ACIG APPLIED CYBERSECURITY & INTERNET GOVERNANCE

NASK

Predictive Modelling of a Honeypot System Based on a Markov Decision Process and a Partially Observable Markov Decision Process

Received: 27.11.2022
Accepted: 02.01.2023
Published: 04.01.2023

Lidong Wang Institute for Systems Engineering Research Mississippi State University, Mississippi, USA, ORCID: 0000-0003-3923-849X
Reed Mosher Institute for Systems Engineering Research Mississippi State University, Mississippi, USA
Patti Duett Institute for Systems Engineering Research Mississippi State University, Mississippi, USA
Terril Falls Institute for Systems Engineering Research Mississippi State University, Mississippi, USA, ORCID: 0009-0006-4468-4928

Cite this article as:
L. Wang, R. Mosher, P. Duett, T. Falls "Predictive Modelling of a Honeypot System Based on a Markov Decision Process and a Partially Observable Markov Decision Process," ACIG, vol. 2, no. 1, 2023, DOI: 10.5604/01.3001.0016.2027

Corresponding author:
Lidong Wang, Institute for Systems Engineering Research Mississippi State University, Mississippi, USA; ORCID: 0000-0003-3923-849X; E-mail: lidong@iser.msstate.edu

Abstract

A honeypot is used to attract and monitor attacker activities and capture valuable information that can be used to help practice good cybersecurity. Predictive modelling of a honeypot system based on a Markov decision process (MDP) and a partially observable Markov decision process (POMDP) is performed in this paper. Analyses over a finite planning horizon and an infinite planning horizon for a discounted MDP are respectively conducted. Four methods, including value iteration (VI), policy iteration (PI), linear programming (LP), and Q-learning, are used in the analyses over an infinite planning horizon for the discounted MDP. The results of the various methods are compared to evaluate the validity of the created MDP model and the parameters in the model. The optimal policy to maximise the total expected reward of the states of the honeypot system is achieved, based on the MDP model employed. In the modelling over an infinite planning horizon for the discounted POMDP of the honeypot system, the effects of the observation probability of receiving commands, the probability of attacking the honeypot, the probability of the honeypot being disclosed, and transition rewards on the total expected reward of the honeypot system are studied.

Keywords

cybersecurity, honeypot, machine learning, Markov decision process, partially observable Markov decision process, Q-learning

1. Introduction

Cybersecurity is concerned with the privacy and security of computers or electronic devices, networks, and any information that is stored, processed, or exchanged by information systems [1]. Parameter design, monitoring, and network maintenance are important to network cybersecurity. The detection and prevention of attacks are generally more significant than any subsequent actions taken after being attacked [2]. It is helpful to obtain as much information as possible from attacks to defend against attackers and improve the cybersecurity of information systems [3]. A honeypot system can collect information from an attack about the attackers and may aid in the practice of robust cybersecurity. A honeypot is used to attract attackers and record their activities [4].

Attackers can be attracted to a fake system by a honeypot in the network infrastructure; valuable information can be obtained from them; and the information can then be used to improve network security [4]. A honeypot constitutes a useful technique or tool to observe the spread of malware and the emergence of new exploits. An attacker tries to avoid connecting to a honeypot as it can disclose the attacker's tools, methods, and exploits [5]. A honeypot is also a source that can be leveraged to build high-quality intelligence against threats, providing a means for monitoring attacks and discovering zero-day exploits [6]. A network honeypot is often used by information security teams to measure the threat landscape for the security of their networks [7]. One example of a stochastic process method, the MDP, has been used for decision-making in cybersecurity. The MDP assumes that both defenders and attackers have observable information, although this is not true in many applications [8]. In actuality, there may be partial observability or an agent's inability to fully observe the state of its environment in numerous real situations [9]. In many real-world problems, their environmental models are not known. There is a considerable need for reinforcement learning to solve problems where agents partially observe the states of their environments (possibly due to noise in the observed data). This leaves the outcomes of actions under uncertainty more dependent on the signal of the current state. The POMDP extends the MDP by permitting a decision-making process under uncertain or partial observability [10]. The artificial intelligence (AI) world has shown a huge leap recently in the research area of the POMDP model [11].

An MDP model for interaction honeypots was created and an analytic formula of the gain was derived. The optimal policy was decided based on comparing the calculated gain of each policy and selecting the one with a maximal gain. The model was then extended using a POMDP. One approach to solving the POMDP problem was proposed. In this method, the system state was replaced with the belief state and the POMDP problem was converted into an MDP problem [12]. The efforts in the research of this paper were to fulfil predictive modelling of the honeypot system, based on the MDP and the POMDP. Various methods and algorithms were used, including VI, PI, LP, and Q-learning in the analyses of the discounted MDP over an infinite planning horizon. The results of these algorithms were evaluated to validate the created MDP model and its parameters. In the modelling of the discounted POMDP over an infinite planning horizon, the effects of several important parameters on the system's total expected reward were studied. These parameters include the observation probability of receiving commands, the probability of attacking the honeypot, the probability of the honeypot being disclosed, and the transition rewards. The analyses of the MDP and POMDP in this paper were conducted using the *R* language and *R* functions. This paper is organised as follows: the second section introduces the methods of MDP and POMDP; Section 3 presents a created MDP model of the system and the parameters in the model; Section 4 shows the analyses of the system based on the MDP method; Section 5 presents analyses of the system based on the POMDP method, and the final section is the conclusion.

