the hybrid model achieves significantly better performance than prior work. Additionally, I validated the hybrid model via designing and deploying an extensive eight-month field test in QEPA where rangers patrolled approximately 452 kilometers across QEPA — the largest field test to date of machine learning-based models in this domain. On the data collected from our field test, I demonstrated that our model successfully differentiated between areas of high and low snaring activity with statistical significance. These findings demonstrated that the model's predictions are selective and also that its superior laboratory performance extends to the real world.

The third major contribution built upon previous work that successfully used decision tree ensembles to predict wildlife poacher behavior and analyzed how a poaching detection model would react to changes in ranger patrol strategies. The detection model bagging ensemble, which newly incorporated ranger's current patrol effort as a feature, outperformed standard baselines (including other ensemble types) in terms of predictive capability. Additionally, I demonstrated

that the detection model reacted reasonably to changes in ranger patrol effort. When ranger effort increased in particular areas, the detection model mostly predicted that there would be increases in the number of poaching activity detections (with statistical significance). Similarly when ranger effort decreased, the detection model mostly predicted that there would be decreases in the number of poaching activity detections (also with statistical significance). Combined with the previous chapters' results that validated bagging ensembles' predictive performance in the field, these results demonstrated that bagging ensembles can be used as input to patrol generation frameworks. Patrol generation frameworks, using this detection model, could determine the optimal patrolling strategy for rangers to maximize their detections of poaching and thus save more animals from poaching.

Fourth, this thesis called into question the issue of predictive reliability — the implicit assumption that a model's prediction accuracy strongly correlates with the performance of its corresponding defender strategy. If that assumption does not hold, then the use of the corresponding strategy could lead to substantial losses for the defender. I first demonstrated that predictive reliability was strong for previous Stackelberg Security Game experiments. I also ran my own set of human subject experiments in such a way that models were restricted to learning on dataset sizes representative of real-world constraints. In the analysis on that data, I demonstrated that predictive reliability was extremely weak for Network Security Games. Following that discovery, however, I identified key factors that influenced predictive reliability results: exposed attack surface of the training data and graph structure.

Finally, this thesis introduced a new game-theoretic application, NECTAR, which aimed to aid inspection agencies in scheduling inspections of tanneries along vital rivers and waterways.

NECTAR provided randomized inspection policies and schedules that incorporated various real-world uncertainties and constraints, and NECTAR also generated explanations and visualizations in the hopes of improving users' perceptions of solution quality and ease of use to support user adoption. NECTAR was proposed to decision makers in governments, pollution control boards, and funding agencies that cover cleaning of large river basins. While field inspectors have not used randomized inspection schemes in the past, they have given positive feedback on this approach, and I anticipate that by allowing them to ask "what-if" questions via the explanation component and by visualizing patrols, they will be more likely to understand NECTAR's solutions and will thus be more likely to adopt the NECTAR approach.

Although this thesis presented novel research in terms of field testing behavior models in the real world, there are certainly areas of future work that exist. First, the natural next step for this research is to generate patrols based on the adversary model's predictions. As noted in this thesis, however, any patrol generation framework (i.e., the intervention) cannot be assumed to generate solutions of similar quality to the prediction model. Field testing of a patrol generation framework would thus be a necessary component to fully evaluate that framework's quality. As also mentioned, interventions should also fully consider user adoption attitudes and make earnest efforts to provide easy-to-understand interpretations of their proposed solutions. For a patrol generation framework, this could correspond to providing automated explanations of why a particular patrolling regime is optimal and will result in increased patrol efficiency.

Chapter 9

Appendix

9.1 Field Test Analysis: Three Experiment Groups

For the eight-month field test in QEPA, we also conducted a three experiment group analysis on those field test results. These three experiment groups corresponded to our model's attack prediction rates from November 2016 – June 2017: High (group 1), Medium (group 2) and Low (group 3). Areas that had an attack prediction rate of 66% or greater were considered to be in a high area (group 1); areas with a rate of between 33% and 66% were in group 2; areas with less than a 33% rate were in group 3. The same areas presented in Section 4.4 are used in this three-group analysis as well. Group memberships are presented in Table 9.1, group results are presented in Table 9.2, and statistical significance results are presented in Table 9.3.

