Learning about Cyber Deception through Simulations: Predictions of Human Decision Making with Deceptive Signals in Stackelberg Security Games

Edward A. Cranford (cranford@cmu.edu) and Christian Lebiere (cl@cmu.edu)
Department of Psychology, Carnegie Mellon University, 5000 Forbes Avenue
Pittsburgh, PA 15213 USA

Cleotilde Gonzalez (coty@cmu.edu)
Dynamic Decision Making Laboratory, Social and Decision Sciences Department
Carnegie Mellon University, 5000 Forbes Avenue
Pittsburgh, PA 15213 USA

Sarah Cooney (cooneys@usc.edu), Phebe Vayanos (phebe.vayanos@usc.edu), and Milind Tambe (tambe@usc.edu)
USC Center for AI in Society, University of Southern California, 941 Bloom Walk
Los Angeles, CA 90089 USA

Abstract

To improve cyber defense, researchers have developed algorithms to allocate limited defense resources optimally. Through signaling theory, we have learned that it is possible to trick the human mind when using deceptive signals. The present work is an initial step towards developing a psychological theory of cyber deception. We use simulations to investigate how humans might make decisions under various conditions of deceptive signals in cyber-attack scenarios. We created an Instance-Based Learning (IBL) model of the attacker decisions using the ACT-R cognitive architecture. We ran simulations against the optimal deceptive signaling algorithm and against four alternative deceptive signal schemes. Our results show that the optimal deceptive algorithm is more effective at reducing the probability of attack and protecting assets compared to other signaling conditions, but it is not perfect. These results shed some light on the expected effectiveness of deceptive signals for defense. The implications of these findings are discussed.

Keywords: cyber deception; cognitive models; instance-based learning; stackelberg security games

Introduction

Fooling the human eye is a skill that magicians learn to do very effectively, and one that cognitive scientists investigate to try to reveal answers about essential aspects of human cognition (Ekroll & Wagemans, 2016). Within the domain of cybersecurity, cognitive scientists try to understand the power of deception: a form of persuasion where one intentionally misleads an agent into a false belief, in order to gain an advantage over the agent and to achieve one's goals (Rowe & Rushi, 2016).

While there is much work in the psychology of deception, most of what is currently known consists of the use of verbal and nonverbal cues, including appearance, gestures, and descriptions, and the role of these attributes of social interaction (Morgan, LeSage & Kosslyn, 2009; Bond & DePaulo, 2008; Riggio & Friedman, 1983). For example, most studies frame the study of deception as it relates to the body, face, and the cues that may be leaked through gestures and words (Riggio & Friedman, 1983). In other words, most of what we know about the psychology of deception relies on the physical observation of behavior.

Increased mobile technology and connectedness among people and communities have moved our societies from the physical to the cyberworld: an adversarial setting where security is of essence. Many cyber attacks occur when attackers take advantage of the power of deception. Cyber attackers intentionally mislead end-users and cyber defenders into believing that an action is "safe" in order to manipulate humans into disclosing information or granting access to security systems. Like magicians, cyber attackers are learning to fool the human mind quite effectively. However, cyber defenders can also use deceptive techniques to try to mitigate attacks and to deter or catch attackers. Understanding how attackers learn to adapt to deceptive techniques is important to developing better defense systems.

In physical security systems, researchers have developed algorithms that plan optimal allocation of limited defense resources (Tambe, 2011; Pita et al., 2008; Shieh et al., 2012). The challenge of security resource optimization carries over to cybersecurity where it is important to assist human administrators in defending networks from attacks (Gonzalez et al., 2014). Researchers have relied on insights from Stackelberg Security Games (SSGs) to develop strategies to optimally allocate defense resources (Tambe, 2011). SSGs model the interaction between a defender and an adversary as a leader-follower game (Tambe, 2011). A defender plays a particular defense strategy (e.g., randomized patrolling of airport terminals) and then the adversary takes action after observing the defender's strategy. Recently, this research leveraged signaling theory, where algorithms aim at finding how to "trick" an adversary through the use of deceptive signals in a two-stage SSG (Xu et al., 2015). While proven to work well against perfectly rational adversaries, this research is still in its infancy when it comes to understanding how a human would interpret and react to deceptive signals.

Our research investigates how humans might learn to react to deceptive signaling algorithms. Whereas there is previous work in modeling human decision making in a single-stage SSG, it has not addressed deceptive signaling (Abbasi et al., 2016). Our methods involve pairing deception strategies, including a known and highly regarded defense algorithm (Xu et al., 2015), against a high-fidelity cognitive model known for its accurate representation of human decisions from experience. We created an Instance-Based Learning (IBL) model of the attacker decisions (Gonzalez, Lerch & Lebiere, 2003) using the ACT-R cognitive architecture (Anderson & Lebiere, 1998). The security algorithms and ACT-R model interacted in a newly created Insider Attack Game over multiple rounds with multiple trials in each round. We studied the adaption and learning of the IBL model against the various defense algorithms in order to predict the effectiveness of the deceptive signals against boundedly-rational human agents.

Our simulation results show that the probability of attack increases when there is no warning and decreases under the optimal deceptive signaling algorithm, almost reaching the optimal probability of attack predicted by the algorithm. Interestingly, the cognitive algorithm learns to attack less often when the signal is sometimes deceptive than when the signal is always truthful. The cumulative score per round also demonstrates that the optimal deceptive signaling algorithm is far better at protecting the assets than truthful warnings.

In what follows, we first describe the Stackelberg Security Game (SSG) in the context of the Insider Attack scenario used in our simulations, and introduce the optimal deceptive signaling algorithm. We next describe the IBL cognitive model and the implementation of the simulations that pair the model against the optimal deceptive signaling algorithm, and other defense algorithms, in a SSG. Finally, we present our results and discuss their implications.

A SSG in an Insider Attack Scenario

For the present work, a SSG was developed under an Insider Attack scenario in order to situate the cover story within the cybersecurity domain. In this Insider Attack Game, an adversarial agent takes the role of an employee at a company whose goal is to maximize their score by "hacking" computers to steal proprietary information and avoid being caught by the security analysts (defenders) that monitor the computers. Agents are presented with six computer targets, each with a different payoff (reward/penalty) structure. Agents are provided the reward and penalty values associated with each computer. On a given trial, two security analysts, controlled by an algorithm, monitor one computer each.

From the perspective of the defenders, the game is a two-stage security game. As in classic single-stage SSGs, the first stage involves allocating defense resources. For the present security game, the allocation of the security analysts is optimized by computing the Strong Stackelberg Equilibrium (SSE), which provides a probability of monitoring (m-prob) each computer. Agents are provided this information as a percentage of time that each computer is monitored. In the second stage, information about whether the computer selected by the attacker is being monitored is strategically revealed to the attacker (Xu et al., 2015). It is in this second stage where defenders can take advantage of deceptive signaling techniques.

For each trial, agents first select one of the targets to attack. They are then presented with information about whether or not the computer is being monitored. They must then make a decision to either continue or withdraw the attack. If an agent attacks a computer that is monitored, then they lose a penalty between 1 and 10 points, but if the computer is not monitored they gain a reward between 1 and 10 points. If they choose to withdraw the attack, they receive 0 points.

Agents perform four rounds of 25 trials each. Each round consists of a different set of computers each with different payoff structures, which results in a different allocation of defense resources. In the present game, the SSE allocates defenses across a round in such a manner that the expected values of attacking each computer are positive and all the same. In the current version of the game, defenses are allocated at the start of each trial in accordance with the SSE for the round. Table 1 shows the rewards, penalties, and m-prob for each computer in each round.

In the second stage of the Insider Attack Game, the warning signal was manipulated in different ways across different conditions of the game to assess the effect of deception on decision making. The baseline condition of the game is called the *DeceptiveWarning* condition, which presents deceptive warning signals according to the optimal deceptive signaling algorithm. The key mechanism in this condition is that the warning presented in the second stage of the game is sometimes a lie. If a computer is being monitored, then the attacking agent will always be given a truthful warning signal that the computer is currently monitored. If a computer is not being monitored, then a certain percentage of the time (which is unknown to the agents) a deceptive warning signal is presented claiming that the computer is monitored.

Table 1: Payoff structure for each computer per round.

Round	Target 1	Target 2	Target 3	Target 4	Target 5	Target 6
Round 1	[2, -1, 0.22]	[8, -5, 0.51]	[9, -9, 0.42]	[9, -10, 0.40]	[2, -6, 0.08]	[5, -5, 0.36]
Round 2	[5, -3, 0.41]	[8, -5, 0.48]	[7, -6, 0.41]	[8, -9, 0.37]	[5, -7, 0.27]	[2, -4, 0.05]
Round 3	[3, -3, 0.30]	[9, -4, 0.60]	[6, -6, 0.40]	[5, -8, 0.29]	[3, -6, 0.20]	[2, -2, 0.20]
Round 4	[4, -3, 0.37]	[6, -3, 0.51]	[7, -7, 0.40]	[5, -10, 0.24]	[5, -9, 0.26]	[3, -4, 0.23]

Note. The first number in brackets is the reward, the second number is the penalty, and the third is the probability that the computer is being monitored on any given trial, [payment, penalty, m-prob].

The percent of time that a deceptive warning signal is presented is determined by computing a conditional probability that brings the expected value of attacking the computer to zero, as defined by the following equation:

$$x = \frac{(penalty \times m\text{-}prob)/reward}{1 - m\text{-}prob}$$

The baseline *DeceptiveWarning* condition was compared to four other signaling conditions:

TruthfulWarning – a warning signal is always presented when an analyst is present, never when an analyst is absent; only truthful signals are presented, no deception is used.

NoWarning – a warning signal is never presented regardless of whether an analyst is present or absent; no deception is used.

AllWarning – a warning signal is presented on every trial regardless of actual coverage, therefore a warning is deceptive every time the computer is not being monitored.

RandomWarning – a warning signal is presented randomly whether an analyst is present or absent, in accordance with the probability of a warning signal being presented in the *DeceptiveWarning* condition.

Model Description

A cognitive model of the attacker was created to make predictions about how agents would perform against the various defense algorithms. The cognitive model is implemented in the ACT-R cognitive architecture (Anderson & Lebiere, 1998; Anderson et al, 2004) and decisions are made following the methodology of instance-based learning (IBL) theory (Gonzalez et al., 2003). A model based on mechanisms of the ACT-R architecture limits free parameters and constrains assumptions of representation and processes. IBL has been used to model decision making processes across a number of tasks with much success, including supply chain management (Gonzalez & Lebiere, 2005), social dilemmas (Gonzalez et al., 2015; Lebiere, Wallach & West, 2000; Juvina et al., 2011), two-person games (Sanner et al., 2000, West & Lebiere, 2001), repeated binary-choice decisions (Gonzalez & Dutt, 2011; Lebiere, Gonzalez, & Martin, 2007), and classical single-stage SSGs (Abbasi et al., 2016).

In IBL, decisions are based on past experiences, or instances, that are similar to the current situation. Typically, experiences are encoded as chunks in declarative memory that contain a description of the context in which each decision is made, the decision itself, and the outcome of that decision. In the current model, the context slots include the probability that a computer is being monitored (m-prob; range 0 to 1.0), the value of the reward (range 0 to 10), the value of the penalty (range 0 to -10), and whether or not a warning signal was provided (present or absent). The decision slot includes the action taken (attack or not attack the computer). The outcome slot includes the feedback representing the actual points received based on the action.

For each decision, the model takes the description of each target and the action to attack as input, and produces a projected outcome of attacking that target by retrieving similar past instances. In ACT-R, the retrieval of past instances is based on the activation strength of the relevant chunk in memory and its similarity to the current context. The activation A_i of a chunk i is determined by this equation:

$$A_i = \ln \sum_{j=1}^{n} t_j^{-d} + MP * \sum_{k} Sim(v_k, c_k)_i + \varepsilon_i$$

The first term provides the power law of practice and forgetting, where t_j is the time since the jth occurrence of chunk i and d is the decay rate of each occurrence which is set to the default ACT-R value of 0.5. The second term reflects a partial matching process, where $Sim(v_k,c_k)$ is the similarity between the actual memory value and the corresponding context element for chunk slot k. The term ε_i represents transient noise, a random value from a logistic distribution with a mean of zero and variance parameter s of 0.25 (ACT-R default), to introduce stochasticity in retrieval.

The activation of a particular chunk determines the probability of retrieving that chunk according to the softmax equation, also known as the Boltzmann equation, reflecting the ratio of chunk activation A_i and its noise level s:

$$P_i = \frac{e^{A_i/s}}{\sum_j e^{A_j/s}}$$

The model uses ACT-R's blending mechanism (Lebiere, 1999) to retrieve an expected outcome of attacking a target based on past instances, computed by the following equation:

$$V = \text{argmin} \sum_i P_i \times \left(1 - Sim(V, V_i)\right)^2$$

The value V is the one that best satisfies the constraints of all matching chunks i weighted by their probability of retrieval P_i. Satisficing is defined as minimizing the dissimilarity between the consensus value V and the actual answer V_i contained in chunk i.

Figure 1 shows how the model operates on a given trial. The first step is to select a computer to attack (left side of Figure 1). The model loops through each of the six computers and retrieves a projected outcome of attacking the computer through blending, as described above. The computer with the highest projected outcome is selected.

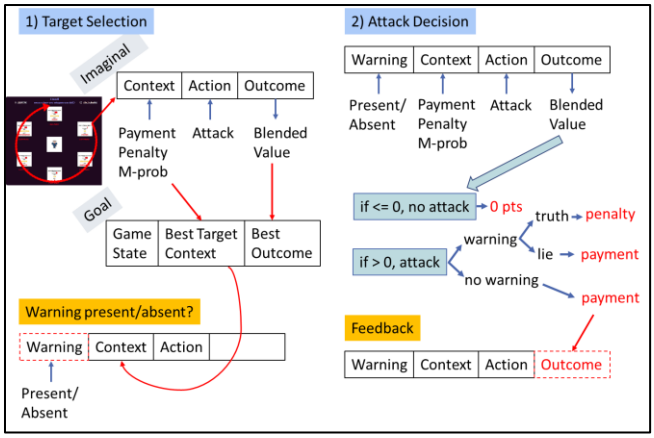

Figure 1: Model procedure.

Next, the model is presented with or without a warning signal for the selected computer. The context is augmented with a slot representing whether a warning is present or absent. In the second step of the decision process (right side of Figure 1), the model retrieves a blended value representing the updated projected outcome of attacking the computer. In the selection step, all past instances are available for retrieval. However, in the second step, only instances whose Warning slot value matches the warning signal are available to be retrieved. If the projected outcome is less than or equal to zero then the model does not attack and the outcome slot is updated with the value 0. If the projected outcome is greater than zero then the model attacks and the outcome slot is updated to reflect the ground truth outcome depending on whether the computer was monitored or not. It is this final chunk that is saved in declarative memory as a new instance.

The model behavior reflects its experiences. If an action results in a positive/negative outcome, then its expectation will be increased/lowered and the model will be more/less likely to select and attack that computer in the future. The impact of experiencing a specific outcome also strengthens with frequency and weakens with time.

The model is initialized with eight instances in declarative memory that represent the edges of the decision space. That is, the chunks represent all the combinations of a reward slot value of 0 or 10, a penalty slot value of 0 or -10, an m-prob slot value of 0.0 or 1.0, an action slot value of "attack", a warning slot value of "neutral", and an outcome slot value of 0, 10, or -10 depending on the given reward, penalty, and m-prob. With these initial chunks, the model can make a heuristically sound decision when starting the game. The activation strengths of these chunks quickly decline and they do not play a large role in subsequent decisions due to low probabilities of retrieval.

Model Results & Discussion

For each condition, the model was run 1000 times to generate stable estimates. The model was assessed based on minimizing both the probability of attacking for the post-warning decisions and the number of points earned per round.

Figure 3 shows the mean probability to attack on a given trial across rounds. In addition to the five conditions of interest, we also plotted the optimal probability of attacking given the positive expected value of the targets if no warnings were presented (*AlwaysAttack*; attack 100% of the time because the expected value of each target is greater than zero), the rationally optimal probability of attacking in the *TruthfulWarning* condition (*OptimalTruthful*; attack 66% of the time, equal to the probability of not receiving a warning in the *TruthfulWarning* condition), and the rationally optimal probability of attacking in the *DeceptiveWarning* condition (*OptimalDeceptive*; equal to the probability of not receiving a warning in the *DeceptiveWarning* condition, assuming the employee attacks only when warnings are absent, which is 0.35, 0.37, 0.33, and 0.30, for rounds 1-4 respectively).

The results indicate that including deceptive warning signals can result in a lowered probability of attack compared to other conditions – particularly compared to the *Truthful-* and *NoWarning* conditions that use the SSE without deception – but the model attacks more often than what is specified by the *OptimalDeceptive* probability of attack.

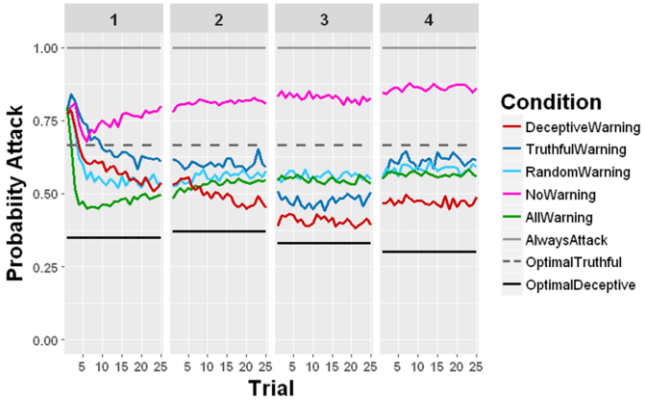

Figure 3: Probability to attack on each trial across rounds.

For all conditions, the model begins by attacking a fairly high proportion of the time, just above 75% and the pattern of attacks adapts quickly in different directions depending on the signaling condition. As expected, the model attacks most often in the *NoWarning* condition and continues to increase its rate throughout the game approaching but never reaching 100%. Although the expected value of attacking is positive, the model is deterred from attacking after experiencing a loss, but over time the model continues to attack as the gains outnumber the losses. In the *DeceptiveWarning* condition the probability of attack ultimately decreases to the lowest level of all conditions but stays just below 0.50, well above what is specified by the *OptimalDeceptive* probability. In the *TruthfulWarning* condition the model stabilizes between 0.60 and 0.65 probability of attack (except in round 3), slightly below the *OptimalTruthful* decision of attacking only when the target is not covered. In the *RandomWarning* condition, the model quickly stabilizes just below 0.60 probability of attack. In the *AllWarning* condition, the model quickly learns to refrain from attacking, likely due to receiving a penalty early on when a warning is truthful and then is deterred from attacking until those past instances decay and their activation strength weakens. The model then begins exploring the possibility of attacking and slowly attacks more often throughout the rest of the game. Interestingly, *Random-* and *AllWarning* conditions converge to roughly the same probability of attack as the *TruthfulWarning* condition, even though those signals carry no information regarding the underlying monitoring probability.

Figure 4 shows the mean probability of attacking on a given trial when a warning signal is present and when one is absent. The results indicate that when warnings are always truthful, in the *TruthfulWarning* condition, the model attacks when a warning is absent but rarely attacks when a warning is present. Interestingly, learning is slower when the warning is present than when it is absent, and the model still occasionally attacks in the presence of a warning, reflecting

uncertainty about foregone payoffs. Contrary to expectations, when deception is added in the *DeceptiveWarning* condition, the model sometimes attacks when a warning is present, at a rate about midway between when only truthful warnings are presented (*TruthfulWarning* condition) and when the warnings carry no information of the underlying monitoring probability (*Random*- and *AllWarning* conditions). However, the *DeceptiveWarning* condition is best at mitigating attacks when a warning is presented in the absence of an analyst.

In the *RandomWarning* condition, the probability of getting a warning is about 66%, but the computer is monitored only 33% of the time. Therefore, the probability of winning an attack is 66% regardless of the presence of a warning. The probability of attack of the model closely matches this probability of winning, reflecting the well-known probability matching bias (Erev & Barron, 2005). Similarly, in the *AllWarning* condition, the model closely matches the probability of winning an attack at about 66%. In both conditions, the model initially falls to 50% or below, and then slowly approaches 66% by the end of the game, suggesting that the model displays a risk aversion behavior reflecting an initial sampling bias, but that can be unlearned through experience (Lebiere et al., 2007). In the *DeceptiveWarning* condition, the probability of attack is just below 33% of the time. In this condition warnings are presented about 66% of the time, but a computer is being monitored on only 50% of those instances. Therefore, the model should attack 50% of the time when a warning is present. However, that the model attacks less than 33% of the time is indicative of another manifestation of risk aversion.

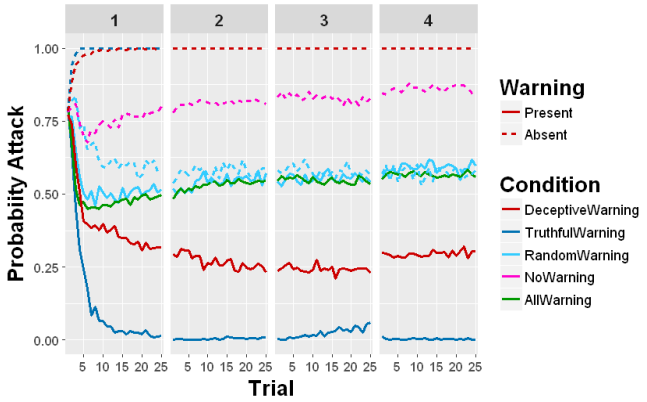

Figure 4: Probability to attack on each trial when warnings are present or absent. The *NoWarning* condition only shows a line for when warnings are absent. The *AllWarning* condition only shows a line for when warnings are present.

Figure 5 shows the mean total score obtained for each round for each condition. The model obtains a much higher score in the *TruthfulWarning* condition than in any other conditions because it suffers little loss after learning to refrain from attacking when a warning is present in order to obtain the best possible outcome on each trial. In other conditions, the model sometimes attacks when a presented warning is truthful and therefore suffers some loss. The next rewarding condition is the *DeceptiveWarning*, which is far better at protecting against loss than the *TruthfulWarning* condition, substantially minimizing the number of points obtained. When deceptive signals are used, the model attacks less often when a computer is not monitored, reducing the possible number of rewards obtained. The model also attacks more often when warnings are true, increasing the number of penalties obtained. In the *NoWarning* condition, the model obtains almost as many points as in the *DeceptiveWarning* condition, indicating that a high rate of attack results in many rewards, but also many penalties, and closely matches the expected gain based on the mean expected value of attacking ($M = 1.43$). In the *Random*- and *AllWarning* conditions, ones in which the warning signals provide no information about the underlying monitoring probabilities, the model obtains the fewest points, indicating that a random warning signal drives randomness in behavior and limits performance. However, these two conditions are worse overall in terms of mitigating attacks in the presence of warning signals.

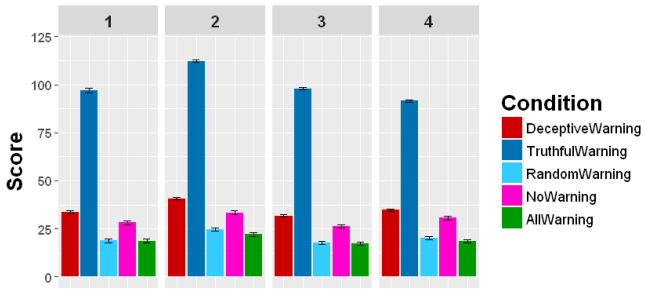

Figure 5: Mean total score across rounds.

Conclusion

The *DeceptiveWarning* condition performs best in terms of both reducing the probability of attack and minimizing loss. However, the probability of attack does not reach the level predicted by the *OptimalDeceptive* probability of attack. Why does deception not reduce the probability of attack to the *OptimalDeceptive* probability of attack? The algorithm governing how often to present a deceptive warning is designed to bring the expected value of attacking to zero. With an expected value of zero, a perfectly rational agent would choose to never attack in the presence of a warning signal, preferring to receive zero points with certainty. However, a boundedly-rational human might attack 50% of the time because the expected value of attacking is equal to the expected value of not attacking. In the Insider Attack game, the model makes risky decisions and sometimes attacks when a warning is presented, learning that it can still win sometimes even when a warning is present. Through its decisions from experience, the model learns to attack when a warning is present at a rate that is still below the probability that a warning signal is deceptive, or 50% of the time.

That the probability of attacking is just below 33% could possibly be due to a risk aversion affect (Lebiere et al., 2007). For example, in Lebiere et al. (2007), the IBL model initially makes risky decisions early on, but a series of poor payoffs results in a projected outcome that is lower than that of the

non-risky alternative. Therefore, the model avoids risky decisions and makes the safe choice more often. Similarly, in the present model, a series of penalties will lower the projected outcome and the model may attack less often. A series of non-attacks will lower the probability of attacking, but as was seen in Lebiere et al., we can expect the probability of attacking to increase to the *OptimalDeceptive* level given more trials as the increasingly long history of instances begins to match the probabilities of receiving a warning.

In conclusion, the allocation of optimal deceptive warning signals reduces the probability of attacking when a computer is not being monitored, and increases the probability of attack when a computer is truly monitored. This latter effect is actually good for cybersecurity because the attacker can be caught in such situations. The optimal deceptive signaling algorithm is effective at persuading adversarial agents to behave in a manner that benefits the defender. Using a cognitive model to test the assumptions predicted by the optimal deceptive signaling algorithm has proven useful to exploit opponent cognitive biases and in optimizing the effectiveness of the algorithm beyond what can be projected by assumptions of perfect rationality. Future research is aimed at validating the model against human participants and optimizing the defense algorithm.

Acknowledgments

This research was sponsored by the Army Research Office and accomplished under Grant Number W911NF-17-1-0370.

References

Abbasi, Y. D., Ben-Asher, N., Gonzalez, C., Kar, D., Morrison, D., Sintov, N., Tambe, M. (2016). Know your adversary: Insights for a better adversarial behavioral model. *Proceeding of the 38th Annual Conference of Cognitive Science Society* (pp.1391-1396). Austin, TX: Cognitive Science Society.

Anderson, J. R., & Lebiere, C. (1998). *The Atomic Components of Thought*. Mahwah, NJ: Erlbaum.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036-1060.

Bond Jr, C. F., & DePaulo, B. M. (2008). Individual differences in judging deception: accuracy and bias. *Psychological bulletin, 134*(4), 477.

Ekroll V., Wagemans J. (2016). Conjuring deceptions: Fooling the eye or fooling the mind?. *Trends in Cognitive Sciences*, *20*(7), 486-489.

Erev, I., & Barron, G. (2005). On adaptation, maximization, and reinforcement learning among cognitive strategies. *Psychological Review, 112*(4), 912-931.

Gonzalez, C., Ben-Asher, N., Martin, J. & Dutt, V. (2015). A cognitive model of dynamic cooperation with varied inter-dependency information. *Cognitive Science, 39*, 457-495.

Gonzalez, C., Ben-Asher, N., Oltramari, A., Lebiere, C. (2014). Cognition and Technology. In Kott, C., Wang, A. & R. Erbacher (eds.), *Cyber defense and situational awareness*. Switzerland: Springer International Publishing.

Gonzalez, C., & Dutt, V. (2011). Instance-based learning: Integrating decisions from experience in sampling and repeated choice paradigms. *Psychological Review, 118*(4), 523-551.

Gonzalez, C., & Lebiere, C. (2005). Instance-based cognitive models of decision making. In D. Zizzo & A. Courakis (Eds.), *Transfer of knowledge in economic decision-making*. Macmillan (Palgrave Macmillan).

Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance based learning in dynamic decision making. *Cognitive Science, 27*(4), 591-635.

Juvina, I., Lebiere, C., Martin, J. M., & Gonzalez, C. (2011). Intergroup prisoner's dilemma with intragroup power dynamics. *Games, 2*(1), 21-51.

Lebiere, C. (1999). A blending process for aggregate retrievals. In *Proceedings of the 6th ACT-R Workshop*. George Mason University, Fairfax, Va.

Lebiere, C., Gonzalez, C., & Martin, M. (2007). Instance-based decision making model of repeated binary choice. *Proceedings of the Eighth International Conference on Cognitive Modeling* (pp. 67-72). Oxford, UK: Taylor & Francis/Psychology Press.

Lebiere, C., Wallach, D., & West, R. L. (2000). A memory-based account of the prisoner's dilemma and other 2x2 games. *Proceedings of International Conference on Cognitive Modeling* (pp. 185-193). NL: Universal Press.

Morgan, C. J., LeSage, J. B., & Kosslyn, S. M. (2009). Types of deception revealed by individual differences in cognitive abilities. *Social neuroscience*, *4*(6), 554-569

Pita, J., Jain, M., Ordónez, F., Portway, C., Tambe, M., Western, C., and Kraus, S. (2008). ARMOR Security for Los Angeles International Airport. In *AAAI*

Riggio, R. E., & Friedman, H. S. (1983). Individual differences and cues to deception. *Journal of Personality and Social Psychology, 45*(4), 899

Rowe, N. C., & Rrushi, J. (2016). *Introduction to Cyberdeception*. Switzerland: Springer.

Sanner, S., Anderson, J. R., Lebiere, C., & Lovett, M. C. (2000). Achieving efficient and cognitively plausible learning in Backgammon. *Proceedings of the Seventeenth International Conference on Machine Learning*. San Francisco: Morgan Kaufmann.

Shieh, E., An, B., Yang, R., Tambe, M., Baldwin, C., & Meyer, G. (2012). Protect: A deployed game theoretic system to protect the ports of the United States. *AAMAS.*

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

West, R. L., & Lebiere, C. (2001). Simple games as dynamic, coupled systems: Randomness and other emergent properties. *Journal of Cognitive Systems Research, 1(4)*, 221-239.

Xu, H., Rabinovich, Z., Dughmi, S., & Tambe, M. (2015). Exploring information asymmetry in two-stage security games. *Proceedings of the National Conference on Artificial Intelligence* (*2*, pp. 1057-1063). Elsevier B.V.