2. Methods

2.1. The MDP

The MDP method is one of the most significant methods employed in artificial intelligence (especially machine learning). The MDP is described using the tuple <S, A, *T*, *R*, γ> [13–15]:

- S is the states' set.
- *A* is the actions set.
- *T* is the transition probability from the state s to the state s' (s ∈ S, s' ∈ S) after action a ($a \in A$).
- *R* is an immediate reward after action a, and
- γ ($0 < \gamma < 1$) is the discounted factor.

An optimal policy is the goal of the MDP that maximises the total expected reward. An optimal policy over a finite planning horizon maximises the vector of the total expected reward until the horizon ends. The total expected reward (discounted) for an infinite planning horizon is employed to evaluate the gain of the discounted MDP in this paper.

2.2. The Algorithms of the MDP

VI, PI, LP, and Q-learning have been the algorithms utilised to find an optimal policy for the MDP. Theoretically, the results of the four kinds of algorithms should be the same. However, the results obtained using the algorithms may potentially differ with a great value, or convergence problems may potentially occur during the iterative process if the created MDP model is unreasonable, owing to unsuitable structure or incorrect model parameters. Thus, all the algorithms are employed, and their results are evaluated to validate the model constructed in this paper.

VI: An optimal policy for the MDP can be achieved by utilizing VI when the planning horizon is finite. In principle, the four algorithms (VI, PI, LP, and Q-learning) can be employed to find the optimal policy when the planning horizon is infinite. VI utilises the following equation of value iterations [16–18] to calculate the total expected reward for each state:

$$V(s) := max_a \sum_{s'} T(s, a, s')(R(s, a, s') + \gamma V(s')) \quad (1)$$

where $T(s, a, s')$ is the transition probability from state s to state s' after action a. $R(s, a, s')$ is the immediate reward of the transition. $V(s)$ and $V(s')$ are the total expected reward in state s and state s', respectively. When the value difference between 2 consecutive iterative steps is lower than the given tolerance, the iteration will be stopped.

PI: A better policy is found using PI, through comparing the current policy to the previous one. PI generally begins arbitrarily with an initial policy and then policy evaluation and policy improvement are followed. The process of iterations continues until the same policy is obtained for 2 successive policy iterations, indicating that the optimal policy has been achieved. For each state s, Equation (2) is used for policy evaluation and Equation (3) is used for updating the policy (policy improvement) [16, 18].

$$V(s) := max_a \sum_{s'} T(s, \pi(s), s')(R(s, \pi(s), s') + \gamma V(s')) \quad (2)$$

where π(s) is an optimal policy of state s.

$$\pi(s) = argmax_a \left(\sum_{s'} T(s, a, s')(R(s, a, s') + \gamma V(s'))\right) \quad (3)$$

LP: Since the MDP can be expressed as a linear program, the LP can find a static policy through solving the linear program. The following LP formulation [19] is used to find the optimal value function:

$$\underset{V}{Solve}\ \min \sum_{s \forall S} V(s) \quad (4)$$

subject to

$$V(s) \geq R(s, a, s') + \gamma \sum_{s' \in S} T(s, a, s')\ V(s') \quad (5)$$

Q-learning: It is used to achieve the best policy with the greatest reward. It is a reinforcement learning method and allows an agent to learn the Q-value function that is an optimal action-value function. Q-learning can also be applied to non-MDP domains [20]. The action-value function Q(s, a) is expressed as follows [21]:

$$Q(s, a) = \sum_{s'} T(s, a, s')(R(s, a, s') + \gamma V(s')) \quad (6)$$

Q(s, a) can be initialised arbitrarily (for example, Q(s, a) = 0, $\forall s \in S$, $\forall a \in A$). From state s to state s', a Q-learning update can be defined as follows [21, 22]:

$$Q(s, a) := (1 - \beta)Q(s, a) + \beta[R(s, a) + \gamma \max_{a} Q(s', a)] \quad (7)$$

where $\beta \in (0, 1)$ represents the learning rate. The best action a at state s can be chosen according to the optimal policy π(s). The iterative process continues until the final step of episode. The optimal policy is described as follows:

$$\pi(s) = \underset{a \in A}{arg\ \max}\ Q(s, a) \quad (8)$$

2.3. The POMDP

A POMDP can be thought as a generalisation of an MDP, permitting state uncertainty in a Markov process [23]. In POMDP applications, the objective is generally to obtain a decision rule or policy to maximise the expected long-term reward [24]. In the POMDP, the belief state is a distribution of probabilities over all possible states. An optimal action relies only on the current belief state [25].