As can be seen in Table 9.1, the high group loses 2 areas to the medium group and then comprises even less of the total experiment areas. However, as shown in Table 9.2, the remaining

Experiment Group	Final Group Memberships
High (1)	3 (11%)
Medium (2)	7 (26%)
Low (3)	17 (63%)

Table 9.1: Patrol area group memberships

Experiment Group	Observation Count(%)	Effort(%)	CPUE
High (1)	15 (79%)	72.59 (14%)	0.21
Medium (2)	2 (11%)	207.95 (41%)	0.01
Low (3)	2 (11%)	224.12 (44%)	0.01

Table 9.2: Field test results: observations

Group Comparison	Mean 1(std)	Mean 2(std)	p-value	Cohen's d
High to Medium	**0.21(0.58)**	0.01(0.14)	p<0.0001	0.62
High to Low	**0.21(0.58)**	0.01(0.09)	p<0.0001	0.66
Medium to Low	**0.01(0.14)**	0.01(0.09)	p<0.5	0.01

Table 9.3: Field test results: statistical significance results

areas contained all of the observed attacks for the high group anyway. Because the high group consisted of fewer areas, less allocated effort, but still contained the same number of attacks, the CPUE increases to 0.21. Moreover, when looking at the statistical significance in Table 9.3, it is clear that there is strong statistical significance and an even stronger effect size (with a Cohen's d value of 0.62 and 0.66) than was seen in Section 4.4. It is worth noting, however, that there is no observed difference between the medium and low groups with these results; the medium and low groups had a similar amount of effort and the same number of observed attacks, and there is no statistically significant differences between those two groups. These results suggest that the decision tree can successfully differentiate between high and low amounts of poaching activity, but perhaps not as successfully for a three group case (high, medium, and low). Because the high group results are stronger with a higher threshold for the high group (66% instead of 50%), it may be the case that the optimal threshold for determining an area of higher poaching activity may be closer to this 66% range than 50%.

Effort Change	Neg to Pos	Pos to Neg	No Change (Pos)	No Change (Neg)
Low to High	28	28	110	1294
High to Low	2	232	71	757

Table 9.4: One-month time scale prediction changes as function of previous effort

Effort Change	Neg to Pos	Pos to Neg	No Change (Pos)	No Change (Neg)
Low to High	62	33	101	1265
High to Low	1	169	131	760

Table 9.5: Three-month time scale prediction changes as function of previous effort

9.2 Attacker Adaptability Analysis

For this analysis, we present the changes in (1) the model's detected attack predictions and (2) the model's detected attack prediction probabilities when the effort in the **previous** time step is changed. Note that this analysis differs from the analysis in Section 5.3 where that analysis focused on how rangers' predicted detections would change as a result of changing the effort allocation in the current time step (e.g.., patrol more in this area to increase detections). This analysis focuses on how rangers' predicted detections would change had their previous time step's effort allocation been different. In other words, how do we predict the attacker would have adapted to the rangers' patrolling strategy?

Results for changes in predictions are shown in Tables 9.4, 9.5, 9.6, and 9.7, and changes in prediction probabilities are shown in Tables 9.8, 9.9, 9.10, and 9.11.

Because the trends are similar for each time scale, let's look at the three-month time scale as an example. In Table 9.5, for each type of change in effort (`low` to `high` or `high` to `low`),

Effort Change	Neg to Pos	Pos to Neg	No Change (Pos)	No Change (Neg)
Low to High	52	28	107	1253
High to Low	6	86	229	761

Table 9.6: Six-month time scale prediction changes as function of previous effort

Effort Change	Neg to Pos	Pos to Neg	No Change (Pos)	No Change (Neg)
Low to High	41	22	118	1304
High to Low	4	64	246	723

Table 9.7: Annual time scale prediction changes as function of previous effort

Effort Change	Inc	Mean Inc	Dec	Mean Dec	No Change(Pos)	No Change(Neg)
Low to High	1298	0.18	157	0.11	0	5
High to Low	271	0.07	759	0.19	0	32

Table 9.8: One-month time scale prediction probability changes as function of previous effort

there are three possible outcomes for a prediction change: a negative prediction (no detection) can change to a positive prediction (detected attack), referred to as **Neg to Pos**, positive can change to negative (**Pos to Neg**), and there can be no change in the prediction (for either the positive or negative prediction cases). Given these outcomes, we make the following observations. First, there are a substantial number of cells whose corresponding detection predictions do not change as a result of changes in effort. In the case of the unchanged positive predictions, these are predicted to be high-risk cells where rangers will find poaching activity regardless of their past patrolling efforts. For unchanged negative predictions, these correspond to low-risk cells that are essentially predicted to not be attacked at all. Second, there is a seemingly paradoxical effect whereby rangers' increased patrol efforts in the past correspond to a predicted increase in positive detections in the future; by patrolling an area more, it is predicted that rangers have an inverse deterring effect on poaching and poachers are attacking more as a result. The trend is reversed for the case when past patrolling is decreased. These trends indicate that there is more analysis to be done in regards to assessing the deterrent effect of patrolling on poaching, and we'll examine one potential confound at the end of this section: data bias.

Because the trends are similar for each time scale, let's look at the three-month time scale as an example. As for the prediction probability changes in Table 9.9, we examine changes in the

Effort Change	Inc	Mean Inc	Dec	Mean Dec	No Change(Pos)	No Change(Neg)
Low to High	1311	0.20	141	0.12	5	4
High to Low	236	0.06	787	0.18	0	38

Table 9.9: Three-month time scale prediction probability changes as function of previous effort

Effort Change	Inc	Mean Inc	Dec	Mean Dec	No Change(Pos)	No Change(Neg)
Low to High	1244	0.14	183	0.09	2	11
High to Low	410	0.08	643	0.16	0	29

Table 9.10: Six-month time scale prediction probability changes as function of previous effort

prediction probability with increases and decreases referred to as **Inc** and **Dec** respectively, the mean changes in prediction probability for the increase and decrease cases (referred to as **Mean Inc** and **Mean Dec** respectively), and also in the instances where there was no change in the probability for both the positive (i.e., probability ≥ 0.50) and negative (i.e., probability < 0.50) cases. Similar to the trends observed with the prediction changes in Table 5.6, a paradoxical effect is observed: as rangers increase their patrolling efforts, detections of poaching activity increase in the future which correspond to poachers deciding to attack more. Similarly, when rangers decrease their patrolling efforts, it is predicted that detections will decrease in the future, corresponding to a deterrent effect.

9.2.1 Data Bias Confound

One plausible explanation for this trend is data bias: rangers are more likely to find snares in areas they are patrolling more heavily. Detections of attacks are imperfect in this domain; just because

Effort Change	Inc	Mean Inc	Dec	Mean Dec	No Change(Pos)	No Change(Neg)
Low to High	1243	0.12	202	0.10	4	36
High to Low	454	0.07	537	0.15	0	46

Table 9.11: Annual time scale prediction probability changes as function of previous effort

a ranger patrols an area that contains a snare does not mean that the snare will be detected. Additionally, rangers are more likely to detect snares if they patrol a snared area more heavily. Finally, if a ranger does not patrol an area that is attacked, there will be no snare detection (and thus no data point). As such, it is possible for a classifier to infer the relationship that more patrolling leads to more snare activity, and even though it could still have good predictive performance (especially if rangers are correctly patrolling where snaring is occurring), it is difficult to conduct any sort of deterrence analysis because of this data bias challenge.

Bibliography

Abbasi, Y. D., Short, M., Sinha, A., Sintov, N., Zhang, C., & Tambe, M. (2015). Human adversaries in opportunistic crime security games: Evaluating competing bounded rationality models. In *Third Annual Conference on Advances in Cognitive Systems ACS*, p. 2.

Alpern, S., Morton, A., & Papadaki, K. (2011). Patrolling games. *Operations research*, *59*(5), 1246–1257.

Avenhaus, R., von Stengel, B., & Zamir, S. (2002). Inspection games..

Basilico, N., & Gatti, N. (2014). Strategic guard placement for optimal response toalarms in security games. In *International Conference on Autonomous Agents and Multiagent systems*.

Basilico, N., Gatti, N., & Amigoni, F. (2012). Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder. *Artificial Intelligence Journal*, *184–185*, 78–123.

Beck, C., & McCue, C. (2009). Predictive policing: what can we learn from wal-mart and amazon about fighting crime in a recession?. *Police Chief*, *76*(11), 18.

Bell, M. G. H., U., K., D., S. J., & A., F. (2008). Attacker-defender models and road network vulnerability. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1872), 1893–1906.

Bishop, C. M. (2006). Pattern recognition. *Machine Learning*, *128*, 1–58.

Bosansky, B., Jiang, A., Tambe, M., & Kiekintveld, C. (2015). Combining compact representation and incremental generation in large games with sequential strategies. In *AAAI*.

Bošanskỳ, B., Lisỳ, V., Jakob, M., & Pěchouček, M. (2011). Computing time-dependent policies for patrolling games with mobile targets. In *AAMAS*, pp. 989–996.

Brown, M., Haskell, W. B., & Tambe, M. (2014a). Addressing scalability and robustness in security games with multiple boundedly rational adversaries. In *Conference on Decision and Game Theory for Security (GameSec)*.

Brown, M., Saisubramanian, S., Varakantham, P. R., & Tambe, M. (2014b). Streets: game-theoretic traffic patrolling with exploration and exploitation. In *IAAI*.

Camerer, C. (2003). *Behavioral game theory: Experiments in strategic interaction*. Princeton University Press.

Census, G. E. (2016). The great elephant census — a paul g. allen project. Press Release.

Charness, G., Gneezy, U., & Kuhn, M. A. (2012). Experimental methods: Between-subject and within-subject design. *Journal of Economic Behavior and Organization*, *81*(1), 1–8.

Correa, J. R., Harks, T., Kreuzen, V. J. C., & Matuschke, J. (2014). Fare evasion in transit networks..

Costa-Gomes, M., Crawford, V. P., & Broseta, B. (2001). Cognition and behavior in normal-form games: An experimental study. *Econometrica*, *69*(5).

Critchlow, R., Plumptre, A., Andira, B., Nsubuga, M., Driciru, M., Rwetsiba, A., Wanyama, F., & Beale, C. (2016). Improving law enforcement effectiveness and efficiency in protected areas using ranger-collected monitoring data..

Critchlow, R., Plumptre, A., Driciru, M., Rwetsiba, A., Stokes, E., Tumwesigye, C., Wanyama, F., & Beale, C. (2015). Spatiotemporal trends of illegal activities from ranger-collected data in a ugandan national park. *Conservation Biology*, *29*(5), 1458–1470.

Cui, J., & John, R. S. (2014). Empirical comparisons of descriptive multi-objective adversary models in stackelberg security games. In *Decision and Game Theory for Security*, pp. 309–318. Springer.

Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML.

Delle Fave, F. M., Jiang, A. X., Yin, Z., Zhang, C., Tambe, M., Kraus, S., & Sullivan, J. P. (2014). Game-theoretic patrolling with dynamic execution uncertainty and a case study on a real transit system. *Journal of Artificial Intelligence Research*, *50*, 321–367.

Dong, X., Li, C., Li, J., Wang, J., & Huang, W. (2010). A game-theoretic analysis of implementation of cleaner production policies in the chinese electroplating industry. *Resources, Conservation and Recycling*, *54*(12), 1442–1448.

Eck, J., Chainey, S., Cameron, J., & Wilson, R. (2005). Mapping crime: Understanding hotspots..

Eppstein, D., & Goodrich, M. T. (2008). Studying (non-planar) road networks through an algorithmic lens. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, p. 16. ACM.

Fang, F., Nguyen, T. H., Pickles, R., Lam, W. Y., Clements, G. R., An, B., Singh, A., Tambe, M., & Lemieux, A. (2016). Deploying paws: Field optimization of the protection assistant for wildlife security. In *Innovative Applications of Artificial Intelligence Conference*.

Fang, F., Stone, P., & Tambe, M. (2015). When security games go green: Designing defender strategies to prevent poaching and illegal fishing. In *International Joint Conference on Artificial Intelligence*.

Fave, F. M. D., Jiang, A. X., Yin, Z., Zhang, C., Tambe, M., Kraus, S., & Sullivan, J. (2014). Game-theoretic security patrolling with dynamic execution uncertainty and a case study on a real transit system..

Filar, J., et al. (1985). Player aggregation in the traveling inspector model. *IEEE Transactions on Automatic Control*, *30*(8), 723–729.

Gutfraind, A., Hagberg, A., & Pan, F. (2009). *Optimal interdiction of unreactive Markovian evaders*, pp. 102–116. Springer.

Haines, A. M., Elledge, D., Wilsing, L. K., Grabe, M., Barske, M. D., Burke, N., & Webb, S. L. (2012). Spatially explicit analysis of poaching activity as a conservation management tool. *Wildlife Society Bulletin*, *36*(4), 685–692.

Haskell, W., Kar, D., Fang, F., Tambe, M., Cheung, S., & Denicola, E. (2014). Robust protection of fisheries with compass. In *Innovative Applications of Artificial Intelligence (IAAI)*.

Institute, B. (2011). Top ten toxic pollution problems: Tannery operations. Report, Blacksmith Institute.

Jain, M., Conitzer, V., & Tambe, M. (2013). Security scheduling for real-world networks. In *AAMAS*.

Jain, M., Korzhyk, D., Vanek, O., Conitzer, V., Pechoucek, M., & Tambe, M. (2011). A double oracle algorithm for zero-sum security games on graphs. In *AAMAS*.

Jainani, D. (2015). Kanpur leather industry in danger as ngt cracks whip on pollution..

Johansson, U., Sönströd, C., Norinder, U., & Boström, H. (2011). Trade-off between accuracy and interpretability for predictive in silico modeling. *Future medicinal chemistry*, *3*(6), 647–663.

Kanevski, M., Pozdnoukhov, A., & Timonin, V. (2008). Machine learning algorithms for geospatial data. applications and software tools. In *4th Biennial Meeting of the International Environmental Modelling and Software Society*, pp. 7–10.

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

Kar, D., Ford, B., Gholami, S., Fang, F., Plumptre, A., Tambe, M., Driciru, M., Wanyama, F., Rwetsiba, A., Nsubaga, M., 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*, pp. 159–167. International Foundation for Autonomous Agents and Multiagent Systems.

Kiekintveld, C., Islam, T., & Kreinovich, V. (2013). Security games with interval uncertainty. In *International Conference on Autonomous Agents and Multiagent systems*.

Koen, H., de Villiers, J. P., Pavlin, G., de Waal, A., de Oude, P., & Mignet, F. (2014). A framework for inferring predictive distributions of rhino poaching events through causal modelling. In *Information Fusion (FUSION), 2014 17th International Conference on*, pp. 1–7. IEEE.

Korzhyk, D., Conitzer, V., & Parr, R. (2011). Solving stackelberg games with uncertain observability. In *International Conference on Autonomous Agents and Multiagent Systems*.

Lee, W. S., & Liu, B. (2003). Learning with positive and unlabeled examples using weighted logistic regression. In *ICML*, Vol. 3.

Leottau, D. L., Ruiz-del Solar, J., MacAlpine, P., & Stone, P. (2015). A study of layered learning strategies applied to individual behaviors in robot soccer. In *Robot Soccer World Cup*, pp. 290–302. Springer International Publishing.

Manadhata, P., & Wing, J. M. (2004). Measuring a system's attack surface. Tech. rep., DTIC Document.

Mc Carthy, S., Tambe, M., Kiekintveld, C., Gore, M. L., & Killion, A. (2016). Preventing illegal logging: Simultaneous optimization of resource teams and tactics for security. In *AAAI*.

McFadden, D. (1973). Conditional logit analysis of qualitative choice behavior..

Morton, D. P., Feng, P., & J., S. K. (2007). Models for nuclear smuggling interdiction. *IIE Transactions*, *39*(1), 3–14.

Munoz de Cote, E., Stranders, R., Basilico, N., Gatti, N., & Jennings, N. (2013). Introducing alarms in adversarial patrolling games. In *International Conference on Autonomous agents and Multiagent systems*.

Mwinyihija, M. (2011). Emerging world leather trends and continental shifts on leather and leathergoods production. In *World leather congress proceedings*.

Nguyen, T. H., Delle Fave, F. M., Kar, D., Lakshminarayanan, A. S., Yadav, A., Tambe, M., Agmon, N., Plumptre, A. J., Driciru, M., Wanyama, F., et al. (2015). Making the most of our regrets: Regret-based solutions to handle payoff uncertainty and elicitation in green security games. In *International Conference on Decision and Game Theory for Security*, pp. 170–191. Springer.

Nguyen, T. H., Sinha, A., Gholami, S., Plumptre, A., Joppa, L., Tambe, M., Driciru, M., Wanyama, F., Rwetsiba, A., Critchlow, R., et al. (2016). Capture: A new predictive anti-poaching tool for wildlife protection. In *International Conference on Autonomous Agents & Multiagent Systems*.

Nguyen, T. H., Yang, R., Azaria, A., Kraus, S., & Tambe, M. (2013). Analyzing the effectiveness of adversary modeling in security games.. In *AAAI*.

O'Kelly, H. J. (2013). Monitoring conservation threats, interventions, and impacts on wildlife in a cambodian tropical forest..

on International Trade in Endangered Species of Wild Fauna, C., & Flora (2016). African elephants still in decline due to high levels of poaching. Press Release.

Palfrey, T. R., & McKelvey, R. (1995). Quantal response equilibria in normal form games. *Games and Economic Behavior (special issue on Experimental Game Theory)*, *10*, 6.

Park, N., Serra, E., Snitch, T., & Subrahmanian, V. (2015a). Ape: A data-driven, behavioral model-based anti-poaching engine. *IEEE Transactions on Computational Social Systems*, *2*(2), 15–37.

Park, N., Serra, E., & Subrahmanian, V. (2015b). Saving rhinos with predictive analytics. *IEEE Intelligent Systems*, *30*(4).

Perry, W. L. (2013). *Predictive policing: The role of crime forecasting in law enforcement operations*. Rand Corporation.

Rashidi, P., Wang, T., Skidmore, A., Mehdipoor, H., Darvishzadeh, R., Ngene, S., Vrieling, A., & Toxopeus, A. G. (2016). Elephant poaching risk assessed using spatial and non-spatial bayesian models. *Ecological Modelling*, *338*, 60–68.

Rashidi, P., Wang, T., Skidmore, A., Vrieling, A., Darvishzadeh, R., Toxopeus, B., Ngene, S., & Omondi, P. (2015). Spatial and spatiotemporal clustering methods for detecting elephant poaching hotspots. *Ecological Modelling*, *297*, 180–186.

Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2010). Rusboost: A hybrid approach to alleviating class imbalance. *IEEE SMC-A: Systems and Humans*, *40*(1), 185–197.

Shieh, E., An, B., Yang, R., Tambe, M., Baldwin, C., DiRenzo, J., Maule, B., & Meyer, G. (2012). Protect: A deployed game theoretic system to protect the ports of the united states. In *International Conference on Autonomous Agents and Multiagent Systems*.

Shieh, E., Jiang, A. X., Yadav, A., Varakantham, P., & Tambe, M. (2014). Unleashing dec-mdps in security games: Enabling effective defender teamwork. In *ECAI*.

Solberg, A. H. S., Taxt, T., & Jain, A. K. (1996). A markov random field model for classification of multisource satellite imagery. *IEEE TGRS*, *34*(1), 100–113.

Stahl, D., & Wilson, P. (1994). Experimental evidence on players' models of other players. *Journal of Economic Behavior & Organization*, *25*(3).

Sukthankar, G., Goldman, R., Geib, C., Pynadath, D., & Bui, H. (Eds.). (2014). *Plan, Activity, and Intent Recognition*. Elsevier.

Tambe, M. (2011). *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, New York, NY.

Tapiero, C. S. (2005). Environmental quality control and environmental games. *Environmental Modeling & Assessment*, *9*(4), 201–206.

Tsai, J., Yin, Z., Kwak, J.-y., Kempe, D., Kiekintveld, C., & Tambe, M. (2010). Urban security: Game-theoretic resource allocation in networked physical domains..

Venkatesh, V., & Davis, F. D. (2000). A theoretical extension of the technology acceptance model: Four longitudinal field studies. *Management science*, *46*(2), 186–204.

von Stengel, B. (2014). Recursive inspection games..

Wright, J. R., & Leyton-Brown, K. (2010). Beyond equilibrium: Predicting human behavior in normal-form games.. In *AAAI*.

Wright, J. R., & Leyton-Brown, K. (2012). Behavioral game theoretic models: A bayesian framework for parameter analysis. In *International Conference on Autonomous Agents and Multiagent Systems*.

Wright, J. R., & Leyton-Brown, K. (2014). Level-0 meta-models for predicting human behavior in games. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, EC.

Yang, R., Fang, F., Jiang, A. X., Rajagopal, K., Tambe, M., & Maheswaran, R. (2012). Modeling human bounded rationality to improve defender strategies in network security games. In *HAIDM workshop at AAMAS*.

Yang, R., Ford, B., Tambe, M., & Lemieux, A. (2014). Adaptive resource allocation for wildlife protection against illegal poachers. In *Iinternational conference on Autonomous Agents and Multiagent Systems*.

Yang, R., Kiekintveld, C., Ordonez, F., Tambe, M., & John, R. (2011). Improving resource allocation strategy against human adversaries in security games. In *International Joint Conference on Artificial Intelligence*.

Yang, R., Ordonez, F., & Tambe, M. (2012). Computing optimal strategy against quantal response in security games. In *AAMAS*.

Yin, Z., & Collins, R. (2007). Belief propagation in a 3d spatio-temporal mrf for moving object detection. In *IEEE CVPR*, pp. 1–8. IEEE.

Yin, Z., Jiang, A., Johnson, M., Tambe, M., Kiekintveld, C., Leyton-Brown, K., Sandholm, T., & Sullivan, J. (2012). Trusts: Scheduling randomized patrols for fare inspection in transit systems. In *IAAI*.

Zhang, C., Bucarey, V., Mukhopadhyay, A., Sinha, A., Qian, Y., Vorobeychik, Y., & Tambe, M. (2016). Using abstractions to solve opportunistic crime security games at scale. In *International Conference on Autonomous Agents and Multiagent Systems*.

Zhang, C., Jiang, A. X., Short, M. B., Brantingham, P. J., & Tambe, M. (2014). Defending against opportunistic criminals: New game-theoretic frameworks and algorithms. In *International Conference on Decision and Game Theory for Security*.

Zhang, C., Sinha, A., & Tambe, M. (2015). Keeping pace with criminals: Designing patrol allocation against adaptive opportunistic criminals. In *International Conference on Autonomous Agents and Multiagent systems*.

Zhang, Y., Brady, M., & Smith, S. (2001). Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE transactions on medical imaging*, *20*(1), 45–57.