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

Benjamin Ford(✉), Thanh Nguyen, Milind Tambe, Nicole Sintov, and Francesco Delle Fave

University of Southern California, Los Angeles, CA, USA
{benjamif,thanhhng,tambe,sintov,dellefav}@usc.edu

Abstract. Interdicting the flow of illegal goods (such as drugs and ivory) is a major security concern for many countries. The massive scale of these networks, however, forces defenders to make judicious use of their limited resources. While existing solutions model this problem as a Network Security Game (NSG), they do not consider humans' bounded rationality. Previous human behavior modeling works in Security Games, however, make use of large training datasets that are unrealistic in real-world situations; the ability to effectively test many models is constrained by the time-consuming and complex nature of field deployments. In addition, there is an implicit assumption in these works that a model's prediction accuracy strongly correlates with the performance of its corresponding defender strategy (referred to as predictive reliability). If the assumption of predictive reliability does not hold, then this could lead to substantial losses for the defender. In the following paper, we (1) first demonstrate that predictive reliability is indeed strong for previous Stackelberg Security Game experiments. We also run our own set of human subject experiments in such a way that models are restricted to learning on dataset sizes representative of real-world constraints. In the analysis on that data, we demonstrate that (2) predictive reliability is extremely weak for NSGs. Following that discovery, however, we identify (3) key factors that influence predictive reliability results: the training set's exposed attack surface and graph structure.

1 Introduction

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 [16]. 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 [15], 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. [10] for perfectly rational attackers.

MHR Khouzani et al. (Eds.): GameSec 2015, LNCS 9406, pp. 35–56, 2015.
DOI: 10.1007/978-3-319-25594-1_3

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 [1,5,11,14]. 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 [7,15], 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 [7,15] 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 [1,5] 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

[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.

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 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.

2 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 1. In NSGs, there is a network (shown in Fig. 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

Table 1. Notations used in this paper

$g(V, E)$	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

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.

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 $\sum_i x_i = 1$.

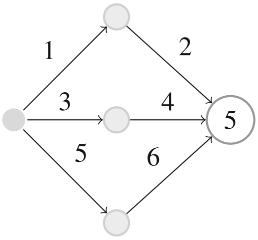

Fig. 1. Example graph

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 Eq. 1 where the term $p_j(x)$ (defined in Eq. 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., $EU_a(x, A_j) = -EU_d(x, A_j)$.

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

In Eq. 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 ($z_{ij} = 0$).

$$p_j(x) = \sum_{X_i \in X} z_{ij} x_i \tag{2}$$

3 Related Work

Human bounded rationality has received considerable attention in Security Game research [1,5,11,14]. 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 [1,11,14]. 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 [11,14]. 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. [14] 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, NSG research covers a wide variety of applications and domains. NSGs have been applied to curbing the illegal smuggling of nuclear material [13], protecting maritime assets such as ports and ferries [15], studying ways to minimize road network disruptions [2], deterring fare evasion in public transit systems [4], and the assignment of checkpoints to urban road networks [9,17]. Although our NSG models most closely resemble the model used by Jain et al. [9,10], 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 [3]. Gutfraind et al. [8] address 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. [18] address 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. [10] 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.

4 Adversary Behavioral Models

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

4.1 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., $A_{opt} = \text{argmax}_{A_j} EU_a(x, A_j)$.

4.2 The Quantal Response Model

The Quantal Response (QR) model for NSGs was first introduced by Yang et al. [18]. 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:

$$q_j(\lambda|x) = \frac{e^{\lambda EU_j^a(x)}}{\sum_{A_k \in A} e^{\lambda EU_k^a(x)}} \tag{3}$$

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.

4.3 The Subjective Utility Quantal Response Model

Unlike QR, the Subjective Utility Quantal Response (SUQR) model [14] 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. [14] for SSGs and Abbasi et al. [1] 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 Eq. 4. Specifically, SUQR predicts the probability that the adversary chooses a path A_j as the following:

$$q_j(\omega|x) = \frac{e^{\omega_1 p_j(x) + \omega_2 \mathcal{T}(t_j)}}{\sum_{A_k \in A} e^{\omega_1 p_k(x) + \omega_2 \mathcal{T}(t_k)}} \tag{4}$$

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 $\mathcal{T}(t_j)$.

4.4 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 Eq. 5. This model will henceforth be referred to as GSUQR1 (i.e., Graph-SUQR w/ 1 parameter). Yang et al. [18] also made use of path length as one of the tested QR heuristics.

$$q_j(\omega|x) = \frac{e^{\omega_1 p_j(x) + \omega_2 \mathcal{T}(t_j) + \omega_3 |A_j|}}{\sum_{A_k \in A} e^{\omega_1 p_k(x) + \omega_2 \mathcal{T}(t_k) + \omega_3 |A_k|}} \tag{5}$$

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 Eq. 6, and this model is henceforth referred to as GSUQR2.

$$q_j(\omega|x) = \frac{e^{\omega_1 p_j(x) + \omega_2 \mathcal{T}(t_j) + \omega_3 |A_j| + \omega_4 MTO_j}}{\sum_{A_k \in A} e^{\omega_1 p_k(x) + \omega_2 \mathcal{T}(t_k) + \omega_3 |A_k| + \omega_4 MTO_k}} \tag{6}$$

5 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. [19]. 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 Eq. 7, where r is the current binary search solution,

[2] The algorithm to generate a Maximin strategy can be found in [10].

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 Eq. 8.[3]

$$r \leq EU_d(x^*) \tag{7}$$

$$EU_d(x) = \frac{\sum_{A_j \in A} e^{\lambda EU_a(x,A_j)} EU_d(x, A_j)}{\sum_{A_j \in A} e^{\lambda EU_a(x,A_j)}} \tag{8}$$

After rewriting Eq. 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 [19]: 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 nonzero 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 [19], 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 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 $L_j^{(1)}(p_j(x))$:

$$\begin{aligned} L_j^1(p_j(x)) &= f_j^{(1)}(0) + \sum_{s=1}^{S} \gamma_{js} p_{js} \\ &= e^{\lambda \mathcal{T}(t_j)} + \sum_{s=1}^{S} \gamma_{js} p_{js} \end{aligned} \tag{9}$$

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

$$L_j^2(p_j(x)) = e^{\lambda \mathcal{T}(t_j)} + \sum_{s=1}^{S} \mu_{js} p_{js} \tag{10}$$

[3] Details on the binary search algorithm can be found in Yang et al.'s original PASAQ formulation [19].

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):

$$\min_{x,b} \sum_{A_j \in A} (e^{\lambda \mathcal{T}(t_j)} + \sum_{s=1}^{S} \gamma_{js} p_{js}) r \tag{11}$$

$$+ \sum_{A_j \in A} \mathcal{T}(t_j)(e^{\lambda \mathcal{T}(t_j)} + \sum_{s=1}^{S} \mu_{js} p_{js}) \tag{12}$$

$$s.t \sum_{X_i \in X} x_i \leq 1 \tag{13}$$

$$p_j(x) = \sum_{s=1}^{S} p_{js} \tag{14}$$

$$p_j(x) = \sum_{X_i \in X} z_{ij} x_i \tag{15}$$

$$b_{js} \frac{1}{S} \leq p_{js}, \forall j, s = 1 \ldots S-1 \tag{16}$$

$$p_{j(s+1)} \leq b_{js}, \forall j, s = 1 \ldots S-1 \tag{17}$$

$$0 \leq p_{js} \leq \frac{1}{S}, \forall j, s = 1 \ldots S \tag{18}$$

$$b_{js} \in \{0,1\}, \forall j, s = 1 \ldots S-1 \tag{19}$$

$$z_{ij} \in \{0,1\}, \forall i, j \tag{20}$$

where b_{js} is an auxiliary integer variable that is equal to 0 only if $p_{js} < \frac{1}{S}$ (Eq. 16). Equation 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 [19], given a small enough binary search threshold ϵ and sufficiently large number of segments S, PASAQ is arbitrarily close to the optimal solution.

6 Human Subject Experiments

6.1 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.[4]

[4] For a more detailed discussion of human subject experiment design considerations, such as steps taken to reduce sources of bias, please see the online appendix at: http://teamcore.usc.edu/people/benjamin/Ford15_GameSecAppendix.pdf.

Each participant was presented with a set of fifteen graphs in which they navigated a path from a source node to a destination node through using a series of intermediate nodes. Participants that successfully attacked a destination (without getting caught on an edge) received the corresponding reward; participants that got caught on an edge received zero points for that round. At the end of the experiment, participants received \$1.50 plus the number of points they received (in cents) during the experiment. To avoid learning effects and other sources of bias, we took the following steps: randomized the order in which graphs were presented to participants, withheld success feedback until the end of the experiment, only allowed participants to participate in the experiment once, and finally, we divided participants into separate subject pools such that each participant only played against a single defender strategy and played on each of the fifteen graphs exactly once. Due to the inevitability of some participants playing randomly (thus confounding any behavioral analysis we may conduct), we included a set of validation rounds such that if participants chose a path that was covered by the defender 100 % of the time, we would drop their data from the analysis.

6.2 Experiment Data Composition

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.

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 [6]. 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.

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 [1,11,14,18], 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. [18], 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.

Experiment Set Composition. As mentioned previously, the experiments are divided into three separate experiment sets. Each combination of coverage strategy × 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 (Eq. 3), SUQR (Eq. 4), GSUQR1 (Eq. 5), and GSUQR2 (Eq. 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 (Eq. 3), SUQR (Eq. 4), and GSUQR1 (Eq. 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.

6.3 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.

Model Prediction Accuracy. In previous empirical analyses [1,5] 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 × graph × 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) (Eq. 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.

$$MAE(g, x, \phi) = \max_{A_j \in A} |q_j - \hat{q_j}| \tag{21}$$

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 1. *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}.

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 2. ***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).*

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. [12], more recently, Kar et al. [11] 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 × 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 Fig. 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 Fig. 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 3. ***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.*

When computing this metric for a dataset $d_{\phi,G} \in D_{\Phi,G}$, we take the sum of EAS scores for each graph × 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 × coverage strategy such that the graph's EAS score is one; all paths to

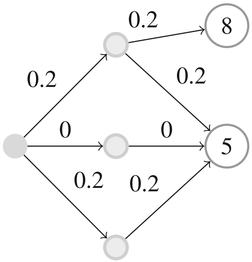

Fig. 2. Example graph 2

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).

7 Predictive Reliability Analysis

After defining predictive reliability in the previous section (Sect. 6.3), we now evaluate predictive reliability in previous work by Nguyen et al. [14] for SSGs, and then follow up with an evaluation of predictive reliability in our work for NSGs.

7.1 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.

7.2 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. [14] (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 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.

Table 2. Guards and treasures predictive reliability

	MAE	AAE
Predictive reliability	91 %	85 %

7.3 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 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 [1,5]) 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).

Table 3. NSG predictive reliability

	MAE	FMAE	FAAE
Predictive reliability	23 %	24 %	22 %

7.4 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 Figs. 3, 4, and 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

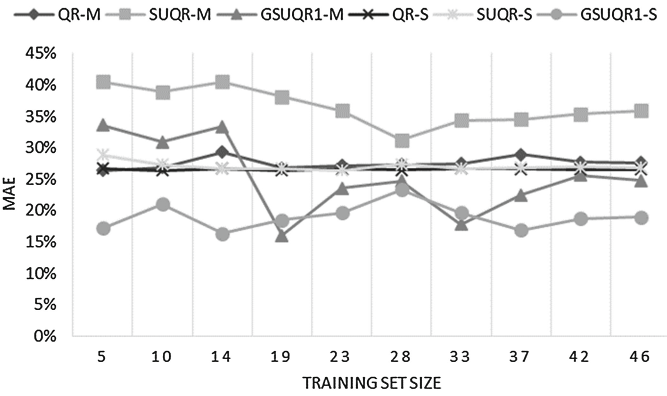

Fig. 3. MAE as a function of training set size (GSUQR2 testing set, graph 7)

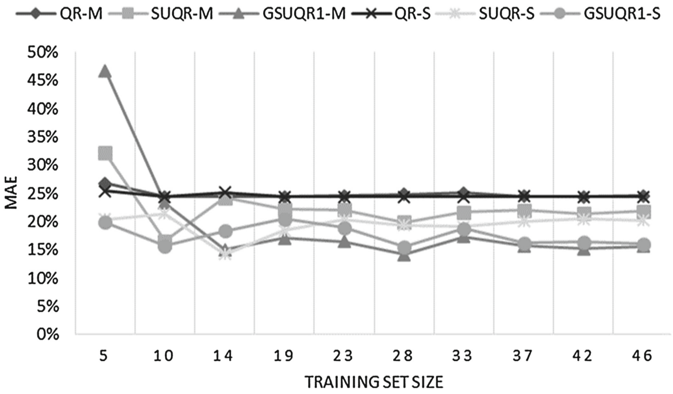

Fig. 4. MAE as a function of training set size (GSUQR2 testing set, graph 9)

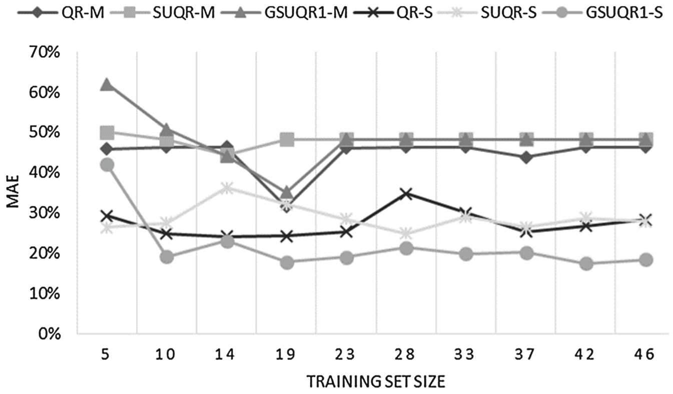

Fig. 5. MAE as a function of training set size (GSUQR2 testing set, graph 11)

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.

8 Predictive Reliability Factors

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

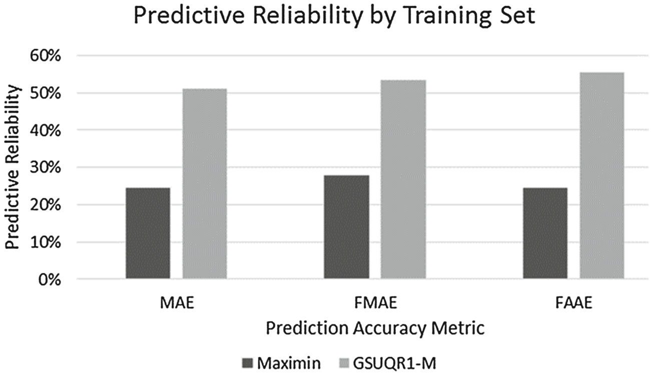

Fig. 6. Predictive reliability as a function of training set and error metric

Training Set Comparison. As discussed in Sect. 6.2, 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 Fig. 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.

Exposed Attack Surface Analysis. Exposed Attack Surface (EAS), as defined in Sect. 6.3, 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 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).

Table 4. Training dataset comparison: sum of exposed attack surfaces

EAS-sum	Maximin	GSUQR1-M
	50	86

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 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.

9 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 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.

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

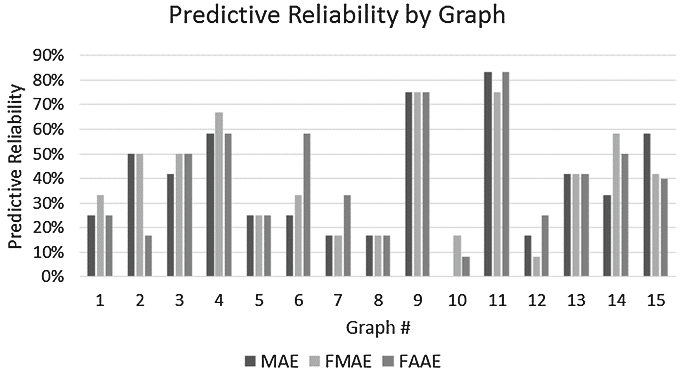

Fig. 7. Predictive reliability as a function of graph

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.

10 Conclusion

Interdicting the flow of illegal goods (such as drugs and ivory) is a major security concern for many countries. However, the massive scale of these networks forces defenders to make judicious use of their limited resources. While existing solutions model this problem as a Network Security Game (NSG), they do not consider humans' bounded rationality. While existing techniques for modeling human behavior make use of large training datasets, this is unrealistic in real-world situations; the ability to effectively test many models is constrained by the time-consuming and complex nature of field deployments. In addition, there is an implicit assumption in these works that a model's prediction accuracy strongly correlates with the performance of its corresponding defender strategy (referred to as predictive reliability). If the assumption of predictive reliability does not hold, then this could lead to substantial losses for the defender. In this paper, we (1) first demonstrated that predictive reliability was strong for previous Stackelberg Security Game experiments. We also ran our 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, we demonstrated that (2) predictive reliability was extremely weak for NSGs. Following that discovery, however, we identified (3) key factors that influenced predictive reliability results: exposed attack surface of the training data and graph structure.

Acknowledgments. This research was supported by MURI Grant W911NF-11-1-0332 and by CREATE under grant number 2010-ST-061-RE0001.

References

1. Abbasi, Y.D., Short, M., Sinha, A., Sintov, N., Zhang, C., Tambe, M.: Human adversaries in opportunistic crime security games: evaluating competing bounded rationality models. In: 3rd Conference on Advances in Cognitive Systems (2015)
2. Bell, M.G.H., Kanturska, U., Schmöcker, J.D., Fonzone, A.: Attacker-defender models and road network vulnerability. Philos. Trans. Roy. Soc. A: Math. Phys. Eng. Sci. **366**(1872), 1893–1906 (2008)
3. Camerer, C.: Behavioral Game Theory: Experiments in Strategic Interaction. Princeton University Press, Princeton (2003)
4. Correa, J.R., Harks, T., Kreuzen, V.J.C., Matuschke, J.: Fare evasion in transit networks. In: CoRR (2014)
5. Cui, J., John, R.S.: Empirical comparisons of descriptive multi-objective adversary models in stackelberg security games. In: Poovendran, R., Saad, W. (eds.) GameSec 2014. LNCS, vol. 8840, pp. 309–318. Springer, Heidelberg (2014)
6. Eppstein, D., Goodrich, M.T.: 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 (2008)
7. Fave, F.M.D., Jiang, A.X., Yin, Z., Zhang, C., Tambe, M., Kraus, S., Sullivan, J.: Game-theoretic security patrolling with dynamic execution uncertainty and a case study on a real transit system. J. Artif. Intell. Res. **50**, 321–367 (2014)
8. Gutfraind, A., Hagberg, A., Pan, F.: Optimal interdiction of unreactive markovian evaders. In: Hooker, J.N., van Hoeve, W.-J. (eds.) CPAIOR 2009. LNCS, vol. 5547, pp. 102–116. Springer, Heidelberg (2009)
9. Jain, M., Conitzer, V., Tambe, M.: Security scheduling for real-world networks. In: AAMAS (2013)
10. Jain, M., Korzhyk, D., Vanek, O., Conitzer, V., Pechoucek, M., Tambe, M.: A double oracle algorithm for zero-sum security games on graphs. In: AAMAS (2011)
11. Kar, D., Fang, F., Fave, F.D., Sintov, N., Tambe, M.: "A game of thrones": when human behavior models compete in repeated stackelberg security games. In: AAMAS (2015)
12. Manadhata, P., Wing, J.M.: Measuring a system's attack surface. Technical report, DTIC Document (2004)
13. Morton, D.P., Pan, F., Saeger, K.J.: Models for nuclear smuggling interdiction. IIE Trans. **39**(1), 3–14 (2007)
14. Nguyen, T.H., Yang, R., Azaria, A., Kraus, S., Tambe, M.: Analyzing the effectiveness of adversary modeling in security games. In: AAAI (2013)

15. Shieh, E., An, B., Yang, R., Tambe, M., Baldwin, C., DiRenzo, J., Maule, B., Meyer, G.: Protect: a deployed game theoretic system to protect the ports of the united states. In: AAMAS (2012)
16. Tambe, M.: Security and Game Theory: Algorithms, Deployed Systems. Lessons Learned. Cambridge University Press, New York (2011)
17. Tsai, J., Yin, Z., Kwak, J.y., Kempe, D., Kiekintveld, C., Tambe, M.: Urban security: Game-theoretic resource allocation in networked physical domains. In: AAAI (2010)
18. Yang, R., Fang, F., Jiang, A.X., Rajagopal, K., Tambe, M., Maheswaran, R.: Modeling human bounded rationality to improve defender strategies in network security games. In: HAIDM Workshop at AAMAS (2012)
19. Yang, R., Ordonez, F., Tambe, M.: Computing optimal strategy against quantal response in security games. In: AAMAS (2012)