The POMDP was defined as a tuple <S, A, T, R, O, B, γ> [26]:

- O = {o_1, o_2, ..., o_k} is an observation set.
- B is a set of conditional observation probabilities $B(o|s', a)$. s' is the new state after the state transition $s \rightarrow s'$, $o \in O$.
- S, A, T, R, and γ are the same as those in the tuple of MDP.

After having taken the action a and observing o, the belief state needs to be updated. If b(s) is the previous belief state, then the new belief state [25] is given by

$$b'(s') = aP(o|s') \sum_{S} P(s'|s, a)\ b(s) \quad (9)$$

where a is a normalizing constant that makes the belief state sum to 1.

The goal of POMDP planning is to obtain a sequence of actions $\{a_0, a_1, ..., a_t\}$ at time steps that maximise the total expected reward [27], i.e., we choose actions that give

$$\max E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right] \quad (10)$$

where s_t and a_t are the state and the action at time t, respectively.

The optimal policy brings up the greatest expected reward for each belief state, which is the solution to the Bellman optimality equation through iterations beginning at an initial value function for an initial belief state. The equation can be formulated as [12]:

$$V(b) = max_{a \in A}[b(s)R(s, a) + \gamma \sum_{o \in O} P(o|b, a)V(b')] \quad (11)$$

3. The MDP Model of the Honeypot System

3.1. The Structure of the MDP Model

The honeypot system is a network-attached system that is put in place to lure attackers. A botnet is utilised to forward spam, steal data, etc. A botmaster keeps a bot online. A honeypot has three states [12]:

- State 1: Not attacked yet (waiting for an attack to join the botnet).
- State 2: Compromised (becoming a member of the botnet).
- State 3: Disclosed (not the botnet's member anymore) due to the real identity having been discovered or interactions with the botmaster having been lost for an extended period of time.

A honeypot can take one of the following actions at each state:

- Action 1: Allows a botmaster to compromise the honeypot system and to implement commands.
- Action 2: Does not allow the botmaster to compromise the system.
- Action 3: Reinitialised as a new honeypot and reset to the initial state.

A model of the honeypot system is established based on the MDP. Fig. 1. shows the state transitions of the states (1, 2, and 3) resulted from each of the actions (Action 1, Action 2, and Action 3).

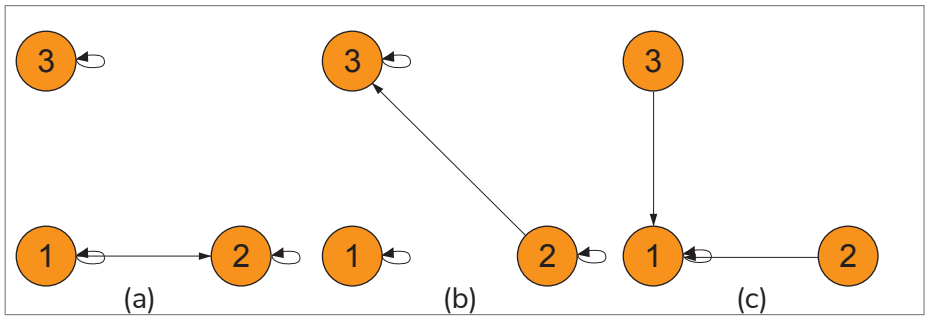

Figure 1. The state transitions due to each of the 3 actions: (a) Action 1, (b) Action 2, and (c) Action 3.

3.2. State Transition Matrix and Reward Matrix

The transitions between the states in the created model of the system rely on one of the actions and on two important probabilities [12]. State 1 cannot be transitioned to State 3 directly; State 3 cannot be transitioned to State 2. The probability of a transition from State 3 to State 1 is 0 (under Action 1 or Action 2) or 1 (under Action 3). The following is a description of the two important probabilities:

- P_a: the probability of attacking the honeypot.
- P_d: the probability of the honeypot being disclosed.

The benefit and expenses due to the state transitions or self-transitions are as follows [12]:

- E_o: the operation expense due to running, deploying, and controlling a honeypot.
- E_r: the expense in reinitializing a honeypot.
- E_l: the expense in liability when a honeypot operator becomes liable for implementing a botmaster's commands if those commands include illicit actions.
- B_i: the benefit of information when a honeypot collects an attacker's information regarding techniques, codes, and tools.

The state transition probability matrix T and the reward matrix R under each action are formulated as follows:

1) T and R under Action 1 are

$$T = \begin{bmatrix} 1-P_a & P_a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{12}$$

$$R = \begin{bmatrix} -E_o & B_i - E_o & 0 \\ 0 & B_i - E_o - E_l & 0 \\ 0 & 0 & -E_o \end{bmatrix} \tag{13}$$

2) T and R under Action 2 are

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1-P_d & P_d \\ 0 & 0 & 1 \end{bmatrix} \tag{14}$$

$$R = \begin{bmatrix} -E_o & 0 & 0 \\ 0 & B_i - E_o & -B_i - E_o \\ 0 & 0 & -E_o \end{bmatrix} \tag{15}$$

3) T and R under Action 3 are

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{16}$$

$$R = \begin{bmatrix} -E_r & 0 & 0 \\ -B_i - E_r & 0 & 0 \\ -E_r & 0 & 0 \end{bmatrix} \tag{17}$$

4. Analyses of the Honeypot System Based on MDP

4.1. MDP-based Analyses over an Infinite Planning Horizon

Let $P_a = 0.6$, $P_d = 0.6$, $E_o = 1$, $E_r = 2.5$, $B_i = 16$, $E_l = 14$, and $\gamma = 0.85$. Analyses are performed using the *R* language and its functions. By substituting the data into equations (12–17), the values of T and R under various actions (due to various policies) can be computed:

T and R under Action 1 become

$$T = \begin{bmatrix} 0.4 & 0.6 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad R = \begin{bmatrix} -1 & 15 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

T and R under Action 2 are

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.4 & 0.6 \\ 0 & 0 & 1 \end{bmatrix}, \qquad R = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 15 & -17 \\ 0 & 0 & -1 \end{bmatrix}$$

T and *R* under Action 3 are

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad R = \begin{bmatrix} -2.5 & 0 & 0 \\ -18.5 & 0 & 0 \\ -2.5 & 0 & 0 \end{bmatrix}$$

Various policies are evaluated, and Tab. 1. shows the result of the total expected rewards for states with various policies. For example, the policy c (1, 1, 3) indicates that Action 1, Action 1, and Action 3 are taken on State 1, State 2, and State 3, respectively. *V*1, *V*2, and *V*3 represent the total expected reward for State 1, State 2, and State 3, respectively.

Table 1. The total expected reward of each state for four various policies ($\gamma = 0.85$).

Policy	c (1, 1, 2)	c (1, 1, 3)	c (1, 2, 3)	c (2, 1, 3)
V1	18.1818	18.1818	13.4431	-6.6667
V2	6.6667	6.6667	0.5342	6.6667
V3	-6.6667	12.9545	8.9266	-8.1667

The four kinds of algorithms (VI, PI, LP, and Q-learning) can be implemented using the values of *T* and *R* under various actions. These algorithms are used in this paper and the optimal policy achieved using the four algorithms is c (1, 1, 3) in each case. The results for the total expected rewards for each state are compared to evaluate the validity of the MDP model in this paper. The results of the honeypot system (based on a discounted MDP with $\gamma = 0.85$) over an infinite planning horizon are shown in Tab. 2.

VI consists of solving Bellman's equation iteratively. Jacob's algorithm and Gauss-Seidel's algorithm are employed in the VI method respectively, so that there are two variants of VI algorithm employed. In Gauss-Seidel's value iterations, $V(k+1)$ is used instead of $V(k)$ whenever this value has been calculated; k is the iteration number. In this situation, the convergence speed is enhanced. It is also shown that its accuracy is improved in comparison to Jacob's algorithm (Tab. 2.). The result of Gauss-Seidel's value iteration algorithm shows that the total expected reward is 18.1818 (the highest value) if the MDP starts in state 1 while it is 6.6667 (the lowest value) if the MDP starts in state 2. The Q-learning result in Tab. 2. was obtained when the number of iterations was 150,000. The results of the VI (Gauss-Seidel's algorithm), PI, and LP are the same, and very close to the Q-learning result, indicating the MDP model created is valid, and that the model parameters are indeed suitable.

Table 2. Analyses of the honeypot system based on various algorithms over an infinite planning horizon ($\gamma = 0.85$).

Algorithm	V1	V2	V3
VI (Jacob's algorithm)	17.9622	6.4470	12.7349
VI (Gauss-Seidel's algorithm)	18.1818	6.6667	12.9545
PI	18.1818	6.6667	12.9545
LP	18.1818	6.6667	12.9545
Q-learning	18.1699	6.6667	12.9206

4.2. The MDP-based Analysis for the Honeypot System over a Finite Planning Horizon

The above data regarding probabilities, the benefit, and expenses (*i.e.*, P_a, P_d, E_o, E_r, B_i, and E_i) are also utilised in the analysis of the system with the discount $\gamma = 0.85$ over a finite planning horizon based on the MDP method. Tab. 3. shows the total expected rewards of the three states that were calculated using value iterations over a 50-step planning horizon. $V1(n)$, $V2(n)$, and $V3(n)$ are the total expected reward at step n for State 1, State 2, and State 3, respectively. It is shown that the total expected rewards $V1(n)$, $V2(n)$, and $V3(n)$ are very close to $V1$, $V2$, and $V3$ for the infinite planning horizon in Tab. 2. when epoch $n \leq 20$.

Table 3. Total expected rewards for three states calculated using value iterations over a 50-step planning horizon ($\gamma = 0.85$).

Epoch n	$V1(n)$	$V2(n)$	$V3(n)$
0	18.1798	6.6647	12.9526
5	18.1774	6.6622	12.9501
10	18.1718	6.6567	12.9445
15	18.1592	6.6441	12.9320
20	18.1309	6.6158	12.9037
25	18.0672	6.5520	12.8399
30	17.9234	6.4083	12.6962
35	17.5995	6.0843	12.3722
40	16.8691	5.3542	11.6415
45	15.1715	3.7086	9.8657
46	14.5479	3.1866	9.0898
47	13.6351	2.5725	7.7289
48	12.0340	1.8500	4.8100
49	8.6	1.0	-1.0
50	0	0	0

5. Analyses of the Honeypot System Based on the POMDP

5.1. Observations and Observation Probabilities in the Honeypot System

The POMDP model of the system is based on the MDP model shown in Fig. 1., and observations as well as observation probabilities are considered to model uncertainty in the POMDP model. Three observations [12] are employed to compute and monitor the system belief state:

- *Unchanged:* The honeypot does not have any observed change, indicating it is still in the waiting state (State 1).
- *Absence:* It means an absence of botmasters' commands after the honeypot was compromised. This situation can be due to 1) the honeypot being detected and disconnected from the botnet, or 2) botmasters being busy with other things (for example, compromising other machines), leading to uncertainty in determining whether the honeypot is in State 2 (compromised) or State 3 (disclosed).
- *Commands:* After the honeypot is compromised, it receives the command information from a botmaster, indicating that it is not disclosed yet and still in State 2.

In State 2, the probability of receiving commands is denoted by P_{o1}, while the probability of absence is denoted by P_{o2}. Therefore, we have the following observation probabilities:

For the honeypot in State 1: $P(Unchanged) = 1$, $P(Commands) = P(Absence) = 0$
For the honeypot in State 2: $P(Unchanged) = 0$, $P(Commands) = P_{o1}$
$P(Absence) = P_{o2} = 1 - P_{o1}$
For the honeypot in State 3: $P(Unchanged) = P(Commands) = 0$, $P(Absence) = 1$

5.2. Analyses Based on Various Solution Methods of the POMDP over An Infinite Planning Horizon

Analyses over an infinite planning horizon for a discounted POMDP of the honeypot system are performed. Let $P_a = 0.6$, $P_d = 0.6$, $E_o = 1$, $E_r = 2.5$, $B_i = 16$, $E_l = 14$, and $\gamma = 0.85$. The following solution methods or algorithms [23, 24, 26–29, 30] are used to solve the POMDP problem: Grid, Enumeration, Two Pass, Witness, Incremental Pruning, and SARSOP. The total expected reward of the honeypot system based on POMDP is denoted by V_t in this paper. The values of V_t at three different observation probabilities of receiving commands ($P_{o1} = 0.5$, 0.6, and 0.7) are computed using various solution methods of POMDP. The result of V_t is shown in Tab. 4. The values of Incremental Pruning and SARSOP are very close to the results of the other four methods and the results of the four methods are the same.

Table 4. The total expected reward of the honeypot system based on various solution methods of POMDP.

Methods	$V_t(P_{o1} = 0.5)$	$V_t(P_{o1} = 0.6)$	$V_t(P_{o1} = 0.7)$
Grid	9.850447	10.187263	10.449232
Enumeration	9.850447	10.187263	10.449232
Two Pass	9.850447	10.187263	10.449232
Witness	9.850447	10.187263	10.449232
Incremental Pruning	9.848475	10.185292	10.447260
SARSOP	9.850403	10.187213	10.449210

5.3. The Analysis for the Honeypot System with Various Observation Probabilities of Receiving Commands

The total expected reward V_t of the honeypot system with various observation probabilities of receiving commands (P_{o1}) is analysed for the discounted POMDP over an infinite planning horizon. Grid is used to solve the POMDP problem. It tries to approximate the value function over an entire state space according to the estimation for a finite number of belief states on the chosen grid [31]. The following data are used in the analysis: $P_a = 0.6$, $P_d = 0.6$, $E_o = 1$, $E_r = 2.5$, $B_i = 16$, $E_l = 14$, and $\gamma = 0.85$; $P_{o1} = 0.1$, 0.2, 0.3, ..., 0.9. Fig. 2. shows that the total expected reward V_t of the honeypot system increases as the observation probability (P_{o1}) of receiving commands rises. In the following sections of this paper, the Grid method is also used in solving the POMDP problem.

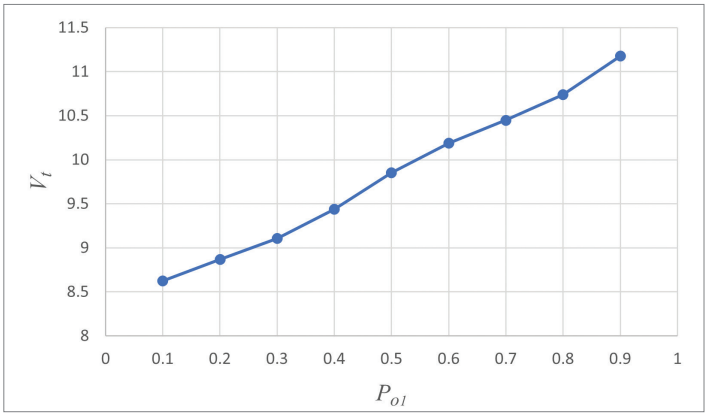

Figure 2. The total expected reward V_t of the honeypot system at various P_{o1}.

5.4. Analyses for the system with various P_a and P_d

An analysis for the discounted POMDP with various P_a over an infinite planning horizon is conducted. The following data are utilised: P_d=0.6, E_o = 1, E_r = 2.5, B_i = 16, E_l = 14, and γ = 0.85. The total expected reward V_t of the honeypot system at various P_a for various P_{o1} is analysed and the result is shown in Fig. 3. V_t increases with higher values of P_a, although the rate of increase steadily diminishes. The increased P_a provides the honeypot with more opportunities for collecting valuable information about attackers. V_t is larger when P_{o1} is larger.

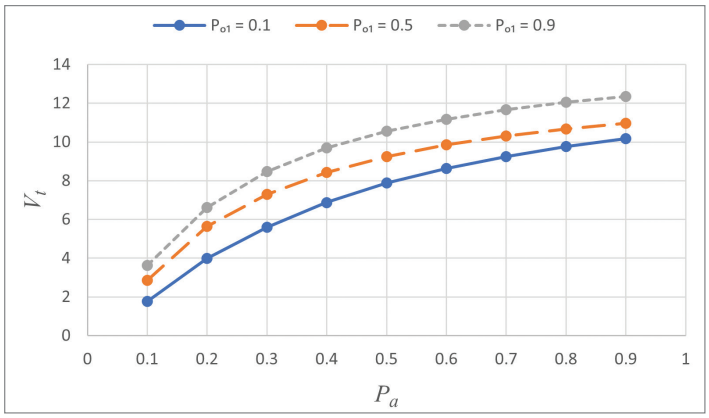

Figure 3. The total expected reward V_t of the honeypot system at various P_a.

Let P_a = 0.6, E_o = 1, E_r = 2.5, B_i = 16, E_l = 14, and γ = 0.85. The V_t at various P_d for various P_{o1} is analysed over an infinite planning horizon, and Fig. 4. shows the results. V_t is higher when P_{o1} is higher, but the value of V_t when P_{o1} = 0.1 is very close to that of V_t when P_{o1} = 0.5 (if P_d < 0.5). For P_{o1} = 0.1, V_t falls as P_d is increased from 0.1 to 0.8 and is unchanged when P_d moves from 0.8 to 0.9; for P_{o1}= 0.5, V_t decreases as P_d is increased from 0.1 to 0.6 and is unchanged as P_d goes from 0.6 to 0.9; for P_{o1} = 0.9, V_t declines as P_d is increased from 0.1 to 0.5, though it does not change as P_d moves from 0.5 to 0.9. There is no significant difference in V_t for P_{o1} = 0.5 and P_{o1} = 0.9 when P_d changes from 0.5 to 0.9.

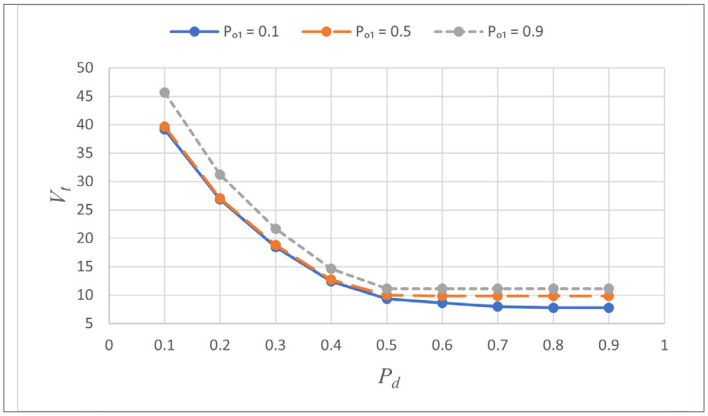

Figure 4. The total expected reward V_t of the honeypot system at various P_d.

5.5. Analyses for the System with Various Transition Rewards

Analyses for the honeypot system with various transition rewards over an infinite planning horizon are performed. The following data are utilised: $P_a = 0.6$, $P_d = 0.6$, $E_o = 1$, $E_r = 2.5$, $E_l = 14$, and $\gamma = 0.85$. The total expected reward V_t at various P_a for various P_{o1} is analysed, and the results are shown in Fig. 5. V_t initially increases slightly ($B_i < 14$) and then more rapidly ($B_i > 14$) with the increase of B_i. V_t for various P_{o1}(0.1, 0.5, and 0.9) is the same when B_i= 10, 11, and 12. V_t is the same for $P_{o1} = 0.1$ and 0.5 when $B_i = 13$. When $B_i > 13$, V_t is larger if P_{o1} is larger.

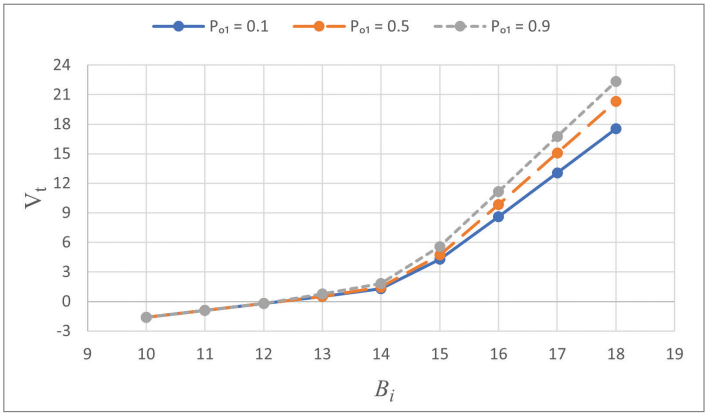

Figure 5. The total expected reward V_t of the honeypot system V_t at various B_i.

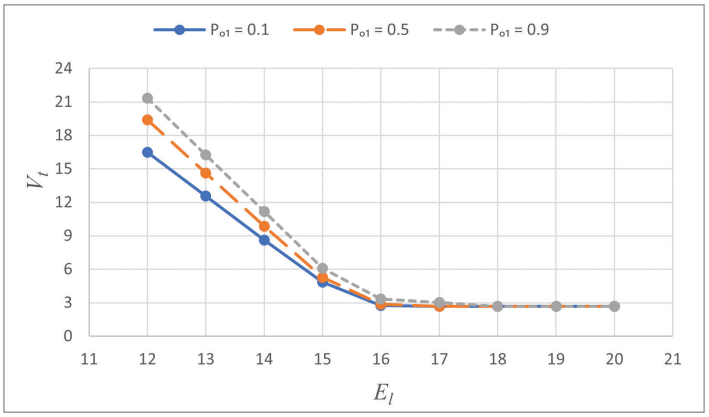

Figure 6. The total expected reward V_t of the honeypot system at various E_l.

Let P_a = 0.6, P_d = 0.6, E_o = 1, E_r = 2.5, B_i = 16, and γ = 0.85. The total expected reward V_t at various E_l for various P_{o1} is analysed over an infinite planning horizon and Fig. 6. shows the results. V_t decreases when E_l is increased from 12 to 16. V_t is the same for P_{o1} = 0.1 and 0.5 as E_l rises from 17 to 20. It is the same for all the three values of P_{o1} (0.1, 0.5, and 0.9) when E_l goes from 19 to 20.

6. Conclusion

The MDP-based predictive modelling for the honeypot system has demonstrated that the model and algorithms in this paper are suitable for performing analyses over both a finite planning horizon and an infinite planning horizon (for a discounted MDP), and that they are effective at finding an optimal policy and maximizing the total expected rewards of the states of the honeypot system. The results of the total expected reward using Gauss-Seidel's algorithm of VI, PI, and LP are the same, and the result of Q-learning is very close to the same result, indicating the MDP model created in this paper is valid and that the model parameters are suitable.

In the predictive modelling of the honeypot system based on the discounted POMDP over an infinite planning horizon, the total expected reward V_t of the honeypot system increases with the increase of the observation probability of receiving commands (P_{o1}). It also rises as P_a is increased or B_i is increased. The increased P_a leads to more opportunities for the honeypot to collect valuable information about attackers. As P_d increases, V_t declines at first and then levels out. As E_l increases, V_t decreases by successively smaller amounts until it eventually flattens out.

Declaration of Competing Interest

The authors in this paper do not have any competing interest.

Data availability

No database or dataset was used or generated in the research of this article.

Acknowledgments

This paper is based upon work supported by Mississippi State University (MSU) in the USA.

REFERENCES

[1] T.M. McKenzie, *Is Cyber Deterrence Possible?. Alabama: Air University Press*, 2017.

[2] S. Srujana, P. Sreeja, G. Swetha, H. Shanmugasundaram, "Cutting edge technologies for improved cybersecurity model: A survey," *International Conference on Applied Artificial Intelligence and Computing* (ICAAIC), 2022, pp. 1392–1396.

[3] L. Vokorokos, A. Pekár, N. Ádám, P. Darányi, "Yet another attempt in user authentication," *Acta Polytechnica Hungarica*, vol. 10, no. 3, pp. 37–50, 2013.

[4] J. Palša, J. Hurtuk, E. Chovancová, M. Havira, "Configuration honeypots with an emphasis on logging of the attacks and redundancy," *IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics* (SAMI), 2022, pp. 000073–000076, doi: 10.1109/SAMI54271.2022.9780801.

[5] F. Franzen, L. Steger, J. Zirngibl, P. Sattler, "Looking for honey once again: Detecting RDP and SMB honeypots on the Internet," *IEEE Looking European Symposium on Security and Privacy Workshops* (EuroS&PW), 2022.

[6] M. Boffa, G. Milan, L. Vassio, I. Drago, M. Mellia, Z.B. Houidi, "Towards NLP-based processing of honeypot logs," *IEEE European Symposium on Security and Privacy Workshops* (EuroS&PW), 2022, pp. 314–321.

[7] Z. Shamsi, D. Zhang, D. Kyoung, A. Liu, (2022, June). "Measuring and clustering network attackers using medium-interaction honeypots," *IEEE European Symposium on Security and Privacy Workshops* (EuroS&PW), 2022, pp. 294–306.

[8] X. Liu, H. Zhang, S. Dong, Y. Zhang, "Network defense decision-making based on a stochastic game system and a deep recurrent Q-network," *Computers & Security*, vol. 111, p. 102480, 2021, doi: 10.1016/j.cose.2021.102480.

[9] H. Itoh, H. Nakano, R. Tokushima, H. Fukumoto, H. Wakuya, "A partially observable markov decision process-based blackboard architecture for cognitive agents in partially observable environments," *IEEE Transactions on Cognitive and Developmental Systems*, 2020, doi: 10.1109/TCDS.2020.3034428.

[10] M. Haklidir, H. Temeltaş, "Guided soft actor critic: a guided deep reinforcement learning approach for partially observable markov decision processes," *IEEE Access*, vol. 9, pp. 159672–159683, 2021, doi: 10.1109/ACCESS.2021.3131772.

[11] A.R. Cassandra. (2003). A survey of POMDP applications. [Online].
Available: http://www.cassandra.org/arc/papers/applications.pdf. [Accessed: Nov. 27, 2022].

[12] O. Hayatle, H. Otrok, A. Youssef, "A Markov decision process model for high interaction honeypots," *Information Security Journal: A Global Perspective*, vol. 22, no. 4, pp. 159–170, 2013.

[13] M. Mohri, A. Rostamrdeh, A. Talwalkar, *Foundations of machine learning. Adaptive computation and machine learning*. Cambridge, Massachusetts: MIT Press, 2012.

[14] M.A. Alsheikh, D.T. Hoang, D. Niyato, H.P. Tan, S. Lin, "Markov decision processes with applications in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, 1239–1267, 2015, doi: 10.1109/COMST.2015.2420686.

[15] Y. Chen, J. Hong, C.C. Liu, "Modeling of intrusion and defense for assessment of cyber security at power substations," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2541–2552, 2018.

[16] R.S. Sutton, A.G. Barto, *Reinforcement learning: An introduction*. Cambridge, Massachusetts: MIT press, 2018.

[17] M. van Otterlo, "Markov decision processes: Concepts and algorithms," in *Reinforcement Learning. Adaptation, Learning, and Optimization*, vol 12, M. Wiering, M. van Otterlo, M., Eds, Berlin, Heidelberg: Springer. doi: 10.1007/978-3-642-27645-3_1.

[18] M. van Otterlo, M. Wiering, "Reinforcement learning and Markov decision processes," *in Reinforcement Learning. Adaptation, Learning*, and Optimization, vol 12, M. Wiering, M. van Otterlo, M., Eds, Berlin, Heidelberg: Springer, pp 3-42. doi: 10.1007/978-3-642-27645-3_1.

[19] O. Sigaud, O. Buffet, *Markov decision processes in artificial intelligence*. Hoboken, New Jersey: John Wiley & Sons, 2013.

[20] S.J. Majeed, M. Hutter, "On Q-learning convergence for non-Markov decision processes," IJCAI, pp. 2546–2552, 2018, doi: 10.24963/ijcai.2018/353.

[21] E. Zanini, *Markov decision processes*. Berlin, Heidelberg: Springer, 2014.

[22] Y. Liu, H. Liu, B. Wang, "Autonomous exploration for mobile robot using Q-learning," *2nd International Conference on Advanced Robotics and Mechatronics* (ICARM), 2017, pp. 614–619, doi: 10.1109/ICARM.2017.8273233.

[23] G.E. Monahan, "State of the art—a survey of partially observable Markov decision processes: theory, models, and algorithms," *Management science*, vol. 28, no. 1, pp. 1–16, 1982, doi: 10.1287/mnsc.28.1.1.

[24] M.L. Littman, A.R. Cassandra, L.P. Kaelbling, *Efficient dynamic-programming updates in partially observable Markov decision processes*, Department of Computer Science. Providence, