Robust Resource Allocation in Security Games and Ensemble Modeling of Adversary Behavior

Arjun Tambe
University of Southern California
941 Bloom Walk,
Los Angeles, CA 90089-0781
arjuntambe@yahoo.com

Thanh Nguyen
University of Southern California
941 Bloom Walk,
Los Angeles, CA 90089-0781
thanhhng@usc.edu

ABSTRACT

Game theoretic algorithms have been used to optimize the allocation of security resources to improve the protection of critical infrastructure against threats when limits on security resources prevent full protection of all targets. Past approaches have assumed adversaries will always behave to maximize their expected utility, failing to address real-world adversaries who are not perfectly rational. Instead, adversaries may be boundedly rational, i.e., they generally act to increase their expected value but do not consistently maximize it. A successful approach to addressing bounded adversary rationality has been a robust approach that does not explicitly model adversary behavior. However, these robust algorithms implicitly rely on an efficiently computable weak model of adversary behavior, which does not necessarily match adversary behavior trends. We therefore propose a new robust algorithm that provides a more refined model of adversary behavior that retains the advantage of efficient computation. We also develop an ensemble method used to tune the algorithm's parameters, and compare this method's accuracy in predicting adversary behavior to previous work. We test these contributions in security games against human subjects to show the advantages of our approach.

Categories and Subject Descriptors

H.4 [**Computing Methodology**]: Game Theory

General Terms

Algorithms, Security

Keywords

Game theory, Robust Optimization, Security, Uncertainty

1. INTRODUCTION

Many security situations can be modeled by Stackelberg games in which one player, the leader, commits to a mixed strategy and adversaries, the followers, respond knowing the leader's strategy [4]. Game theoretic algorithms allow limited resources to be randomly planned and scheduled accounting for the different values associated with attacks on different targets and for the predicted adversary response. This approach has been used to develop many algorithms, including algorithms deployed for many years to allocate security resources for LAX, several major US ports and transit systems, the Federal Air Marshals, and sustainability schemes for preventing environmental crime [13].

April 13 - 17 2015, Salamanca, Spain
Copyright 2015 ACM 978-1-4503-3196-8/15/04…$15.00
http://dx.doi.org/10.1145/2695664.2695686

Many currently deployed algorithms, such as DOBSS [9] and ASPEN [13], generate an expected-utility-maximizing solution assuming that adversaries are perfectly rational, always seeking to maximize their expected reward. However, the assumption of perfect adversary rationality is not ideal, as substantial evidence has suggested the bounded rationality of human adversaries where human adversaries do not consistently make expected utility-maximizing choices [3, 14]. Two general conceptual approaches seek to address bounded adversary rationality. The first is to begin with a detailed model of adversary behavior and build an algorithm that exploits this model, as in [15, 16]. The second is a robust approach that has an implicit model of adversary behavior, as in [5, 10, 11]. While the question of which approach performs better has not been settled, we focus on this second approach because it has a few advantages over the alternate approach: (1) it is more robust to potential inaccuracies within the model of adversary behavior since it uses milder modeling assumptions than the strict modeling assumptions of the other approach; and (2) these algorithms tend to have significantly faster runtimes.

The most effective robust approach to date has been the MATCH algorithm [11] based on robust optimization [1], and research has found MATCH to be highly effective when tested against human adversaries [11]. By coupling the performance of the attacker and defender, it guards against the possibility of large losses to the defender. However, MATCH has some important limitations that this study seeks to address. The model implicit in MATCH is a weak model of adversary behavior that fits our data poorly. Also, while MATCH prevents large *disproportionate* losses to the defender, it leaves open the possibility of large losses to the defender resulting from poor attacker choices, which may be an unacceptable outcome. We propose a new algorithm, RADAR, that attempts to correct these problems by using a refined model of adversary behavior.

We also test RADAR against MATCH in an online game against human adversaries, an approach that has been used in many previous studies to test other algorithms using the same framework [8, 11, 15]. We find that RADAR generally performs better than MATCH.

2. BACKGROUND AND RELATED WORK

Previous game-theoretic approaches, including currently deployed approaches such as DOBSS [9] and ASPEN [13] and other research into game theoretic algorithms for security [2, 6], assume adversaries choose the strategy that maximizes their expected utility. Real-world adversaries often choose sub-optimal strategies, causing these approaches to perform poorly in tests against human adversaries in non-zero sum games [15].

To date, two approaches have been developed to address the bounded rationality of human adversaries. One is to begin with a model of adversary behavior and craft a response to that model. Algorithms using this approach are less robust, since the model

may not accurately predict how adversaries will actually behave. Moreover, since the models these algorithms are based on are often nonlinear, they lead to optimization problems for the defender that are nonlinear and non-convex, which are difficult to solve and have very large runtimes. For example, this is the case with one of the most successful model-based approaches based on the Quantal Response model [16].

MATCH [11], an algorithm based on robust optimization, was developed to address these deficits. Its runtime is shown to be significantly smaller than model-based approaches [11]. MATCH's general approach is to maximize the defender's expected utility with a constraint bounding the loss of the defender with regard to the loss of the attacker when the attacker deviates from the optimal action. We present the Mixed Integer Linear Program (MILP) for MATCH below, but first, we define our problem space, using the same notation as [11].

The defending force has K resources to assign to a mixed strategy for protecting a set of targets $t_1, t_2, \ldots t_n \in T$. Each target t_i has a reward R_i^a for the attacker and P_i^d penalty for the defender if the target is attacked when unprotected, and has a penalty P_i^a for the attacker and reward R_i^d for the defender if the target is attacked when protected. The defender's strategy x is set of probabilities, where x_i is the probability that target t_i is protected at a given time. The attacker's strategy is given by $q \in \{1, \ldots, n\}$, and represents the single target the attack chooses to attack. $U^d(i,x)$ represents the expected utility for the defender of an attack on target i given strategy x, which can be calculated by $x_i R_i^d + (1 - x_i)P_i^d$. Likewise, $U^a(i,x)$ represents the expected utility for the attacker of an attack on target i, calculated by $(1 - x_i)R_i^a + x_i P_i^a$.

MATCH can be represented as the following MILP [11]:

$$\max \quad \gamma$$

$$\text{s.t.} \quad \sum_{i \in T} x_i = K \tag{1}$$

$$0 \le x_i \le 1 \tag{2}$$

$$q = \arg\max_{\hat{q} \in \{1, \ldots, n\}} U^a(\hat{q}, x) \tag{3}$$

$$\gamma \le U^d(q,x) \tag{4}$$

$$\beta \cdot \left(U^a(q,x) - U^a(\hat{q},x)\right) \ge \gamma - U^d(\hat{q},x) \; \forall \, x_{\hat{q}} \tag{5}$$

MATCH attempts to maximize the defender's expected utility, given by γ. Constraints (1) and (2) ensure that the defender uses all her resources and that the probability each target is protected is between zero and one. Constraint (3) sets q as the target that maximizes the attacker's expected utility. Constraint (4) requires the defender to maximize her expected utility of the attacker's optimal target choice. Finally, the key constraint (5) bounds the loss of the defender with respect to the loss of the attacker. The left side calculates the loss in expected utility resulting from the attacker deviating from the optimal target. The right side calculates the loss in expected utility for the defender resulting from this deviation. *The loss in expected utility to the defender is constrained to be no more than β times the loss to the attacker.*

MATCH's strength is that it addresses bounded adversary rationality using a more robust approach than relying on an explicit model of adversary behavior. MATCH constrains the loss to the defender when the attacker deviates from the optimal by adding a lower bound on the expected utility for the defender of sub-optimal targets. Therefore, even when adversaries select sub-optimal targets, the defender will not suffer a very large loss in expected utility.

MATCH implicitly assumes a model of adversary behavior in which the frequency with which a target is attacked has a linear relationship with the expected utility of the target. A large deviation by the attacker from the optimal choice will also lead to large losses to the defender, which MATCH accepts because it assumes larger deviations are less likely; it performs well if the size of the deviation relates to the frequency of that deviation. Since the loss to the defender correlates in a linear fashion with the size of the deviation by the attacker, MATCH implicitly assumes the likelihood of a deviation should correlate in a linear fashion with the size of that deviation.

3. RADAR ALGORITHM

3.1 Key Insights into RADAR

This study proposes RADAR (Risk-Averse Defense against bounded Adversary Rationality), an algorithm based on a modification of MATCH. MATCH has a constant ratio of expected utility sacrificed by the defender to expected utility sacrificed by the attacker, whereas in RADAR, this ratio decreases with greater attacker sacrifices. This results in a risk-averse strategy since it limits the risk that large attacker sacrifices result in large defender losses.

One potential problem with MATCH is that as adversaries select increasingly sub-optimal targets, the defender will receive increasingly sub-optimal outcomes. While MATCH prevents large *disproportionate* losses resulting from small deviations by the attacker from their optimal choice, it still allows for the possibility of a poor attacker choice resulting in a large loss to the defender. Such large losses may be an unacceptable outcome. As security agencies are typically risk-averse [12], larger losses may be seen as disproportionately worth avoiding. An algorithm that generates strategies consistent with this preference for avoiding large risks, such as RADAR, may thus be desirable.

Second, the assumption of a linear relationship between a target's expected utility and its likelihood of being attacked does not fit our experimental data, where we test defender strategies in security games against human subject. In Figure 1, each data point represents one target in one game. The x-axis represents expected utility, and the y-axis represents frequency, or the number of times a target was selected for attack. Since expected utilities differ across different games, we normalize each expected utility between -1 to 1 by dividing the expected utility of a target by the highest expected utility in that game. The frequency of attack is expressed as a percentage by dividing the number of users selecting it for attack by the total number of users, i.e., we normalize each point between 0 to 1. We gathered 200 data points from 82 human subjects. Our experimental setup is explained in further detail in section 5.

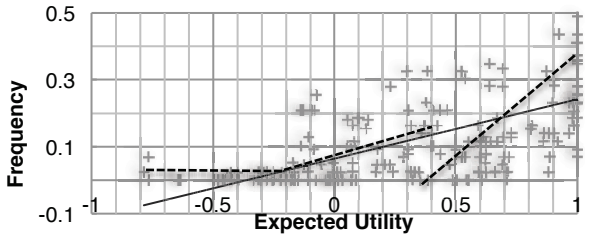

Figure 1. Best fit lines for attack data

The solid line in Figure 1 represents the single best fit line. MATCH assumes that because there is a consistent linear relationship between expected utility and frequency, one line should best fit this graph. However, we find that using three best fit lines for each third of the graph, shown as the dotted lines in the figure, better fits the data. These three lines have significantly different slopes: the first line has a slope of -0.015, the second has a slope of 0.193, and the third has a slope of 0.660. Using three best fit lines has a stronger correlation with the data, with a correlation coefficient of 0.6683 compared to the single line's coefficient of 0.6188. This finding is statistically significant ($p < 0.05$).

3.2 MILP

Based on this analysis, we introduce RADAR. Constraints (5b) through (7) further limit the potential for large losses resulting from sub-optimal adversary choices, resulting in a risk-averse strategy. The MILP for RADAR is shown below:

$$\max_{x,\gamma,q} \quad \gamma$$

s.t.

$$\sum_{i \in T} x_i = K \tag{1}$$

$$0 \le x_i \le 1 \tag{2}$$

$$q = \arg\max_{\hat{q} \in \{1,\dots,n\}} U^a(\hat{q}, x) \tag{3}$$

$$\gamma \le U^d(q, x) \tag{4}$$

$$\gamma - U^d(\hat{q}, x) \le \beta_1 \cdot \left(U^a(q,x) - U^a(\hat{q},x)\right) + m_1 \; \forall \hat{q} \tag{5a}$$

$$\gamma - U^d(\hat{q}, x) \le \beta_2 \cdot \left(U^a(q,x) - U^a(\hat{q},x)\right) + m_2 \; \forall \hat{q} \tag{5b}$$

…

$$\gamma - U^d(\hat{q}, x) \le \beta_k \cdot \left(U^a(q,x) - U^a(\hat{q},x)\right) + m_k \; \forall \hat{q} \tag{5c}$$

$$\beta_k < \beta_{k-1} \; \forall \, k \tag{6}$$

$$m_k > m_{k-1} \; \forall \, k \tag{7}$$

Constraints (1) through (4) serve the same functions in RADAR as they do in MATCH. Constraint (5) constrains the defender's losses in the same way MATCH does, with β controlling the ratio of the defender's loss to the attacker's loss. However, since each additional constraint within line (5) has a greater value of m_k and a smaller value of β_k than the previous constraint, the losses to the defender are constrained to decreasing values as the attacker deviates further from his optimal strategy.

The constraints in (5) can be displayed in the form of a graph. In Figure 2, the x-axis represents the potential difference in expected value for the attacker between the attacker's optimal target choice and chosen target, and the y-axis represents the same difference in expected value for the defender. The slopes and intercepts in the inequalities represent a possible set of m_k and β_k values for a version of RADAR. The loss for the defender relative to the loss for the attacker is bounded to the region at the bottom of the graph, the intersection of all the inequalities.

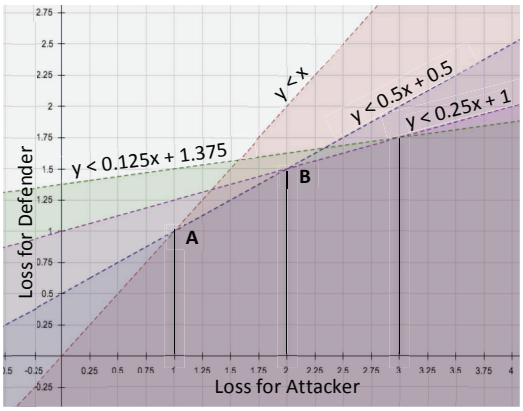

Figure 2. Graph of equations in constraint 5 for RADAR

In this figure, if the attacker loses one unit of expected utility by selecting a sub-optimal target, both MATCH and RADAR will ensure that this selection will be up to one unit away from the optimal for the defender, marked as point A in the graph (if β and $\beta_1 = 1$). If the attacker's choice deviates two units from the optimal, then a defender using MATCH will lose two units of expected utility as well, while RADAR will lose only 1.5 points, marked as point B in the graph.

In all forms of RADAR in this paper, we used four constraints within constraint (5) of RADAR, with four values of β_k and m_k.

4. TUNING AND WEIGHTING ALGORITHMS

In both MATCH and RADAR, the parameters β in MATCH and (β_k, m_k) in RADAR are key pre-determined values controlling the loss of the defender. Therefore, it is important to tune these parameters effectively to obtain a better patrolling strategy for the defender. In this section, we first introduce the adaptation of an iterative program originally used to tune MATCH to RADAR. We then propose an ensemble model of adversary behavior to make more accurate behavioral predictions.

4.1 Tuning Program

We develop an iterative tuning program to tune RADAR. This program is adapted from a program used to tune MATCH [8], and can be represented as below:

Initialize $\gamma^* \leftarrow -Z$ (1)

Iterate through values of $\beta_{1\dots k}, m_{1\dots k}$ (2)

  $x \leftarrow RADAR(\beta_{1\dots k}, m_{1\dots k})$ (3)

  $\gamma \leftarrow \sum_{i \in T} U^d(i, x) \cdot P_{j,i}$ (4)

  If $\gamma \ge \gamma^*$: (5)

    $\gamma^* \leftarrow \gamma$, $\beta^*_{1\dots k} \leftarrow \beta_{1\dots k}, m^*_{1\dots k} \leftarrow m_{1\dots k}$ (6)

Return $\beta^*_{1\dots k}, m^*_{1\dots k}, \gamma^*$ (7)

The program uniformly generates a set of N samples for each parameter $\beta_{1\dots k}$ and $m_{1\dots k}$ within a range, and then iterates through each sampled value for each parameter to find the best combination of parameters. Specifically, Line 1 initializes the value of γ (the final expected utility for the defender) at a large negative number -Z. Line (2) iterates through values of $\beta_{1\dots k}$ and $m_{1\dots k}$ (for example, with a set of nested for loops). Line (3) uses the current parameter values to generate a protection strategy x using RADAR, and line (4) uses this protection strategy to

determine the defender's expected utility for each target, and to generate a prediction of adversary behavior P using a model of adversary behavior j, e.g., SUQR [8], explained in section 5.3. $P_{j,i}$ refers to the predicted likelihood of a target i being attacked, which comes from the model of human behavior. In line (6), the defender's total expected utility is computed given x under the assumption that the attacker responds stochastically according to the human behavior model j. This process is repeated for each parameter value combination, and the output of the program in line (7) is the set of parameter values that generates the highest defender expected utility. Note that for the MATCH and RADAR algorithms, tuning is helpful even if not enough data is available to accurately estimate the parameters of the SUQR model; this is because SUQR is not directly used to predict adversary behavior, but is used as a heuristic to tune the parameters of our robust approach.

4.2 Ensemble Model

The tuning program in section 4.1 requires a model of adversary behavior to generate a prediction of which targets adversaries may choose to attack. The more accurate these models are, the better the tuned algorithms will perform against human adversaries. The following ensemble model of adversary behavior replaces the single model j used in the section above.

The ensemble model is a heuristic to tune RADAR's parameters, but is not used in the algorithm itself. We favor this approach because using the ensemble model mitigates the risk of a single model being flawed by including several models, whereas a single model is not robust to this risk. However, using a model of adversary behavior in the algorithm itself greatly slows the algorithm's runtimes (as shown in the Introduction), which is especially problematic given that the ensemble model uses several models of adversary behavior and is therefore more complex.

The following algorithm allows multiple models to be combined into weighted average prediction (ensemble modeling of adversary behavior), replacing the use of a single model in the original tuning program. The algorithm multiplies the prediction made by each model by a certain value less than one (a "weight"), where the sum of all the weights is one. These products are summed to generate a final prediction. The algorithm, shown below, optimizes these weights such that the final prediction matches as closely as possible the actual behavior of attackers.

In the following algorithm, P_f corresponds to the final prediction, w_j refers to the weight corresponding to a model j, and D refers to the adversary behavior from actual data. The predictions, $\{P_{j,i}\}$, are a set of probabilities denoting the probability each target is attacked, while the data, $\{D_i\}$ are a set of percentages denoting how many times each target is attacked.

$$\min_{w} \quad \sum_{i \in T} G_i$$

$$\text{s.t.} \quad G_i \geq P_{f,i} - D_i \ \forall i \qquad (1)$$

$$G_i \geq -\left(P_{f,i} - D_i\right) \forall i \qquad (2)$$

$$P_{f,i} = \sum_j w_j P_{j,i} \ \forall i \qquad (3)$$

$$\sum_j w_j = 1 \qquad (4)$$

Constraints (1) and (2) replace the use of an absolute value, which cannot be used in a linear program, and define G_i as the 1-norm distance between the final prediction P_f and the actual data D. The algorithm minimizes G by adjusting the values of w_j to bring $P_{f,i}$ closer to D_i. The output is an optimal value for each weight, and an optimal final prediction. The final prediction is used to replace the prediction P_j in the tuning program in section 4.1. This ensemble method is used in the tuning programs for both MATCH and RADAR for the sake of consistency.

We used four models of adversary behavior in the ensemble model: Quantal Response (QR), Subjective Utility Quantal Response (SUQR), a uniform attack distribution (Uniform), and a perfectly rational adversary (Rational Adversary), all of which are explained in section 5.3.

We argue that the ensemble model has two advantages over any singular model. First, it is more robust to inaccuracies of a single model. If one model in the ensemble approach happens to fit the data poorly, the incorporation of additional models will mitigate the impact of the inaccuracy. Second, *the ensemble model can be no worse than any single model.* Even if a single model fits the test set better than any combination of models, the output of the ensemble model would assign a weight of 1 to that single model.

5. EXPERIMENTS

5.1 Format of Experiments

The algorithms were tested in an online game, in the same format as [8, 11, 15]. In all the experiments, the defending force used three guards to protect eight gates. A screenshot of the game interface is shown in Figure 3.

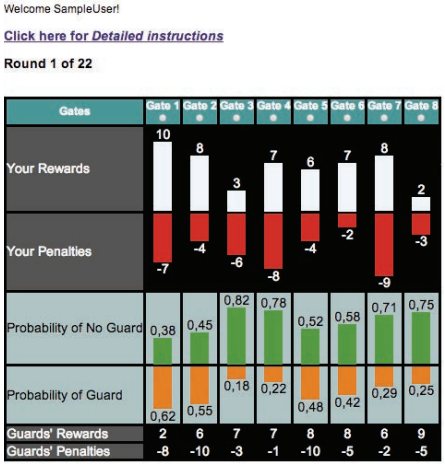

Figure 3. Screenshot of game interface.

As shown in Figure 3, each human subject, playing the role of the attacker in the security game, chooses one gate in each game, with full information about the rewards and penalties associated with an attack on each gate. Subjects also know the probability of each gate being protected, but do not know which gates will be protected at a given time.

Before playing, subjects were given a brief tutorial and a practice round to ensure they understood the game. The experiments were run in Amazon Mechanical Turk with a payment of US $1.50 for playing. Subjects received an additional $.10 for each "point" of reward earned. Two "dummy" games with obvious answers were included (ie, the gate with the highest reward and lowest penalty had the lowest probability of being protected), and if subjects did not select the obvious answers their results were not counted.

Since the weighting algorithm requires some data to learn the weights for the different models, we conducted our experiments in

two batches, a training data set and a test set. In both sets, we used four reward structures from a previous study [15] that found these structures to be representative of a larger group of reward structures. In the training set, we compared versions of RADAR and MATCH whose parameter values were determined without tuning, i.e., by hand. Our training set includes data from 39 subjects who each played 12 games each in the training set, using 3 reward structures[1] with 4 algorithms, DOBSS, MATCH with β =1, and 2 parameter sets of RADAR[2].

The data from the training set was used in the weighting and tuning algorithms to generate a tuned version of RADAR and MATCH. The test set of experiments included the tuned and untuned versions of MATCH and RADAR, and DOBSS. 43 subjects played 20 games each (4 reward structures with 5 algorithms) in the test set. *The training and test sets had an entirely different subject pool.*

In the following sections, we compare the performance of each algorithm against human adversaries, then provide findings regarding the ensemble model.

5.2 Algorithm Results

We present our results in four sets of pairwise comparisons between RADAR and each other algorithm. In each of the following graphs, the x-axis represents an algorithm's performance, measured by the defender's expected utility obtained by different algorithms given the subjects' responses. On the y-axis, each pair of bars represents the results from one game.

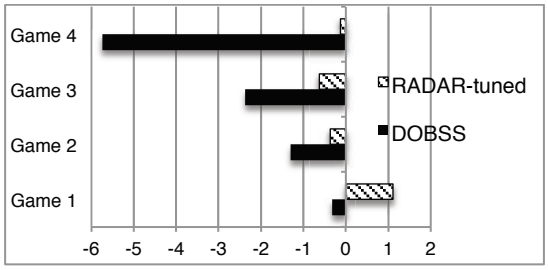

Figure 4. Bar graph comparing performance of RADAR-tuned vs DOBSS

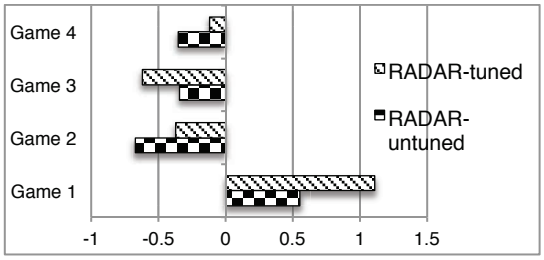

Figure 5. RADAR-tuned vs RADAR-untuned

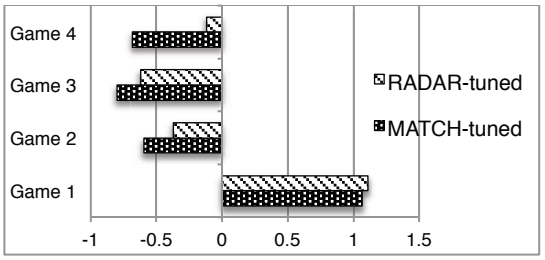

Figure 6. RADAR-tuned vs MATCH-tuned

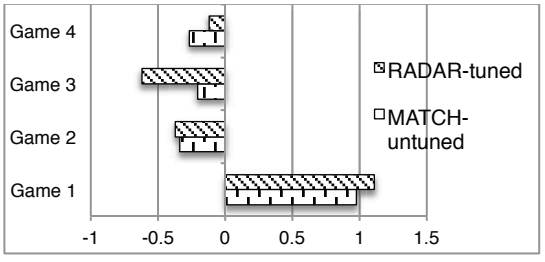

Figure 7. RADAR-tuned vs MATCH-untuned

We note several key findings. First, as shown in Figure 4, RADAR-tuned outperforms DOBSS by a very large margin in all four games. Since DOBSS' approach represents a common assumption of adversary rationality also found in some recent algorithms, RADAR's better performance over DOBSS is an important finding. Second, as shown in Figure 5, tuning RADAR causes it to perform better in 3 of 4 games. This shows that the process of tuning RADAR is an important step to improve its performance. Third, in Figure 6, RADAR-tuned outperforms MATCH-tuned in all four games.

Figure 7, where RADAR-tuned is compared to MATCH-untuned, is the only instance where RADAR-tuned does not perform the best in at least three games. Here, RADAR-tuned performs better in games 1 and 4, and there is no statistically significant difference in game 2. While MATCH performs better in game 3, the fact that tuning both algorithms made them perform worse in this game casts doubt on this finding. Since the tuning process was calibrated using past data, subjects in the training set may have played game 3 different than subjects in the test set, skewing the results. Interestingly, MATCH suffers more from tuning in this game than RADAR does, i.e. the difference between MATCH-tuned and MATCH-untuned is larger than the difference between RADAR-tuned and RADAR-untuned, which provides evidence of RADAR's robustness. Based on these results, we can conclude there are some reward structures for which RADAR is superior to both the tuned and untuned versions of MATCH.

5.3 Weighting Algorithm Results

In this section, we explain why the tuned version of RADAR performed better than the untuned version: the ensemble model used to tune it predicted adversary behavior well.

We use four models of adversary behavior: Quantal Response (QR), Subjective Utility Quantal Response (SUQR), a uniform distribution (Uniform), and a perfectly rational adversary who attacks only the optimal target (Rational Adversary). QR predicts that targets with a higher expected utility are more likely to be attacked. SUQR is based on QR, but replaces the expected utility function with a heuristic it assumes humans use in place of calculating expected utilities. For a detailed discussion of QR, see [7], and for a detailed discussion of SUQR, see [8]. Using the data

[1] We omitted the fourth reward structure in the training set since the strategies generated by each untuned algorithm were very similar. In the test set, the tuned algorithms generated different strategies.

[2] Both versions set β_1, β_2, β_3, and β_4 to 1, 0.5, 0.25, and 0.125. In one version, m_1, m_2, m_3, and m_4 are 0, 0.5, 1, and 3.75; and in the other, m_1, m_2, m_3, and m_4 are 0, 0.875, 1.75, and 2.40625.

from the training set of experiments, the weighting program generated a set of weights used in the ensemble model, which generated predictions of adversary behavior. These predictions are compared to the predictions made by each model individually by their fit with the actual data.

In Figure 8, predictions made by the ensemble model were compared to the predictions of other models in two ways: first, by how well they fit the training set, and second, by how well they fit the test set. On the y-axis, the models' accuracies are expressed as the average 1-norm distance between each model's prediction and the actual data, then normalized between 0 and 1 by dividing it by the highest possible 1-norm distance.

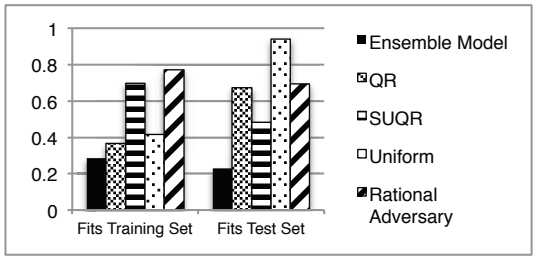

Figure 8. Comparison Between Accuracy of the Ensemble Model and Other Models

In the figure, the ensemble model's predictions match the adversary's behavior more closely than any other model with statistical significance. We additionally note that the ensemble model is more robust when the test data varies from the training data. In Figure 8, QR fits the training set best, whereas SUQR fits the test set best. Selecting QR as the single model based on its performance in the training set would have failed to fit the test set accurately, whereas the ensemble model fits both data sets well. These findings, in addition to the ensemble method's two general advantages shown in section 4.2, are a strong indication that the ensemble model is an effective method to predict human behavior.

6. CONCLUSIONS

Game theoretic algorithms have been used to improve the protection of critical infrastructure against threats, but many algorithms assume perfect adversary rationality. A leading robust approach to addressing imperfect human decision-making, MATCH, has been successful in experimental evaluation. Unfortunately, MATCH implicitly relies on a weak model of adversary behavior, which should be refined to fit our data more closely. Our new robust algorithm, RADAR, provides a refined model of adversary behavior, and plays a more risk-averse strategy than MATCH, which may be desirable for risk-averse security agencies. We also develop an ensemble method to predict human behavior and use this method to tune the parameters in MATCH and RADAR. We test these contributions in human subjects experiments, and find that RADAR performs as well or better than MATCH in three of four games. RADAR also consistently outperforms an algorithm assuming perfect adversary rationality. We also find that our ensemble modeling approach predicts adversary behavior significantly more accurately than alternative models of adversary behavior, highlighting the advantages of such an approach.

7. ACKNOWLEDGEMENTS

Thanh Nguyen is supported by MURI Grant W911NF-11-1-0332.

8. REFERENCES

[1] Aghassi, M., and Bertsimas, D. Robust game theory. *Math Program, 107* (1-2), 231–273.

[2] Basiloco, N., Gatti, N., and Amigoni, F. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. in *AAMAS*, (2009).

[3] Camerer, C. *Behavioral Game Theory: Experiments in Strategic Interaction.* Princeton University Press, Princeton, 2003.

[4] Conitzer, V. 2012. Computing Game-Theoretic Solutions and Applications to Security. in *Twenty-Sixth AAAI Conference*, (2012), Association for the Advancement of Artificial Intelligence.

[5] Jiang, A., Nguyen, T., Tambe, M., Procaccia, A. Monotonic Maximin: A Robust Stackelberg Solution Against Boundedly Rational Followers. In *GameSec 2013.*

[6] Letchford, J., and Vorobeychik, Y. Computing randomized security strategies in networked domains. in *AARM Workshop* in AAAI (2011).

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

[8] Nguyen, T., Yang, R., Azaria, A., Kraus, S., and Tambe, M. Analyzing the Effectiveness of Adversary Modeling in Security Games. in *AAAI 2013*.

[9] Paruchuri, P., Pearce, J., Marecki, J., Tambe, M., Ordonez, F., and Kraus, S. Playing Games for Security: An Efficient Exact Algorithm for Solving Bayesian Stackelberg Games. in *AAMAS 2008*, 895-902.

[10] Pita, J., Jain, M., Tambe, M., Ordonez, F., and Kraus, S. Robust solutions to stackelberg games: Addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence Journal, 174* (15), 1142–1171.

[11] Pita, J., John, R., Maheswaran, R., Tambe, M., and Kraus, S. A Robust Approach to Addressing Human Adversaries in Security Games. in *20th European Conference on Artificial Intelligence*, (2012).

[12] Stewart, M., and Mueller, J. Aviation Security, Risk Assessment, and Risk Aversion for Public Decisionmaking. *Journal of Policy Analysis and Management, 32* (3). 615-633.

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

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

[15] Yang, R., Kiekintveld, C., and Ordonez, F. Improving Resource Allocation Strategies Against Human Adversaries in Security Games: An Extended Study. *Artificial Intelligence*, 195. 440-469.

[16] Yang, R., Ordonez, F., and Tambe, M. Computing Optimal Strategy against Quantal Response in Security Games. in *AAMAS* (2012).