1-1-2022

Markov Decision Process for Modeling Social Engineering Attacks and Finding Optimal Attack Strategies

Faranak Abri
San Jose State University, faranak.abri@sjsu.edu

Jianjun Zheng
Stephen F. Austin State University

Akbar Siami Namin
Texas Tech University

Keith S. Jones
Texas Tech University

Recommended Citation

Faranak Abri, Jianjun Zheng, Akbar Siami Namin, and Keith S. Jones. "Markov Decision Process for Modeling Social Engineering Attacks and Finding Optimal Attack Strategies" *IEEE Access* (2022): 109949-109968. https://doi.org/10.1109/ACCESS.2022.3213711

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 August 2022, accepted 13 September 2022, date of publication 10 October 2022, date of current version 20 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3213711

RESEARCH ARTICLE

Markov Decision Process for Modeling Social Engineering Attacks and Finding Optimal Attack Strategies

FARANAK ABRI[1], JIANJUN ZHENG[2], AKBAR SIAMI NAMIN[3], AND KEITH S. JONES[4]
[1]Department of Computer Science, San Jose State University, San Jose, CA 95192, USA
[2]Department of Computer Science, Stephen F. Austin State University, Nacogdoches, TX 75962, USA
[3]Department of Computer Science, Texas Tech University, Lubbock, TX 79409, USA
[4]Department of Psychological Sciences, Texas Tech University, Lubbock, TX 79409, USA

Corresponding author: Faranak Abri (faranak.abri@sjsu.edu)

This work was supported by the National Science Foundation (NSF) under Grant 1723765.

ABSTRACT It is important to comprehend the attacker's behavior and capacity in order to build a stronger fortress and thus be able to protect valuable assets more effectively. Prior to launching technical and physical attacks, an attacker may enter the reconnaissance stage and gather sensitive information. To collect such valuable data, one of the most effective approaches is through conducting social engineering attacks, borrowing techniques from deception theory. As a result, it is of utmost importance to understand when an attacker behaves truthfully and when the attacker opts to be deceitful. This paper models attacker's states using the Markov Decision Process (MDP) and studies the attacker's decision for launching deception attacks in terms of cooperation and deception costs. The study is performed through MDP modeling, where the states of attackers are modeled along with the permissible actions that can be taken. We found that the optimal policy regarding being deceitful or truthful depends on the cost associated with deception and how much the attacker can afford to take the risk of launching deception attacks. More specifically, we observed that when the cost of cooperation is low (e.g., 10%), by taking MDP optimal policy, the attacker cooperates with the victim as much as possible in order to gain their trust; whereas, when the cost of cooperation is high (e.g., 50%), the attacker takes deceptive action earlier in order to minimize the cost of interactions while maximizing the impact of the attack. We report four case studies and simulations through which we demonstrate the trade-off between cooperative and deceptive actions in accordance with their costs to attackers.

INDEX TERMS Attack strategy, cooperative, deceptive, Markov decision process, MDP, optimal solution, social engineering attacks.

I. INTRODUCTION

Social engineering remains the primary avenue for launching cyber attacks [1]. Attackers often gather required sensitive information (e.g., credentials) about the underlying infrastructure using techniques such as sending out phishing emails or texts, setting up phishing websites, executing malicious payloads on the target computer, or even through engaging with the target through interactive conversations (e.g., vishing or phishing over the phone).

In order to build a strong defense layer against harmful cyber attacks and thus understand the intention behind attacks, it is important to study the attacker's mindset and mental models that reflect how these adversarial entities decide when to demonstrate truthful or deceitful behavior. By understanding these mental models and taking into account the circumstances that may lead to such behaviors, defenders will be able to employ effective countermeasure strategies and thus be more proactive in building their defense systems.

The game-theoretical modeling of interactions between attackers and defenders has been studied in prior

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaojie Su.

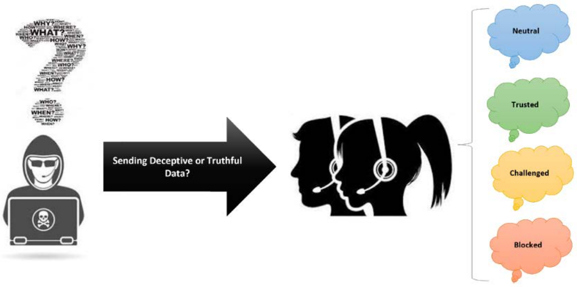


FIGURE 1. A generic social engineering attack.

research [2], [3], [4], [5], [6], [7]. In this paper, we use the Markov Decision Process (MDP) to model attacker's possible states in a generic social engineering attack and understand how the attacker decides whether to be deceptive or truthful when conducting an attack. Figure 1 illustrates a typical social engineering attack. The attacker agent on the left starts a stream of communication data with users on the right. The attacker's goal is to deceive users into disclosing their sensitive information. One of the most important techniques used by the attacker is to "*build trust*" by communicating the truthful data and then conducting an attack and finally sending the misleading data in a proper state. As a result, the attacker sends mixed signals that include both truthful and deceptive data. Users, on the other hand, comprehend the received data and, based on their perception, classify the sender as Neutral, Trusted, Challenged or Blocked and respond accordingly. The users are usually in a neutral state the first time they receive data. If the users believe the data they have received is genuine, they consider the sender to be trustworthy and respond appropriately. If users are suspicious about the received data at any stage, they place the sender in the Challenged state and expose less data in their response, or they may ask the sender some challenging questions. Finally, if the users discover a sufficient number of red flags, they place the sender in the Blocked state and prevent further communication.

To the best of our knowledge, such an MDP-based approach to model an attacker's mindset has not been discussed in the literature. The existing approaches to modeling optimal policies in the security domain primarily focus on the defender's side [8], [9], [10], [11]. While it is of utmost importance to determine the optimal strategies for defenders in order to build an effective security defense mechanism, it is also important and very informative to learn about the attacker's side and their optimal policies in order to predict their next moves and thus proactively build a strategic-based defense system. The determination of optimal attack deception strategies is the focus of this paper.

This paper models optimal policies in performing deception from the attacker's point of view. In doing so, appropriate cost parameters are incorporated into the model to reflect the potential defense strategies that can be utilized in order to secure the system. More specifically, the costs associated with deceptive and cooperative actions are formulated and controlled through a number of case studies to study and analyze their effects on the overall optimal decisions made by the attacker. The key contributions of this paper are as follows:

- Introduce an MDP-based mathematical model to represent the interaction of a deceitful attacker in a generic social engineering attack in a way that the agent may transit to different states by taking a specific action. To the best of our knowledge, there is no other similar work on this line of research and the authors are the first formulating the problem in this manner.
- Present the analysis of the MDP-based model to show the trade-off between actions and the cost of deceptive behavior and finding the optimal attack strategy,
- Evaluating the MDP optimal strategy by comparing it with the random-based strategy, and
- Report the results of a number of experiments in which a set of sensitivity analysis were performed through four case studies and simulations designed for different levels of cooperative and deceptive costs.

The paper is organized as follows: Section II reviews the literature related to modeling cybersecurity problems using MDP. In Section III, an MDP model is presented, along with the state variables, permissible actions, and their rewards in each state. The process for selecting the value for the parameters of the proposed model is discussed in section IV. Section V provides a quantitative dynamic analysis of the model and the trade-off between the costs and state values. Performance analysis of the MDP-based model is presented in Section VI. A discussion on the possible implications of the introduced model is presented in Section VII. Section VIII concludes the paper and highlights future research directions.

II. RELATED WORK

In this section, we describe different cyber security scenarios that need optimal decision-making by using game theory and MDP techniques. In each cyber security scenario, at least two agents are involved: an attacker and a defender, both of whom make decisions about how to act or respond optimally.

Kiennert et al. [12] provided a survey of game theory and MDP approaches for optimal decision-making in intrusion detection systems (IDS). They classified these optimal decisions into three main categories: Resource Allocation Optimization, IDS Configuration, and Countermeasure Optimization that can be formulated and solved using MDP techniques (i.e. single agent decision-making with uncertain outcomes) or game theory techniques (i.e. multi-agent decision-making with interaction and conflicts). They also discussed evaluation parameters, validation methodologies, limitations, and practical or real-world challenges for these techniques.

Considering single agent decision-making, Bao and Musacchio [10] focused on the defender's optimal decision. They designed their model for IDS in which the defender takes the optimal action by using MDP. Their scenario consists of three states and each state is composed of tuples which are : (NotConnected, NotDetected), (Connected, NotDetected) and (Connected, Detected). In each state, the defender can take two actions: 1) stay in that state or 2) go to the next state. During the process, the probability of each action and the defender/attacker's learning speed determine the defender's optimal policy provided by the MDP solution.

On the other hand, considering multi-agent security games [12], Casey et al. [2], proposed a model based on signaling game theory for cyber-identity management, which tries to challenge and add some cost to deceptive agents (i.e. players of the game). Their system was applied to wireless ad hoc networks (WANETs) to avoid Sybil attacks, in which an attacker tries to penetrate a network by using a non-real identity and impersonating a cooperative agent.

Moosavi and Bui [3] proposed a robust framework for intrusion detection in a wireless sensor network (WSN) by modeling it as a stochastic game. In their robust framework, the parameters of their model are not fixed and their values belong to a set. Therefore, their model is applicable to different types of WSNs.

Huang et al. [13] combined MDP and game theory techniques in their proposed IDS for wireless sensor networks. They used MDP for predicting which nodes were weakest and thus presented the greatest risk and game theory for choosing optimal defense strategies.

In certain cyber security scenarios, the defender tries to conduct deceptive actions to mislead the attacker. Because these deceptive actions may have costs, the defender needs to consider optimal decision-making.

Han et al. [14] conducted a survey about defensive deception applications in cyber security and identified four ways to model defensive deception: 1) process models, 2) probabilistic models, 3) practical models, and 4) game-theoretical models. In addition, they discussed the definition, benefits, limitations, and evaluation methods for each category. They also explained possible ways to design and deploy defensive deception in a target system. The differences and similarities of deception and moving target defense techniques are also explained.

Pawlick et al. [15] provided a survey about common game theory techniques, including Stackelberg, Nash, and signaling games for modeling defensive deception as it relates to cyber security and privacy. They also divided defensive deception applications into six categories: 1) perturbation, 2) moving target defense, 3) obfuscation, 4) mixing, 5) honey-x, and 6) attacker engagement. Each category is defined based on its structures, agents, actions, and duration. Related works are studied.

Given the moving target defense (MTD) as a defensive deception strategy, Cho et al. [16] explained several aspects of the MTD mechanism. They divided MTD methodologies into three categories: 1) shuffling, 2) diversity, and 3) redundancy. In addition, important algorithms for implementing MDT including game theory, genetic algorithms, and machine learning are discussed. They also provided a comparison between deception and MDT and other security mechanisms, especially those deception techniques used for changing the attack surface, which is the baseline of the MDT concept, considered in detail. Other aspects such as evaluation methods, including analytical models, simulation, emulation, and real test-beds, and application domains are discussed.

Crouse et al. [17] used probabilistic models to evaluate defense performance in different scenarios, including the honeypot strategy from deception defense and the shuffling strategy from moving target defense. They compared the honeypot defense model, shuffle defense model, no defense model, and a combined (shuffle and honeypot) model. To do so, they calculated the probability of attacker success in all models and also by considering two different attacks: 1) foothold attacks, in which the attacker needs only one victim node, and 2) minimum-to-win attacks, in which the attacker needs a specific number (minimum number) of victim nodes for his/her goal. Their results demonstrated that the honeypot defense strategy decreases attacker success (after a certain initial scan) in both attack types, but the combined model performed the best.

To the best of our knowledge, there is no other work that poses the problem of modeling interactions between attackers and victims from the attacker's perspective. The analysis of interactions between attackers and victims from the attacker's point of view is useful in understanding the attackers' mental models and hopefully predicting their next attacks. Indeed, we are the first to pose such a perspective and model the problem using MDP.

III. MODEL FORMULATION

The Markov Decision Process (MDP) is a modeling approach suitable for multi-stage problems with the characteristic of being partly random and partly under the control of a decision maker. In such problems, the decisions are controlled by the actions taken at each state; whereas, the state transition processes are governed by the Markov Decision Process itself. It is important to note that randomness is introduced into the system where there is a decision that needs to be made. In the case of state transitions, the underlying probability distributions define randomness in the system. In the case of decision making through actions, the decision choices and possible actions usually are drawn from a set of options that may introduce randomness to the system. For instance, consider a special case where the decision to be made is based on a split choice (50/50) of constraints and probability distributions. In this case, there is a good chance that the decision maker chooses an action randomly, since both actions may return the same number of rewards. There is a special case where the choices of decisions and actions are limited to one and thus randomness is eliminated since there is no decisions (i.e., option) to be made and it is just a deterministic finite state machine.

There may or may not be an optimal policy for any MDP. A randomized policy can be optimal when the constraints are imposed during decision making. Expressing such problems as an MDP is the first step towards solving it and it enables abstractions of a problem. Such modeling approaches offer a formalization of the underlying problem where a sequential decision-making process and strategy are needed. In more sophisticated modeling techniques such as Reinforcement Learning (RL), MDP plays a crucial role in formalizing the interaction between an agent and its surrounding environment.

Inherently, MDP-based modeling approaches enable us to maximize returns in long ranges by predicting the rewards obtained in the subsequent states. Therefore, formalizing a problem using an MDP-based model is the first step to simplifying the underlying interaction problem that can be utilized in the next stage of modeling through techniques such as reinforcement learning for optimization purposes. From the application's point of view, any problems involving interactions (e.g., game theory) can be effectively transformed to an MDP formalization. Thus, the next step or actions that can maximize the rewards or returns can be predicted [18].

In the context of social engineering attacks, an instance problem of game theory either with one player or two players, the objective of the adversarial agent is to find a way to lure the victim to respond to the requests made by the adversarial agents. Such modeling is beneficial when dealing with an interactive social engineering attack such as phishing on the phone or physical social engineering attacks where there is a sequence of messages exchanged between the two parties, and each performs specific tasks at each state to maximize their returns (i.e., two-players game) before reaching equilibrium.

The current formulation of the problem is with the assumption of a ''fully observable system'' where the agent has full knowledge, obtained through interactions with the environment, about the system and its states. This choice makes sense in the context of social engineering since the interactions between the adversarial agent and victim reveal whether the victim trusts or challenges the adversarial agent. In more complex situations, the model can be formulated using POMDP (Partial Observable MDP) or even HMM (Hidden Markov Model).

The MDP model employed in this work is based on discounted cost because there was little or no certainty about the rewards being received in future stages or actions. The model is also an infinite horizon MDP instead of a finite horizon MDP because a typical social engineering endeavor may last for a long time and thus there may not exist any upper bound for limiting the process. The finite horizon approach is usually best for cases in which there is a constraint (e.g., time) that should be considered. Having said that, it is possible to model the given problem using different MDP methods including modeling using the finite horizon approach where certain levels of reward can be treated as an upper bound and constraints. As stated earlier, the modeling of interactions in social engineering can be performed through different variations or methodologies that need to be explored in future work on this line of research.

In this study, we model the interactions in a generic social engineering attack from the attacker's point of view using Markov Decision Process (MDP). Figure 2 illustrates the overall picture of the proposed MDP-based state machine to model the attacker's states and actions.

A. MDP DESCRIPTION

The MDP is a stochastic, sequential, discrete-base model. Based on the Markov property, the state captures all relevant information from history. Once the state is known, the historical data may be discarded, i.e., the state has sufficient information about the future. A Markov process is a memoryless random process, i.e., a sequence of random states S1, S2, …with the Markov property. It is an environment in which all states are Markov which helps the simplicity of applying it to Social Engineering attacks that are not planned in detail. Policies are stationary (time-independent). In this paper, we implemented the Value Iteration method for solving MDP. As part of future work, other iterative solution methods such as Policy Iteration, Q-learning, Sarsa, etc. exist that can also be utilized for this problem [19].

MDP is formulated as a 4-tuple (S, A, P, R), where:

- S is the finite set of states.
- A is the finite set of actions.
- P is the probability of transition from one state to another upon performing an action.
- R is the expected immediate rewards received after state transition associated with the control action performed.

In order to find the optimal policy, we rely on the ''Bellman Equation'' [20] which asserts that the optimal value V^* of a decision problem at a certain point in time (i.e. state) is the expected discounted sum of rewards obtained, starting from that state and taking optimal policy π:

$$V^*(s) = \max_{\pi} E(\sum_{t=0}^{\infty} \gamma^t r_t) \tag{1}$$

Therefore, considering our problem with finite states and finite actions, the general form of the value function is as follows [21]:

$$V^*_{i+1}(s) = \max_{a \in A} \sum_{s' \in S} P(s, a, s') \left[R(s, a, s') + \gamma V^*_i(s')\right], \forall s \in S \tag{2}$$

where:

- $V^*_{i+1}(s)$ is the value function in the state s by taking the optimal action.
- $P(s, a, s')$ is the transition probability starting from state s and ending at state s' after taking action a.
- $R(s, a, s')$ is the expected rewards received after state transition from s to s' after taking action a.
- γ is the discount factor.

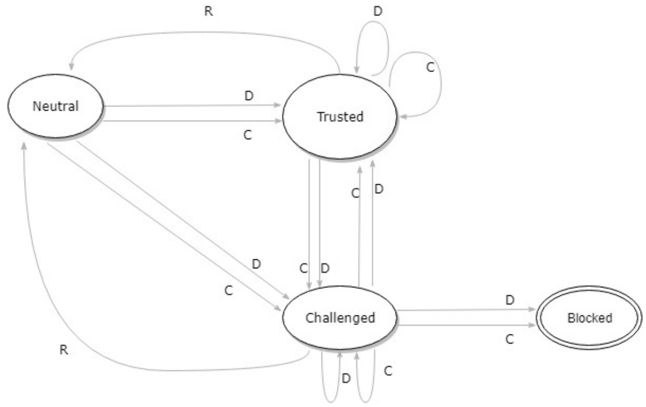


FIGURE 2. An MDP model for a generic social engineering attack: The Attacker's view. C: Cooperate; D: Deceive.

The expected instantaneous reward plus the predicted discounted value of the following state, employing the best possible action, is the value of a state s. Therefore, the optimal policy is as follows:

$$\pi^*(s) = \arg\max_{a \in A} \sum_{s' \in S} P(s, a, s') \left[R(s, a, s') + \gamma V_i^*(s') \right] \quad (3)$$

B. MODEL DESCRIPTION

The MDP model for a generic social engineering attack is implemented as below:

The four states that the attacker might encounter in this model are considered as:

$$S \in \{Neutral, Trusted, Challenged, Blocked\}$$

These states are chosen in an analogous way by defenders based on their employed defense strategies. Accordingly, the set of states may change with respect to the employed defense strategies.

The three actions the attacker might select in this model are considered as:

$$A \in \{Deceptive, Cooperative, Reset\}$$

The attackers choose these actions based on their attack strategy and the rewards and costs.

As demonstrated in Figure 2, the Neutral state is the starting point of the decision-maker (i.e., adversarial agent). The Neutral state has been defined to comply with the general formalism of MDP and finite state machine in which an initial state is needed to start the sequential processes. When a deceitful or cooperative action is demonstrated by the adversarial agent, this action will be received by the victim and thus will evoke a response. As a result of the initial deceptive or cooperative action and the victim's reaction to it, the state of the adversarial agent will be changed immediately. In other words, when the adversarial agent is in the Neutral state and the agent receives a deceptive or cooperative action, the state will change to the Trusted or Challenged state.

The adversarial agent cannot stay in the Neutral state without triggering any action. Taking any cooperative or deceitful action will take the agent to the other states. The attacker is aware of its transitioned state based on the responses it receives from the victim. For instance, if the victim asks verification questions, the adversarial agent may realize that its state is in the Challenged state; whereas, if the victim responds to questions naturally, the attacker may believe it is in the Trusted state. In an analogous way, if the attacker decides to end the conversation (e.g., not having enough resources, getting enough information from the victim, putting a delay or end the conversation before being blocked), it can go back to the Neutral state. If the victim classifies the received message as threatening, it can put the sender in the Blocked state and not let them send more messages and the attacker can recognize that it is in the Blocked state.

There are three major states (except the blockage state) that require defining value functions.

"Challenged" state

The utility function for the "*Challenged*" state is as follows, equation $V_{i+1}^*(s = C)$, as shown at the bottom of the next page.

1) "TRUSTED" STATE

The utility function for the "*Trusted*" state is as follows, equation $V_{i+1}^*(s = T)$, as shown at the bottom of the next page.

2) "NEUTRAL" STATE

The utility function for the "*Neutral*" state is as follows, equation $V_{i+1}^*(s = N)$, as shown at the bottom of the next page.

C. IMPLEMENTATION FOR FINDING OPTIMAL STRATEGY

Value iteration is a simple iterative algorithm for determining optimal value function V^* in Equation2 that converges to the right values [20], [22]. Algorithm 1 shows the pseudo-code for value iteration in MDP. First, the initial values for all states are set to zero. Next, new values are calculated for each state using Equations 2. This process is repeated until the values are reached equilibrium and do not change. In addition, a maximum number of repetitions (e.g., 1000) is taken into account to avoid falling into an infinite loop when the values are changing very slightly (a small changing value δ can also be used as the stop point) [21].

IV. INITIAL SETTINGS FOR MODEL PARAMETERS

Having presented an MDP-based model through which a set of states are defined, these states represent the states that an adversarial agent can be in while interacting with victims and launching social engineering attacks. In such MDP-based models, the obtained optimal strategy relies on the values of the model parameters including the transition probabilities and expected rewards. While the initial probability values are context-dependent, the reward values should demonstrate the gains and losses for each action taken by the adversarial agent, so the model converges properly. Table 1 shows the reward values gained when the attacker changes its state from one state to another.

These reward values should be scaled in a pre-determined range to avoid the effects of large rewards superficially, and they should be meaningful, representing the value gained through the transition from one state to another. As an example, when an adversarial agent is Challenged and triggering an action causes the adversarial agent to end up with the Blocked state, there should not be any gains (if not losses), and therefore no rewards should be given to the attacker (i.e., *reward* = 0); whereas, while in the Challenged state, the attacker tends to act smart and performs an action that gains the trust of the victim and thus ends up in the Trusted state. As a result, the adversarial attacker should gain a good number of reward points for being able to achieve the victim's trust while being challenged (i.e., *reward* = 8). These initial reward values are representative of such gains and losses when the adversarial agent triggers an action when being

$$V^*_{i+1}(s=C) = \max_{a\in A}\sum_{s'\in S} P(C,a,s')\left[R(C,a,s')+\gamma V^*_i(s')\right]$$

$$V^*_{i+1}(s=C) = \max\begin{cases} P(C,D,T)\left[R(C,D,T)+\gamma V^*_i(T)\right]+ \\ P(C,D,B)\left[R(C,D,B)+\gamma V^*_i(B)\right]+ \\ P(C,D,C)\left[R(C,D,C)+\gamma V^*_i(C)\right] \; < Deceptive > \\ \\ P(C,C,T)\left[R(C,C,T)+\gamma V^*_i(T)\right]+ \\ P(C,C,B)\left[R(C,C,B)+\gamma V^*_i(B)\right]+ \\ P(C,C,C)\left[R(C,C,C)+\gamma V^*_i(C)\right] \; < Cooperative > \\ \\ P(C,R,N)\left[R(C,R,N)+\gamma V^*_i(N)\right] \; < Reset > \end{cases}$$

$$V^*_{i+1}(s=T) = \max_{a\in A}\sum_{s'\in S} P(T,a,s')\left[R(T,a,s')+\gamma V^*_i(s')\right]$$

$$V^*_{i+1}(s=T) = \max\begin{cases} P(T,D,T)\left[R(T,D,T)+\gamma V^*_i(T)\right]+ \\ P(T,D,B)\left[R(T,D,B)+\gamma V^*_i(B)\right]+ \\ P(T,D,C)\left[R(T,D,C)+\gamma V^*_i(C)\right] \; < Deceptive > \\ \\ P(T,C,T)\left[R(T,C,T)+\gamma V^*_i(T)\right]+ \\ P(T,C,B)\left[R(T,C,B)+\gamma V^*_i(B)\right]+ \\ P(T,C,C)\left[R(T,C,C)+\gamma V^*_i(C)\right] \; < Cooperative > \\ \\ P(T,R,N)\left[R(T,R,N)+\gamma V^*_i(N)\right] \; < Reset > \end{cases}$$

$$V^*_{i+1}(s=N) = \max_{a\in A}\sum_{s'\in S} P(N,a,s')\left[R(N,a,s')+\gamma V^*_i(s')\right]$$

$$V^*_{i+1}(s=N) = \max\begin{cases} P(N,D,T)\left[R(N,D,T)+\gamma V^*_i(T)\right]+ \\ P(N,D,C)\left[R(N,D,C)+\gamma V^*_i(C)\right] \; < Deceptive > \\ \\ P(N,C,T)\left[R(N,C,T)+\gamma V^*_i(T)\right]+ \\ P(N,C,C)\left[R(N,C,C)+\gamma V^*_i(C)\right] \; < Cooperative > \end{cases}$$

Algorithm 1 Pseudo-Code of Value Iteration in MDP

```
Input
    $S$      States
    $A$      Actions
    $P$      Transition probability matrix
    $R$      Reward matrix
    $\gamma$      Discount factor
Output
    $V^*$     values for each state using utility function
    $A^*$     Optimal actions for each state (optimal policy)
─────────────────────────────
$i \leftarrow 0$
$V_i^* \leftarrow 0$
while ($i < maxItr$) do
    for ($s \in S$) do
        $V_{i+1}^*(s) \leftarrow \max_{a \in A} \sum_{s' \in S} P(s,a,s') \left[R(s,a,s') + \gamma V_i^*(s')\right]$
        $A^*(s) \leftarrow \arg\max_{a \in A} \sum_{s' \in S} P(s,a,s') \left[R(s,a,s') + \gamma V_i^*(s')\right]$
    end for
    if ($V_{i+1}^* == V_i^*$) then
        break;
    else
        $V_i^* \leftarrow V_{i+1}^*$
        $i \leftarrow i + 1$
    end if
end while
    return $V_{i+1}^*, A^*$
```

in the underlying state and trying to gain the trust of the victim.

This section presents the process and the rationales for choosing values for the parameters of the proposed model. It also demonstrates the practical implications of the introduced MDP-based deception model through simulations. The simulations present the trade-off between costs, actions, and impacts of different strategies that are available to the deceivers. The goal is to model the best deceptive or cooperative actions that the deceivers can take with respect to the costs involved in each action. As a result, the optimal policies in various scenarios are recommended to the deceivers in order to optimize their pay-offs of the game-based interactions.

A. VALUES FOR EXPECTED REWARDS

The model that we simulated is based on the Markov Decision Process model presented in Figure 2. The model consists of four states, along with actions and their probability values, as well as the rewards associated with each action. While simulating the model, we considered a few assumptions on the magnitude of the probability and rewards for each action. Some of the assumptions about the costs, rewards, and probability transitions for the deceivers are as follows:

- The cost of being deceptive is much higher than the cost of being cooperative. We consider the costs for deceptive and cooperative actions as 10 and 5, respectively (i.e., $Cost(Deceptive) > Cost(Cooperative)$).
- The rewards achieved for the transition between states depend on the source and destination states. Here are some considerations:
    - There will be some high reward if the deceivers change their status from being "*Challenged*" to a "*Trusted*" entity regardless of any actions being taken by the deceivers (i.e., cooperative or deceptive). The "*Trusted*" state is the most desirable state for deceivers and they would like to stay in this state and continue deceiving the target.
    - There will be absolutely no rewards for the cases where the deceivers are first challenged and then blocked regardless of any actions being taken by the deceivers.
    - The relative amount of rewards received by the deceivers for each change and regardless of the possible actions are as follows:

$$\begin{aligned} R(Challenged, -, Trusted) &> R(Neutral, -, Trusted) \\ &> R(Neutral, -, Challenged) \\ &> R(Challenged, -, Challenged) \\ &\geq R(Challenged, -, Neutral) \\ &\geq R(Trusted, -, Trusted) \\ &> R(Trusted, -, Challenged) \\ &\geq R(Trusted, -, Neutral) \\ &> R(Challenged, -, Blocked) \end{aligned}$$

As shown in these inequalities, without considering the action taken, the most desirable transition for the attacker is from Challenged to Trusted and the least desirable transition is from Challenged to Blocked. Being in the Neutral state, moving to Trusted is safer than moving to Challenged and so the expected reward is greater. Being in the Challenged state, the expected reward for staying in the Challenge sate is equal to or greater than that for ending the communication and going back to Neutral. Being in the Trusted state, the agent may prefer to keep the trust rather than move to Challenged, which is less safe, or ending the conversation and going back to Neutral. The effect and cost of actions taken will be considered in the next section to calculate the final expected reward values.

The initial setting and values for each reward for this simulation are listed in Table 1:

TABLE 1. Initial rewards.

Case	Rewards
$R(Challenged, -, Trusted)$	8
$R(Neutral, -, Trusted)$	5
$R(Neutral, -, Challenged)$	3
$R(Challenged, -, Challenged)$	2
$R(Challenged, -, Neutral)$	2
$R(Trusted, -, Trusted)$	2
$R(Trusted, -, Challenged)$	1
$R(Trusted, -, Neutral)$	1
$R(Challenged, -, Blocked)$	0

The discount factor is $\gamma = 0.9$, and the initial reward value is set $R = 10$. The selection of these parameters was based on the pilot study of selecting different values for each parameter and observing which value settings would better represent the problem. It is important to emphasize that different parameter settings result in different optimal policies and thus different results.

B. VALUES FOR PROBABILITY TRANSITIONS

Table 2 lists the probability values and the rewards computed by the Markov Decision Process, along with some other assumptions regarding the likelihood of the probabilities that occur for each transition.

1) NEUTRAL → {TRUSTED | CHALLENGED}

a: PROBABILITY ASSUMPTIONS

As shown in Table 2, when the attacker is in the Neutral state, there are two possible states to which the state of the attacker can transfer. These states are Trusted, which will be the new state of the attacker with the possible actions of being cooperative and deceptive with probability of $P1 = 0.7$ and $P2 = 0.3$, respectively. It is also possible to transition into the Challenged state with the same actions of being cooperative or deceptive, with probabilities of $P3 = 0.3$ and $P4 = 0.7$, respectively. As indicated in the table, our initial assumption was that the probability of changing the state of the attacker from Neutral to Trusted when the attacker opts to cooperate ($P1 = 0.7$) is much higher than the probability of ending up with the Trusted state and being deceptive ($P2 = 0.3$).

A similar justification can be made when the state of the attacker can be changed to Challenged through cooperative and deceptive actions. More specifically, the probability of ending up in the Challenged state when the attacker is deceitful ($P4 = 0.7$) should be much higher than the probability of being cooperative and still be challenged ($P3 = 0.3$).

b: REWARDS ASSUMPTIONS

Given that the ultimate goal of the attacker is to deceive as much as possible, it is desirable to remain in the Trusted state and yet perform deceptive actions. As a result, rewards that the attacker can gain for landing at the Trusted state ($R1 = 0$) are greater than the rewards that the attacker can gain by ending up in the Challenged state ($R3 = -2$) if the attacker opts to demonstrate the behavior of a good citizen. On the other hand, if the attacker decides to be deceitful with the hope of ending at the Trusted state, the rewards should be greater ($R2 = -5$) than when the attacker ends at the Challenged state ($R4 = -7$).

It is important to note that *the costs associated with cooperative and deceptive actions are already incorporated into the rewards* and therefore the rewards are a function of pure rewards minus the cost of the actions. It is also important to emphasize that the cost of being deceitful is (or should be) much greater than the cost of being cooperative.

2) TRUSTED → {TRUSTED | CHALLENGED | NEUTRAL}

a: PROBABILITY ASSUMPTIONS

As shown in Figure 2, there are three possible states that an attacker can be in if the state is Trusted. With the probability of $\pi_1 = 0.8$, it is possible to continue being a good citizen and be cooperative and thus remain in the same state (i.e., Trusted). It is also possible to take advantage of the trust and thus perform some deceitful activities with the probability of $\pi_2 = 0.3$. There is no strict relationship between π_1 and π_2 except that it is likely to observe more normal behavior than deceitful behavior from the attacker (i.e., $\pi_2 = 0.3 \leq \pi_1 = 0.8$), and yet stay at the Trusted state.

While in the Trusted state, the attacker might be challenged and thus arrive in the Challenged state. This unwanted consequence might occur due to an attacker's abnormal behavior and deceitful actions. The probability of such a transition due to deceitful action is high (π_4). That is, if the attacker demonstrates normal behavior, the landing state would likely be some other state than Challenged. However, there is a slight possibility that even with normal behavior, the attacker may end up in the Challenged state ($\pi_3 = 0.2$).

b: REWARDS ASSUMPTIONS

Intuitively, if the attacker is already in the Trusted state and continues to behave properly, it is likely to stay at the Trusted state and thus rewards would be minimal ($\rho_1 = -3$). On the other hand, if the attackers take advantage of the trust, perform deceptive actions, and still retain the trust, then the rewards would be significant ($\rho_2 = -8$).

The unwilling scenario occurs when the attacker behaves normally yet ends up in the Challenged state. Such a scenario may yield very low rewards for the attacker as reflected by the negative value ($\rho_3 = -4$). The worst scenario is when the attacker acts deceptively and then ends up in the Challenged state with minimal rewards of ($\rho_4 = -9$).

3) CHALLENGED → {TRUSTED | CHALLENGED | BLOCKED | NEUTRAL}

a: PROBABILITY ASSUMPTIONS

As modeled and shown in Figure 2, when the attacker is in the Challenged state, there are possibilities of ending in any state (i.e., Trusted, Blocked, Neutral, or even remain Challenged) with any action (i.e., being cooperative or deceitful). With respect to the probability transition assumptions and with the goal of attaining trust again and landing at the Trusted state, there is a high possibility that a challenged attacker starts to behave rationally and thus be cooperative ($\phi_1 = 0.5$). In an analogous way, there is a small probability that the attacker will demonstrate deceitful behavior ($\phi_2 = 0.1$) yet hope to gain trust and thus change the state to Trusted. It is also possible that with some probability ($\phi_5 = 0.3$ and $\phi_6 = 0.2$) the attacker may not be able to convince the other party and thus remain in the Challenged state. The worst scenario occurs when with some probability of actions ($\phi_3 = 0.2$ and $\phi_4 = 0.7$) the attacker ends up in the Blocked state.

TABLE 2. Expected rewards and probability transition values ("≪" means much smaller than).

State			Converged		
Start	End	Action	Probability	Rewards	Initial Assumptions
s	s'	a	$P(s,a,s')$	$R(s,a,s')$	
Neutral	Trusted	Cooperative	$P1 = 0.7$	$R1 = 0$	1) Probability: $P3 \leq P2 \ll P4 \leq P1$ 2) Rewards: $R4 \ll R2 < R3 \ll R1$
Neutral	Trusted	Deceptive	$P2 = 0.3$	$R2 = -5$	
Neutral	Challenged	Cooperative	$P3 = 0.3$	$R3 = -2$	
Neutral	Challenged	Deceptive	$P4 = 0.7$	$R4 = -7$	
Trusted	Trusted	Cooperative	$\pi_1 = 0.8$	$\rho_1 = -3$	1) Probability: $\pi_3 \leq \pi_2 \leq \pi_4 \leq \pi_1 < \pi_5$ 2) Rewards: $\rho_4 < \rho_2 \ll \rho_3 < \rho_1$
Trusted	Trusted	Deceptive	$\pi_2 = 0.3$	$\rho_2 = -8$	
Trusted	Challenged	Cooperative	$\pi_3 = 0.2$	$\rho_3 = -4$	
Trusted	Challenged	Deceptive	$\pi_4 = 0.7$	$\rho_4 = -9$	
Trusted	Neutral	Reset	$\pi_5 = 1.0$	$--$	
Challenged	Trusted	Cooperative	$\phi_1 = 0.5$	$\omega_1 = +3$	1) Probability: $\phi_2 < \phi_6 \leq \phi_3 < \phi_5 < \phi_1 < \phi_4 < \phi_7$ 2) Rewards: $\omega_4 < \omega_7 \simeq \omega_6 < \omega_3 < \omega_5 < \omega_2 \ll \omega_1$
Challenged	Trusted	Deceptive	$\phi_2 = 0.1$	$\omega_2 = -2$	
Challenged	Blocked	Cooperative	$\phi_3 = 0.2$	$\omega_3 = -5$	
Challenged	Blocked	Deceptive	$\phi_4 = 0.7$	$\omega_4 = -10$	
Challenged	Challenged	Cooperative	$\phi_5 = 0.3$	$\omega_5 = -3$	
Challenged	Challenged	Deceptive	$\phi_6 = 0.2$	$\omega_6 = -8$	
Challenged	Neutral	Reset	$\phi_7 = 1.0$	$\omega_7 = -8$	

b: REWARDS ASSUMPTIONS

When the attacker is in the Challenged state, there should be high rewards if the attacker can gain the trust and move back to the Trusted state ($\omega_1 = +3$ and $\omega_2 = -2$). As a matter of fact, the rewards should be higher when the attacker deceives and yet gains the trust of the system. On the other hand, there should be huge negative rewards when the attacker loses the game and end up in the Blocked state ($\omega_3 = -5$ and $\omega_4 = -10$). Given the high cost of deception for the attacker, the negative rewards gained by conducting expensive deception and landing at the Blocked state should be extremely high ($\omega_4 = -10$). A moderate level of negative rewards ($\omega_5 = -3$ and $\omega_6 = -8$) should be also considered when the attacker is still struggling to convince the victim and remains in the Challenged state.

V. QUANTITATIVE DYNAMIC ANALYSIS

This section presents the numerical results of the presented model. The numerical results are presented in terms of 1) MDP states (i.e., Neutral, Challenged, Trusted, and Blocked), 2) Actors' actions (i.e., cooperative or deceptive), and the cost associated with each action. For the sensitivity analysis, we change one of the costs and fix the second one to demonstrate the effects of high or low costs on the optimal policy and model. Without loss of generality, the fixed values for costs are 10% and 50%.

We believe further analysis of showing the impact of changes in the parameter settings is a valuable topic and should provide additional insights for researchers. The analysis presented in this section contains and presents interesting results about the sensitivity of the model to different settings that can be helpful in designing such an application for a real problem domain and can shed light on the practical implications of the proposed abstract model.

The case studies demonstrate realistic scenarios in which the cooperation and deception costs are controlled to observe each scenario's best-optimized policy or strategy. These four scenarios are part of use cases of a more extensive and more comprehensive social engineering detection framework in which exchanged communication data (e.g., textual or verbal) are utilized to decide about the state of the attacker or defender in each stage. In such a framework, the state (i.e., Neutral, Trusted, Challenged, and Blocked) of each party (i.e., either attacker or defender) is determined using the exchanged communication data. To estimate the state of each party, contemporary machine learning and, in particular, Natural Language Processing (NLP) techniques can be employed.

The four scenarios resemble different scenarios where the associated costs are different. For instance, when the cooperation cost is minimal (i.e., case study I in which the cooperation cost is equal to 10%), the adversarial agent may decide to be more cooperative in order to gain the trust of the victim. Whereas, if the cooperation cost is high (i.e., case study II in which the cooperation cost is equal to 50%), it is costly for the adversarial agent to be cooperative and thus the agent needs to find a way to lure the victim as soon as possible in order to reduce its cost but successfully luring the victim.

Similarly, when the deception cost is minimal (i.e., case study III in which the deception cost is equal to 10%), the adversarial agents may conduct a great number of deceptive actions in order to lure the victim. Whereas, when the cost of deception for the adversarial agent is high (i.e., case study IV in which the deception cost is equal to 50%), the adversarial agent may be cautious about taking a deceptive action in order not to end up with the Blocked state.

Figures 3, 4, 5 and 6 illustrate these four case studies. In these figures, the x-axis depicts the deceptive or cooperative cost and the y-axis depicts the state value, which is the expected return while being in the underlying state and taking the corresponding action of cooperative, deceitful, or rest. Put differently, the state value shows how much return we expect when taking each action.

A. OPTIMAL POLICY AND DECEPTIVE COST

In the first two case studies, we consider cases in which the cooperative costs are set at a fixed rate and then we study the model for various levels of deception costs. Hence, the process is similar to a typical sensitivity analysis, in which one parameter is controlled at a certain level, and the other parameter is changed systematically to observe the impacts of changes.

1) CASE STUDY I: *COOPERATIVE COST* = 10%

Figure 3 illustrates the trend of state values for each action (i.e., Cooperative, Deceptive, and Reset) when the deceptive cost increases. The x-axis represents the deceptive cost changing between 0.05 and 1.0 with steps of 0.05 unit, where the deceptive cost is calculated and then deducted based on some percentages of the rewards; whereas, the y-axis represents the state value when each action is performed.

a: CHALLENGED STATE

As illustrated in Figure 2, an attacker can take three permissible actions while in the Challenged state: 1) be cooperative (C), 2) be deceptive (D), or reset (R) and change the state to Neutral.

Figure 3.(a) illustrates the trend of state value when the attacker is in the Challenged state. As illustrated in the figure, with the increase of deception cost, and when the attacker is in the Challenged state, the state values for all permissible actions decrease. The state value for deceptive action starts just below 5 with the Deceptive cost of 0.05 and then smoothly decreases to −5 when the cost of being deceptive reaches 1. However, being Cooperative and/or Resetting offer better state values compared to the Deceptive action. Therefore, while being in the Challenged state, the attacker is better off being Cooperative or Resetting than being Deceptive and thus avoiding the potential of ending in the Blocked state.

The state values of being cooperative and resetting start at 17 and 20, respectively, suggesting that when the deception cost is at 0.05, it is a better option to reset than cooperate. The recommended policy by the model, however, changes with the increase of deception cost. When that occurs, cooperative and reset actions gain equal state values for the attacker when the deception cost is 0.25. The increase of deception cost from 0.25 to 1 increases the likelihood of cooperative actions compared to exhibiting any other actions due to the high risk of being identified and thus blocked. As a result, there is a mixed suggestion of actions that can be taken by the attacker while maximizing the state value.

Figure 3.(d) illustrates the optimal policy when the deception cost changes between 0.05 and 1 and the attacker is being challenged and thus cooperative cost is fixed at 10% of rewards. The optimal policy is taken from the upper bound of the fitted curves in Figure 3.(a) for each action with the objective of maximizing the state value. According to the figure, the optimal action policy for the attacker, when the attacker is in the Challenged state, is to reset if the deceptive cost is below 0.025, and be cooperative otherwise, when the deception cost is greater than 0.25.

b: TRUSTED STATE

As illustrated in Figure 2, an attacker can take two permissible actions while in the Trusted state: 1) be cooperative (C) or 2) be deceptive (D).

Figure 3.(b) demonstrates the trend of state values with respect to changes in the deception cost when the attacker is in the Trusted state. As shown in the figure, it is equally beneficial to take either action (i.e., being cooperative or deceptive) when the deception cost is below 0.15. However, the state value gained by the deception action drops substantially when the deception cost is increased; whereas, the state value gained by being cooperative remains steady at 10.

Figure 3.(e) depicts the optimal policy for this case, which is drawn from the upper bound of curves depicted in Figure 3.(b). As the figure suggests, an attacker may opt to take the action of being deceptive (i.e., the attacker goal) when the deception cost is below 0.15, even though taking the action of being cooperative is also possible. However, if deception cost increases to 0.15 or greater, it is better for the attacker to be cooperative in order to avoid the risk of being challenged again.

c: NEUTRAL STATE

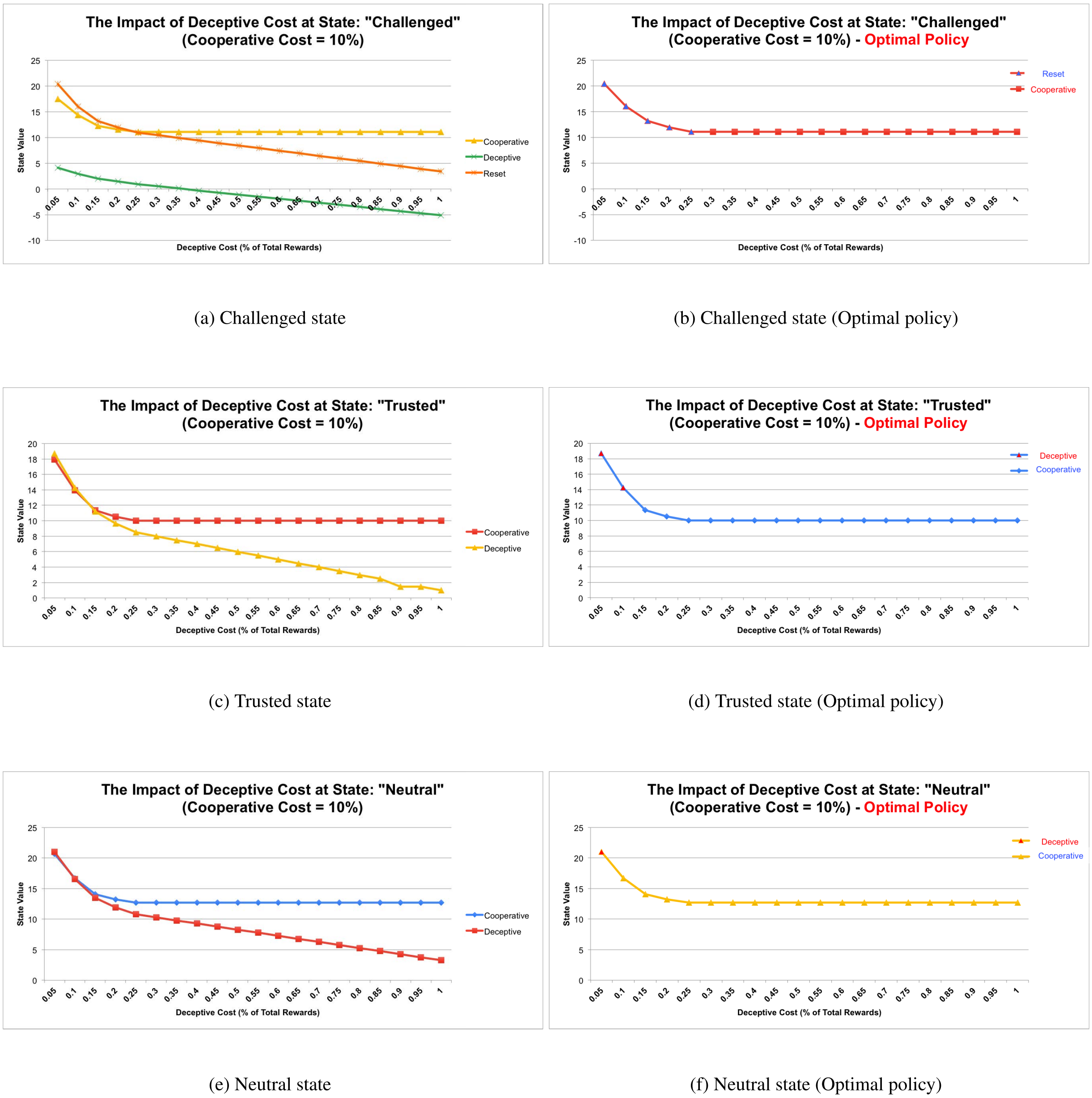
As illustrated in Figure 2, an attacker can take two permissible actions while in the Neutral state: 1) be cooperative (C) or 2) be deceptive (D).

As demonstrated in Figure 3.(c), A similar trend is observed when the attacker is in the Neutral state. Given equal state values gained by both actions (i.e., cooperative and deceptive), and with respect to the goal of the attacker to be more deceptive, it is more beneficial for the attacker to be deceptive when the deception cost is below 0.1. However, if the cost of deception increases above 0.1, and given the risk of being challenged, it is better for the attacker to be more cooperative than deceptive. The state value gained by being deceptive dramatically drops to 3.5 when the deception cost is elevated to 1; whereas, the state value gained by being cooperative remains steady at 13 when the deception cost is greater than 0.15.

Figure 3.(f) demonstrates the optimal policy when the attacker is in the Neutral state. As demonstrated in the figure, the attacker may be deceptive when the deception cost is below 0.1. As soon as the deception cost increases above 0.1, the attacker needs to be more cooperative in order to avoid being challenged, avoiding the expensive consequences associated with being challenged.

2) CASE STUDY II: *COOPERATIVE COST* = 50%

In Case Study I, we maintained the cooperative cost at 10% of the reward. In order to study whether an increase in the cooperation cost would affect the optimal policy and thus the actions that are recommended to the attacker, in this case study, we increase the cost of being cooperative to 50% of the



(a) Challenged state

(b) Challenged state (Optimal policy)

(c) Trusted state

(d) Trusted state (Optimal policy)

(e) Neutral state

(f) Neutral state (Optimal policy)

FIGURE 3. Optimal policy at different states when *cooperative cost* = 10%.

total rewards and then capture how the dynamics of the model change. Figure 4 illustrates the trends for state values for the three attacker states (i.e., Challenged, Trusted, and Neutral) along with the permissible actions the attacker can take at each state.

a: CHALLENGED STATE

Figure 4.(a) illustrates the trends of state values for the three permissible actions when the attacker is in the Challenged state. Similar to Figure 3, the x-axis depicts the increase of deception cost in the range of 0.05 and 1 with the step size of 0.05; whereas, the y-axis depicts the state values gained by taking the corresponding actions.

As seen in the figure, given how risky it is for the attacker to be in this state (i.e., Challenged), it is better to either reset actions and move to the Neutral state or be more cooperative than deceitful. The state values gained by being deceitful, cooperative, or resetting actions are 4.5, 14, and 20, respectively, when the deception cost is at 0.05. The simulation suggests the attacker should reset and move to the Neutral state rather than taking any other action. This trend for state values remains the same until the deception cost is increased to 0.25% of the total reward, at which point the attacker can take risks and try deceiving the target. The trend remains the same when the cost of deception is even greater.

Figure 4.(d) illustrates the optimal policy for such a setting, taking the upper bound of the curves drawn in Figure 4.(a), with the goal of maximizing the state values for the attacker. As the figure suggests, it is better for the attacker to reset actions and return to the Neutral state when the deception cost is below 0.25. However, if the cost of deception increases, then it is better for the attacker to take the risk to start deceiving the target. The primary reason for this recommendation is that the cooperation cost is already at its peak, and thus it is not beneficial for the attacker to remain cooperative, and the attacker can try to reach its goal in deceiving the target.

Figure 4.(d) demonstrates the situation in which the attacker is being challenged and the optimal policy (i.e., upper bound of curves in Figure 4.(a)) actions that are recommended according to the level of the deception cost. As the figure suggests, it is recommended that the attacker reset the activity and return to the Neutral state if the deception cost is below 0.25. However, it is recommended that the attacker perform some deceitful action when the deception cost is greater than 0.25.

b: TRUSTED STATE

Figure 4.(b) compares the state value gains obtained by performing cooperative or deceptive actions when the attacker is in the Trusted state. Inspection of the figure suggests it is best for an attacker to deceive when they are in the Trusted state and cooperation cost is high (i.e., 50%), regardless of any consequences. The primary reason for that recommendation is due to the high cost of cooperation and relatively low cost of deception. Therefore, it is intuitive and reasonable to conduct deception at all costs.

Figure 4.(e) demonstrates the optimal policy (i.e., the upper bound of curves in Figure 4.(b)) for such a situation in which it is recommended to take the deception action regardless of deception cost.

c: NEUTRAL STATE

Similar and intuitive results are also obtained when the attacker is in the Neutral state (Figure 4.(c)). Given the high cost of cooperation and low cost of deception, it is best for the attacker to deceive the target up to a certain level of deception cost. More specifically, inspection of the figure suggests that the attacker should deceive the target when the deception cost is below 0.7; whereas the attacker should cooperate when the deception cost elevates above 0.7. The point at which the policy changes (e.g., 0.7), is called the “*turning point*”, where another action will be taken to maximize the state value. Figure 4.(f) illustrates the optimal policy drawn from the upper bounds of curves drawn in Figure 4.(c) where the turning point (i.e., 0.7) decides about changes in the policy.

B. OPTIMAL POLICY AND COOPERATIVE COST

In the last two case studies, we explore cases in which the deception costs are set at the fixed rate and then we study the model for various levels of cooperative costs.

1) CASE STUDY III) *DECEPTIVE COST* = 10%

a: CHALLENGED STATE

Figure 5.(a) demonstrates the trend of state values for permissible actions when the attacker is in the Challenged state. Because the attacker has already been challenged, it makes sense to be cautious and thus to not play deceitful games. Inspection of the figure suggests the state value is around 3 and remains steady regardless of changes in cooperative costs. On the other hand, both cooperative and reset actions are recommended to the attacker as a better choice than being deceitful. The state value for the reset and cooperative actions are 18 and 16, respectively. However, as the cooperative cost grows, the model suggests taking the conservative action of resetting the entire activity and returning to the Neutral state.

Figure 5.(d) demonstrates the optimal policy for this case in which it has been recommended to take the reset action and thus avoid the risk of being blocked.

b: TRUSTED STATE

Figure 5.(b) represents the recommended actions when the attacker is in the Trusted state, with the deceptive cost fixed at 10% of the total rewards. Inspection of the figure suggests, given the low cost of deception, it is recommended to be deceptive regardless of the cost of being cooperative. The state value for deceptive action starts at 16 when the cooperation cost is at 0.05 and remains steady when the cooperative cost elevates even to 1.

Figure 5.(e) shows the optimal policy for this case taken from the upper bound of the curves shown on 5.(b), suggesting taking the action of deception at all risks. This recommendation is expected because there is little risk and harm to the attacker.

c: NEUTRAL STATE

A similar recommendation is given when the attacker is in the Neutral state. Figure 5.(c) depicts the trend of state value for both cooperative and deceptive actions. However, in this case, it is recommended to be cooperative when the cooperation cost is below 0.1. Once the cooperation cost is greater than 0.1, however, it is recommended to start deceiving the target.

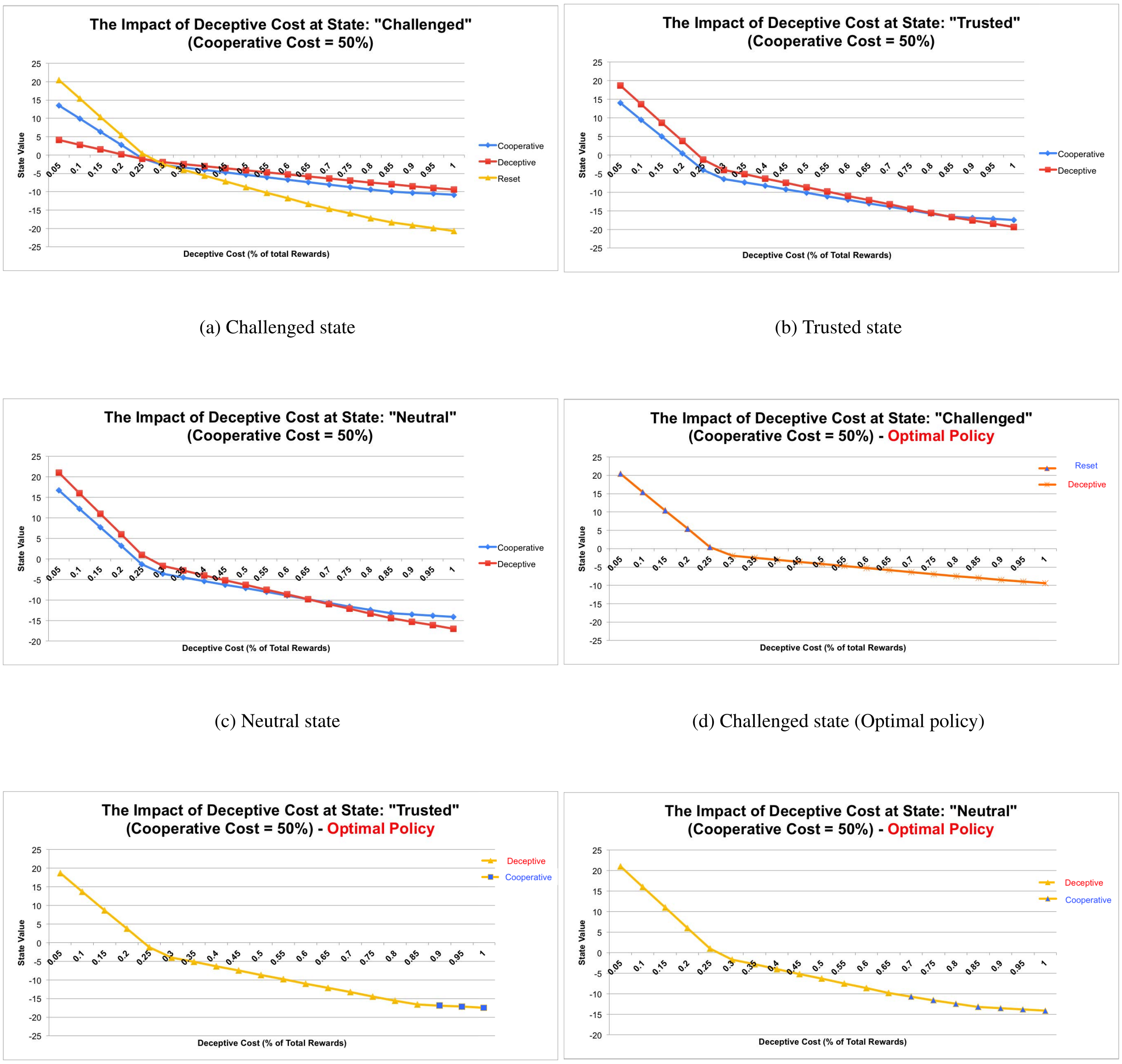
Figure 5.(f) illustrates the optimal policy for this case when it is highlighted to be cooperative when the cooperative cost is below 0.1, but perform deception otherwise.

2) CASE STUDY IV) *DECEPTIVE COST* = 50%

In this case, we set the deception cost at a higher level (i.e., 50%) and observe the behavior of the model and thus capture the optimal policy recommended in this case.

a: CHALLENGED STATE

Figure 6.(a) illustrates the trends of state values for the reset, cooperative, and deceptive actions. Inspection of the figure suggests it is not recommended to take deceptive action when the cooperative cost is low. The state value starts with 0 at the



(a) Challenged state

(b) Trusted state

(c) Neutral state

(d) Challenged state (Optimal policy)

(e) Trusted state (Optimal policy)

(f) Neutral state (Optimal policy)

FIGURE 4. Optimal policy at different states when *cooperative cost* = 50%.

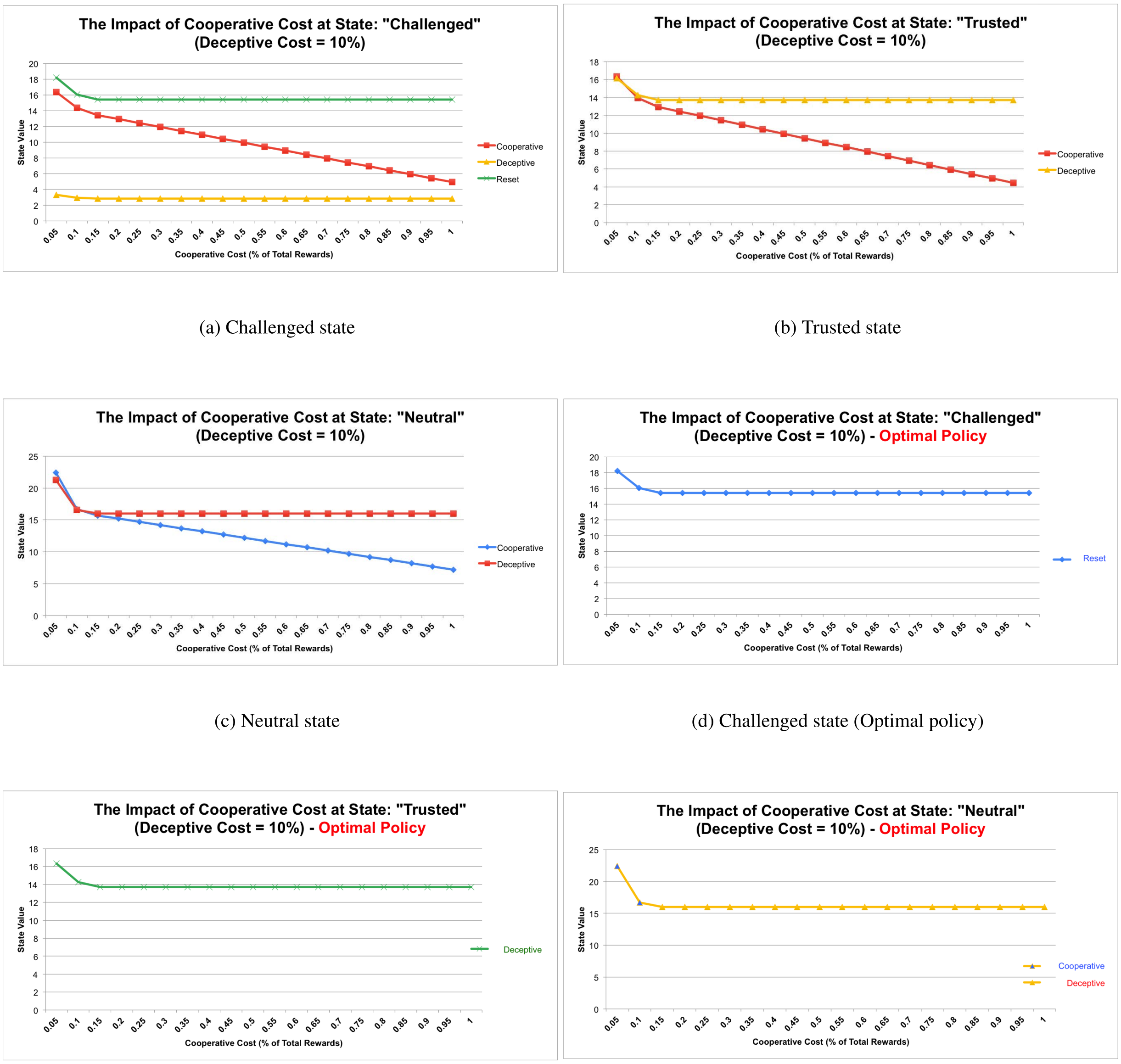
0.05% level of the cooperative cost and turns into a negative value −5 at 1% of cooperative cost. On the other hand, it is highly recommended to be cooperative when the cooperative cost is below 0.35. Once the cooperative cost goes beyond 0.35, then deceptive action is recommended.

Figure 6.(d) shows the optimal policy for this case taken from the upper bound of curves represented in Figure 6.(a). As is apparent from this optimal policy, it is recommended to be cooperative when cooperative cost is low, <0.35. However, once the cooperation cost elevates beyond 0.35, it is recommended to be deceitful.

b: TRUSTED STATE

Figure 6.(b) illustrates the case when the attacker is in the Trusted state. Both cooperative and deceptive actions demonstrate a similar trend for the state value. However, there is a turning point when the deception cost is at 0.9 where the optimal policy changes. More specifically, when the cost of deception is below 0.9 it is recommended to be deceptive. However, when the turning point of 0.9 is reached, it is recommended to be cooperative.

Figure 6.(e) shows the optimal policy for this case illustrating the turning point and policy changes at 0.9.



(a) Challenged state

(b) Trusted state

(c) Neutral state

(d) Challenged state (Optimal policy)

(e) Trusted state (Optimal policy)

(f) Neutral state (Optimal policy)

FIGURE 5. Optimal policy at different states when *deceptive cost* = 10%.

c: NEUTRAL STATE

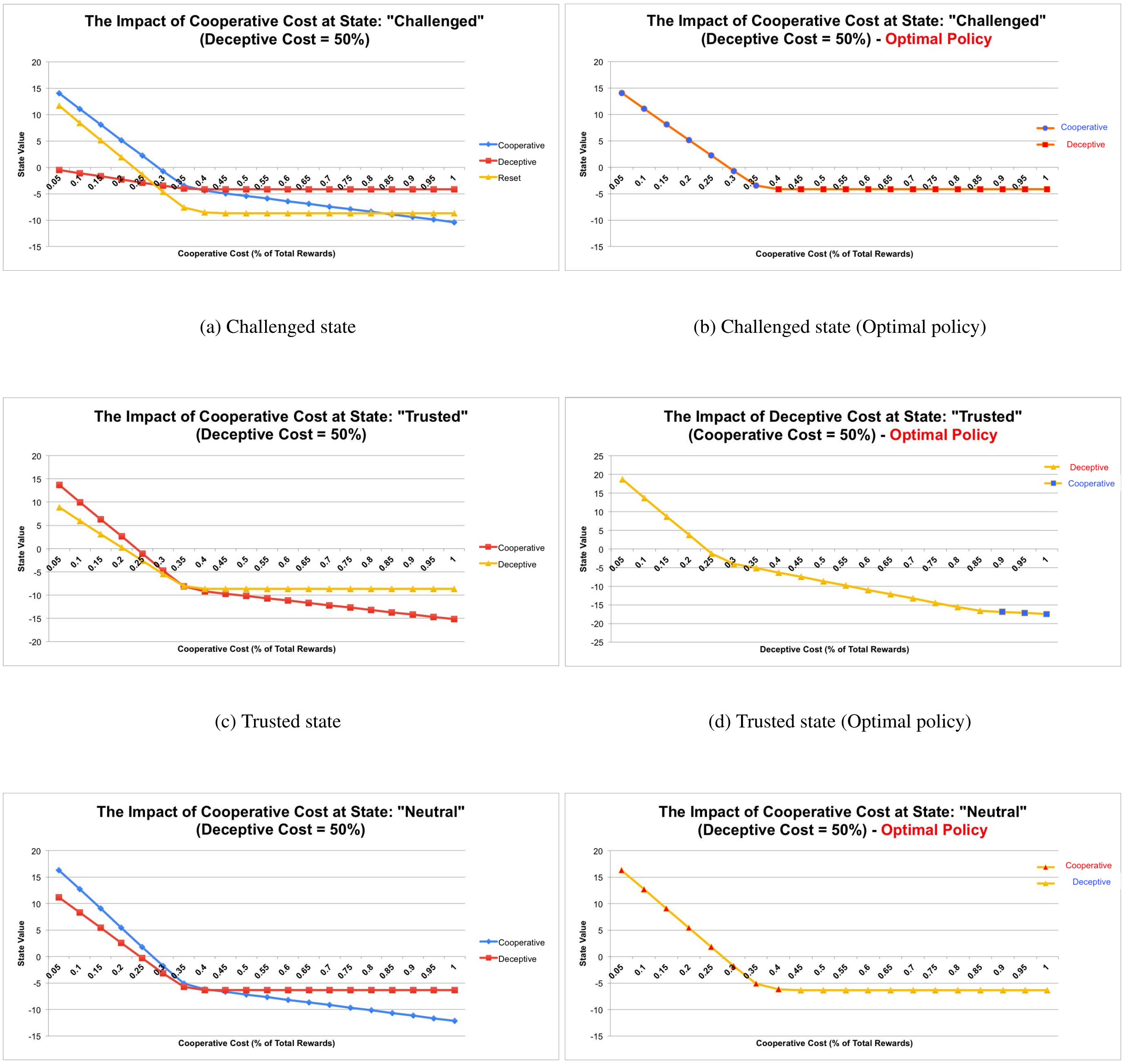
A similar observation is produced when the attacker is in the Neutral state (Figure 6.(c)). There is a turning point at 0.4% of the cooperative cost. Accordingly, it is recommended to be cooperative when the cooperation cost is below 0.4. However, it is recommended to be deceptive once the turning point is passed and cooperation has a high cost.

Figure 6.(f) illustrates the optimal policy for this case demonstrating the turning point and the optimal policy for this case.

VI. PERFORMANCE ANALYSIS

A typical issue of modeling interactions through probabilistic settings, and in particular models such as MDP and Reinforcement Learning, is that these models are very sensitive to the initial probabilistic values set for each parameter. This section investigates this issue through a numerical case study and reports the performance of the model when the parameter settings and values are controlled. The main objectives of this case study were:

– To identify the optimal strategy by MDP using the new set of values for model parameters,
– To analyze the performance/effectiveness of the obtained optimal strategy on the model and capture the accumulative rewards and the number of steps that the adversarial agent can continue interactions before being blocked,
– To compare the performance/effectiveness of the optimal strategy with a purely random strategy and compare



(a) Challenged state

(b) Challenged state (Optimal policy)

(c) Trusted state

(d) Trusted state (Optimal policy)

(e) Neutral state

(f) Neutral state (Optimal policy)

FIGURE 6. **Optimal policy at different states when *deceptive cost* = 50%.**

their acquired rewards and number of steps taken by the agent before being blocked.

Taking Optimal Actions. Having the above objectives in mind, first the MDP optimal strategy for the attacker was obtained using algorithm1. The agent's behavior (i.e., the attacker) was then simulated in the model by applying the MDP optimal policy through the following steps:

1) Start with the Neutral state, as the initial step of the adversarial agent,
2) Choose the *MDP Optimal Action "a_{mdp}"* for the underlying state (i.e., "s"),
3) Based on the chosen action, identify the states that can be transitioned to (i.e., S'_p) from state "s" where S'_p is the subset of states that can be reached from *s* by action "a_{mdp}",
4) Taking into account the transition probabilities of S'_p, proceed with taking the transition to the next state by considering the weights defined for each of the transition probabilities,
5) Receive the reward for the transition and add it to the previous rewards and also increase the number of steps by one,
6) if the new state is the Blocked state then return the rewards and the number of steps; otherwise go to Step 2

Taking Random Actions. In a similar manner, the authors repeated the same steps with the exception of taking *random*

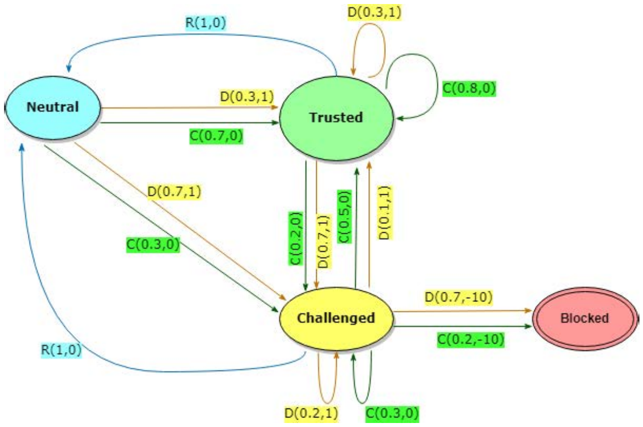


FIGURE 7. **The proposed MDP model with values for rewards and transition probabilities.**

actions instead of taking the steps identified by the MDP optimal strategy (Step 2) to compare the results.

Rules for selecting values for Transition Probability. For transition probabilities, we considered the following rules and constraints:

- The summation of transition probabilities from one state to another for each action needs to be equal to 1. In the case of obtaining the summation not equal to one, we can standardize the values to ensure they fall into the proper interval of [0, 1].
- Taking the Cooperative action, the probability of transitioning to the Trusted state should be greater than the probability of transitioning to the Challenged state. Furthermore, the probability of transitioning to the Challenged state should be greater than the probability of transitioning to the Blocked state.
- The probability of transitioning to the Blocked state should be greater than the probability of transitioning to the Challenged state. Taking the Deceptive action, the probability of transitioning to the Challenged state should be greater than the probability of transitioning to the Trusted state.

It should be noted that, in this model, the Reset action can be taken only on the Trusted and Challenged states, and the destination state will be the Neutral state. Therefore, if the agent takes the Reset action, it will transit to the Neutral state with the probability equal to 1.00.

The Neutral state is the start state and also the safest state for the adversarial agent to remain in the model. Therefore, during the communication, if the attacker feels threatened by being blocked, it can take the Reset action and transit to the Neutral state to refresh its status. The Reset action can be considered as asking the receiver to ''end the communication for a while'', which would allow the attacker to collect more information with the goal of launching more effective attacks or even changing the attack strategy.

Rules for selecting values for expected rewards. For transition rewards, we considered the following simple rules:

- Taking the Deceptive action and not transitioning to the Blocked state, the agent will be rewarded by one point ($r = +1$).
- Taking the Deceptive action and transitioning to the Blocked state, the agent will be penalized by 10 point($r = -10$).
- Taking the Cooperative and Reset actions, do not bring any rewards to the agent.

Having regulated the rules for transition probabilities and awards, the considered values to examine the performance of the model are shown in Figure 7. The label of the edges is shown by the name of the action (C as Cooperative, D as Deceptive and R as Reset) followed by the pair of transition probability and reward. For instance, $D(0.7, -10)$ on the edge from the Challenged state to the Blocked state implies that being in the Challenged state and taking the Deceptive action, the agent transitions to the Blocked state with the probability of 0.7 and the obtained rewards will be equal to -10, which means the agent will incur a big penalty for being blocked. To ease the visualization of the actions, a coloring scheme has been adopted in Figure 7 where Reset, Cooperative, and Deceptive actions are colored in blue, green, and yellow. The Probability Transition Matrix (T) and the Reward matrix (R) values are shown in Table 3. These matrices can also be implemented as 3D matrices as below by considering state indices Neutral=0, Trusted=1, Challenged=2, Blocked=3, equations T and R, as shown at the bottom of the page.

Algorithm 2 lists the pseudocode for evaluating the value iteration MDP for the proposed model. In this algorithm, optimal actions, A^*, are obtained for each state using pseudocode 1. Starting from the initial state *Neutral*, the optimal action for Neutral state $A^*(s = Neutral)$ is extracted. Given that it is possible to transition to *Trusted* and *Challenged* states with different probabilities, the next state is selected randomly following the distribution probability of the state. Being in the current state and having the optimal action in

$$T = \begin{bmatrix} [[0, .7, .3, 0], [0, .8, .2, 0], [0, .5, .3, .2], [0, 0, 0, 0]], & a = c \\ [[0, .3, .7, 0], [0, .3, .7, 0], [0, .1, .1, .6], [0, 0, 0, 0]], & a = d \\ [[0, 0, 0, 0], [1, 0, 0, 0], [1, 0, 0, 0], [0, 0, 0, 0]], & a = r \end{bmatrix}$$

$$R = \begin{bmatrix} [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, -10], [0, 0, 0, 0]], & a = c \\ [[0, 1, 1, 0], [0, 1, 1, 0], [0, 1, 1, -10], [0, 0, 0, 0]], & a = d \\ [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]], & a = r \end{bmatrix}$$

TABLE 3. Expected rewards and probability transition values considered for the performance analysis conducted for simulation of the numerical case study.

State			Converged	
Start	**End**	**Action**	**Probability**	**Rewards**
s	s'	a	$P(s,a,s')$	$R(s,a,s')$
Neutral	Trusted	Cooperative	$P1 = 0.7$	$R1 = 0$
Neutral	Trusted	Deceptive	$P2 = 0.3$	$R2 = 1$
Neutral	Challenged	Cooperative	$P3 = 0.3$	$R3 = 0$
Neutral	Challenged	Deceptive	$P4 = 0.7$	$R4 = 1$
Trusted	Trusted	Cooperative	$\pi_1 = 0.8$	$\rho_1 = 0$
Trusted	Trusted	Deceptive	$\pi_2 = 0.3$	$\rho_2 = 1$
Trusted	Challenged	Cooperative	$\pi_3 = 0.2$	$\rho_3 = 0$
Trusted	Challenged	Deceptive	$\pi_4 = 0.7$	$\rho_4 = 1$
Trusted	Neutral	Reset	$\pi_5 = 1.0$	$\rho_5 = 0$
Challenged	Trusted	Cooperative	$\phi_1 = 0.5$	$\omega_1 = 0$
Challenged	Trusted	Deceptive	$\phi_2 = 0.1$	$\omega_2 = 1$
Challenged	Blocked	Cooperative	$\phi_3 = 0.2$	$\omega_3 = -10$
Challenged	Blocked	Deceptive	$\phi_4 = 0.7$	$\omega_4 = -10$
Challenged	Challenged	Cooperative	$\phi_5 = 0.3$	$\omega_5 = 0$
Challenged	Challenged	Deceptive	$\phi_6 = 0.2$	$\omega_6 = 1$
Challenged	Neutral	Reset	$\phi_7 = 1.0$	$\omega_7 = 0$

the current state along with the next selected state, the reward is extracted. This process is repeated to transition from one state to the next, adding the rewards obtained in each iteration until the agent transitions to the *Blocked* state or reaches the maximum number of moves.

Algorithm 2 Pseudo-Code of Evaluating MDP

```
Input
    S        States
    A        Actions
    P        Transition probability matrix
    R        Reward matrix
    A*       MDP optimal actions for each state (optimal policy)
    maxItr   Maximum number of iterations (stop point)
Output
    itrs     number of iterations
    R*       Summation of the awards obtained)
$i \leftarrow 0$
$s_i \leftarrow Neutral$
$R^* = 0$
while $(s_i! = Blocked \ \& \ i < maxItr)$ do
    $i \leftarrow i + 1$
    $A_i \leftarrow A^*(s_i)$
    $S_{i+1} \leftarrow weightedRandomChoice(P, S_i, A_i)$
    $R^* \leftarrow R^* + R(S_i, A_i, S_{i+1})$
    $S_i \leftarrow S_{i+1}$
end while
    $itrs \leftarrow i$
    return $R^*, itrs$
```

Having applied pseudocode 1 on this model, the optimal solution was found after 335 iterations. More specifically, the optimal strategy using the value iteration algorithm were:

- The optimal action when the agent is in the Neutral state is being Deceptive,
- The optimal action when the agent is in the Trusted state is also being Deceptive,
- The optimal action when the agent is in the Challenged state is taking the Reset action.

Following that optimal strategy, the attacker avoids being blocked. Utilizing the MDP optimal policy for the attacker and limiting the maximum number of steps to 30, the adversarial agent was able to stay in the model for all the 30 steps by not being blocked while gaining +23 points. However, taking random actions, the agent was blocked after only 4 steps and losing −7 points.

Given that the transited states are chosen by weighted random selection, we repeated the process 10 times (i.e., episode) to make sure that the failure of the random strategy to save the adversarial agent from being blocked did not happen by chance. Table 4 shows the result of running the process for each episode, the average number of steps, and the acquired rewards for both strategies. Performing the experiment for 10 episodes, MDP with its optimal policy, on average obtained +21.7 points without being blocked, while taking random actions, on average, the agent paid −7.5 as the penalty and transitioned to the Blocked state after approximately 7 random actions.

TABLE 4. The performance of MDP-based optimal vs. random-based adversarial agents.

	MDP		Random	
Episode No	Reward	#Actions	Reward	#Actions
1	+23	30	-7	4
2	+21	30	-9	3
3	+23	30	-7	7
4	+23	30	-9	2
5	+23	30	-7	5
6	+20	30	-3	20
7	+21	30	-7	6
8	+20	30	-10	2
9	+22	30	-10	3
10	+21	30	-6	12
Average	+21.7	30	-7.5	6.4

As indicated in Table 4, using the MDP optimal strategy, the adversarial agent still plays the game without being blocked (i.e., $\#actions = 30$, the upper limit considered in the model); whereas, the random-based adversarial agent is blocked within a very short number of actions.

VII. DISCUSSIONS AND IMPLICATIONS

The presented model can be applicable to different settings and problems. This section discusses the scenarios and cases where the MDP-based model can be adapted. Furthermore, we discuss the automation aspects of the model as well as the possible implications of the presented model for defenders.

A. MODEL USABILITY

To understand the usability and the applications of the presented model better, two points should be considered:

Firstly, this model presents a *general-purpose model* for the attacker's optimal decision strategy. By general-purpose model, we mean that not only can this model be adapted when the attacker faces different cyber security defense scenarios (e.g., IDS, MDT, honeypot, phishing, etc.), but also in different phases of the attack (e.g., reconnaissance, scanning, exploitation, access maintenance, etc.) in which the attacker encounters a diverse form of defense infrastructures.

Secondly, this model provides a *general view* for modeling different attack scenarios and also attack phases. By general view, we mean this model can serve as the basis for similar models with a different set of states and/or actions based on the defender or the attacker's infrastructures. For example, one can add another state such as a ''wait'' state to the attacker's state in which the attackers take some time to be stealthy or gain enough credits before conducting a costly deceptive action that they could not afford before.

Given these applications of the model, we conclude that this model has broad applications in attack and defense modeling. As an example, when attackers face defense systems, such as IDS, WANET, WSN, MDT, honeypot, mixed networks, and perturbation, they may adapt similar models to decide when to launch attacks in order to elevate their gains and attack the target more effectively. Another example is the situation in which an attacker conducts phishing or vishing (i.e., phishing over the phone) attacks in the reconnaissance phase or a Sybil attack in the exploit or access maintenance phase.

In conclusion, the key point for designing such a model in other usages is to design the states, actions, and transition parameters (such as probabilities, costs, rewards, and learning rate) properly and then by forming the utility function the optimal decision and policies can be obtained. The designing process presented through this model can be in its simplest version in which only one agent (attacker or defender) is making the decision and the states are as few as possible (e.g. only two states: start-state and end-state) and fixed parameters (such as fixed costs) or it can be more complicated with additional states, more actions, and various parameters. Finally, it should be mentioned that such a model can be designed from other agents' point of view (e.g. the defenders) or considering several interacting agents.

B. AUTOMATION

The model introduced in this paper can be used as a base for further automation in launching attacks. Furthermore, a complementary model also can serve as a model for defending systems. There are some other mathematical models such as reinforcement learning and hidden Markov model (HMM) that can be integrated with the proposed MDP-based model for cases when the exact modeling states are unknown in advance. In particular, the basic foundation of reinforcement learning is the Markov Decision Process. The reinforcement learning module augment and enhance the model with the capability of learning from the environment and thus enhancing the performance of the model in suggesting the best optimal actions where there are some uncertainty. On the other hand, HMMs are capable of estimating the current state of the agent where there is little to no knowledge about the surrounding areas and conditions.

C. IMPLICATIONS

The proposed decision-making model for optimal policy can have several implications for defenders and system administrators. The model demonstrates the influence of deception and cooperation costs from the attacker's point of view. According to the model and the trade-off analysis presented in this paper, the attackers are reluctant to perform any deceptive actions if the cost is high for such malicious actions. As a result it is recommended to keep the cost of deception as much as possible and maintain a relatively low cost for cooperative actions in order to prevent possible deceptive activities. These types of analysis can be further incorporated into risk analysis and management in order to find out about the attackers' tolerance level in absorbing the cost and launching attacks.

D. LIMITATIONS

Kiennert et al. [12] discuss the limitations of game-theoretic approaches as the main concept of ''high abstraction level,'' which is due to enormous assumptions made by the designer and results in a challenge for real-world applications and consequently a challenge for validating such an actual system. Some of these assumptions are the perfect rationality of the agents, the model designing assumptions, complete information about the costs and rewards, fixed and not changing costs and rewards, and so on.

Some of these limitations hold also for our proposed model. The main limitation for models similar to the proposed model is ''*model designing assumptions*'' such as fixed sets of states and actions for the agent. This hurdle is worst for scenarios that the opponent agent is a human compared to a machine or an automated agent. For example, when the attackers deploy vishing (i.e., phishing over phone) attacks on their victims, designing the states that the attacker will end up might not be that simple due to unpredictability of the human responses. We mitigated this drawback by considering the attacker perspective for designing our model which makes the action set determined by the attackers and not their opponents.

VIII. CONCLUSION AND FUTURE WORK

Humans are the weakest link in information security [23], so it is impossible to ignore the critical role of human operators in functioning critical infrastructure. Unsurprisingly, humans have been the prime targets for attacks due to the low cost yet effective outcomes of such attacks. According to analysis and reports, phishing is the number one cause of data breaches [24]. This type of attack can be launched through various forms of channels such as emails, phones, and shoulder-surfing. To launch effective social engineering attacks, attackers are thus utilizing techniques drawn from deception theory. In order to build a strong fortress around critical infrastructure, it is essential to study the attacker's mindset and predict their next move.

This paper modeled the deceiver's strategies and policy optimization for optimally conducting deceitful activities. We formulated the optimal decision problem through a Markov Decision Process (MDP), thus being able to yield the dynamic characteristics of deception in social engineering attacks. By modeling and analyzing the attacker's behavior, the defenders are able to learn about deception strategies and thus guard themselves against such attacks. Through the presented MDP-based model for capturing deceiver's optimal policy in deception, we are able to predict the attacker's next move and thus be prepared for such deceptions and social engineering attacks.

The objective of formulating the problem using MDP from the attacker's point of view is to enhance the practice of ethical hacking and penetration testing practices by extracting and analyzing the best optimal policy for the attacker to launch more cost-effective social engineering attacks. As a result, the immediate practical implication of this research result is that the attacker needs to identify the cost of being cooperative or deceitful as well as the consequences of being challenged or blocked. By knowing the cost, learning about the victim, and the probability transitions, then a typical penetration tester or ethical hacker can decide on how to approach the victim and then make a proper and informed decision when taking the next step.

A critical factor that influences the attacker's behavior in conducting deception attacks is the costs associated with cooperation and deception. According to the MDP-based model presented in this paper, attackers should be reluctant to perform deceitful activities if the deception cost is high relative to the cooperation cost, which discourages attackers from launching any deception attacks.

There are several security controls that might help in elevating the cost associated with deception. For instance, employing proper and continuous authentication schemes would help in preventing adversaries from launching socio-technical attacks. Another example would be proper registrations along with the effective collection of credentials of individuals in interactive settings. This way, we are able to verify the identity and credentials of individuals. These security controls hurdle adversaries of performing social engineering attacks without revealing their own true identities.

An interesting application of the MDP-based model is to trap adversaries by *intentionally* keeping the cost of deception low. Such settings can be built in honeypots with the goal of identifying potential attackers. Furthermore, it will be extremely useful in analyzing attacker's behavior and strategies in deceiving individuals who have access to critical information. In particular, this model would help in the automated detection of interactive social engineering attacks such as phishing over the phone (i.e., vishing). Such a strategy would help organizations to learn about attackers and what assets they target and thus tighten security controls for prime targets. Furthermore, it would be very useful to understand the capability of attackers with respect to cost and how much security controls would be needed for protecting critical assets.

As part of future work, it would be useful to model both attacker and defender's behavior and build an interactive MDP game-based model with the goal of analyzing their optimal decisions under various conditions and contexts. It is important to learn about the attacker's capacity and willingness to conduct deception attacks in terms of payoffs between costs and gains. The important aspects that are beneficial to learn about attackers are their ultimate decisions when stronger and more preventive security controls are incorporated into the system. Such conditions and contexts can be formulated using entropy-based information gain analysis and reasoning under uncertainty.

AVAILABILITY

The simulation code along with the data and analysis will be made available upon request.

REFERENCES

[1] Virginia Information Technologies Agency. (2017). *Common Wealth of Virginia Information Security Report*. [Online]. Available: https://www.vita.virginia.gov

[2] W. Casey, A. Kellner, P. Memarmoshrefi, J. A. Morales, and B. Mishra, "Deception, identity, and security: The game theory of sybil attacks," *Commun. ACM*, vol. 62, no. 1, pp. 85–93, Dec. 2018.

[3] H. Moosavi and F. M. Bui, "A game-theoretic framework for robust optimal intrusion detection in wireless sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 9, pp. 1367–1379, Sep. 2014.

[4] Y. W. Law, T. Alpcan, and M. Palaniswami, "Security games for risk minimization in automatic generation control," *IEEE Trans. Power Syst.*, vol. 30, no. 1, pp. 223–232, Jan. 2015.

[5] T. Alpcan and T. Basar, "A game theoretic approach to decision and analysis in network intrusion detection," in *Proc. 42nd IEEE Int. Conf. Decis. Control*, Dec. 2003, pp. 2595–2600.

[6] K.-W. Lye and J. M. Wing, "Game strategies in network security," *Int. J. Inf. Secur.*, vol. 4, nos. 1–2, pp. 71–86, 2005.

[7] K. Sallhammar, "Stochastic models for combined security and dependability evaluation," Ph.D. thesis, Fac. Inf. Technol., Math., Elect. Eng., Dept. Telematics, Norwegian Univ. Sci. Technol., Trondheim, Norway, 2007.

[8] J. Zheng and A. S. Namin, "Markov decision process to enforce moving target defence policies," 2019, *arXiv:1905.09222*.

[9] Q. Zhu and T. Basar, "Dynamic policy-based IDS configuration," in *Proc. 48th IEEE Conf. Decis. Control (CDC) Held Jointly 28th Chin. Control Conf.*, Dec. 2009, pp. 8600–8605.

[10] N. Bao and J. Musacchio, "Optimizing the decision to expel attackers from an information system," in *Proc. 47th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2009, pp. 644–651.

[11] D. Shen, G. Chen, J. Cruz, L. Haynes, M. Kruger, and E. Blasch, "A Markov game theoretic data fusion approach for cyber situational awareness," *Proc. SPIE*, vol. 6571, Apr. 2007, Art. no. 65710F.

[12] C. Kiennert, Z. Ismail, H. Debar, and J. Leneutre, "A survey on game-theoretic approaches for intrusion detection and response optimization," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–31, Sep. 2019.

[13] J.-Y. Huang, I.-E. Liao, Y.-F. Chung, and K.-T. Chen, "Shielding wireless sensor network using Markovian intrusion detection system with attack pattern mining," *Inf. Sci.*, vol. 231, pp. 32–44, May 2013.

[14] X. Han, N. Kheir, and D. Balzarotti, "Deception techniques in computer security: A research perspective," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–36, Jul. 2019.

[15] J. Pawlick, E. Colbert, and Q. Zhu, "A game-theoretic taxonomy and survey of defensive deception for cybersecurity and privacy," *ACM Comput. Surv.*, vol. 52, no. 4, p. 82, Aug. 2019.

[16] J.-H. Cho, D. P. Sharma, H. Alavizadeh, S. Yoon, N. Ben-Asher, T. J. Moore, D. S. Kim, H. Lim, and F. F. Nelson, "Toward proactive, adaptive defense: A survey on moving target defense," 2019, *arXiv:1909.08092*.

[17] M. Crouse, B. Prosser, and E. W. Fulp, ''Probabilistic performance analysis of moving target and deception reconnaissance defenses,'' in *Proc. 2nd ACM Workshop Moving Target Defense*. New York, NY, USA: Association for Computing Machinery, Oct. 2015, pp. 21–29.

[18] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Cambridge, MA, USA: MIT Press, 2018.

[19] M. A. Samsuden, N. M. Diah, and N. A. Rahman, ''A review paper on implementing reinforcement learning technique in optimising games performance,'' in *Proc. IEEE 9th Int. Conf. Syst. Eng. Technol. (ICSET)*, Oct. 2019, pp. 258–263.

[20] R. Bellman, *Dynamic Programming*. Mineola, NY, USA: Dover, 1957.

[21] L. P. Kaelbling, M. L. Littman, and A. W. Moore, ''Reinforcement learning: A survey,'' *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 237–285, Jan. 1996.

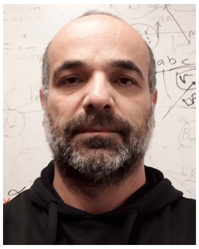
[22] D. P. Bertsekas, *Dynamic Programming: Deterministic and Stochastic Models*. Upper Saddle River, NJ, USA: Prentice-Hall, 1987.

[23] Kratikal Tech Pvt. Ltd. (2017). *Humans are the Weakest Link in the Information Security Chain*. [Online]. Available: https://medium.com/

[24] S. Shelley. (2019). *Phishing Number One Cause of Data Breaches: Lessons From Verizon DBIR*. [Online]. Available: https://info.phishlabs.com/blog/phishing-number-1-data-breaches-lessons-verizon

FARANAK ABRI received the second master's and Ph.D. degrees in computer science from Texas Tech University, in 2020 and 2022, respectively. She is currently an Assistant Professor of computer science at San Jose State University. She has research experience in a wide range of topics in this area, including malware analysis, cloud security, automated deception detection, social engineering, security comprehension, and usable security. Her research interest includes modeling cybersecurity problems using artificial intelligence (AI) and machine learning (ML) techniques.

JIANJUN ZHENG received the first master's degree in statistics, and the second master's and Ph.D. degrees in computer science from Texas Tech University, Lubbock, in 2004, 2013, and 2020, respectively. He is currently an Assistant Professor of computer science at Stephen F. Austin State University. His research interests include modeling moving target defense and network defense strategy optimization.



AKBAR SIAMI NAMIN received the Ph.D. degree in computer science from Western University, London, ON, Canada, in August 2008. He is currently an Associate Professor of computer science at Texas Tech University. He has coauthored over 80 research articles published in premier journals and venues. His research on cyber security research and education is funded by the National Science Foundation. His research interests include software engineering, testing and program analysis, software and cyber security and malware analysis, machine learning, and deep learning.

KEITH S. JONES is currently a Human Factors Psychologist and a Professor at Texas Tech University who specializes in human–computer interaction. To date, he has been awarded over $2.9M in research funding from the National Science Foundation, the Office of Naval Research, the Air Force Office of Scientific Research, and Microsoft, and has published numerous journal articles in peer-reviewed outlets and conference proceeding papers at international venues. His current research interests include human–robot interaction and human factors issues related to cybersecurity.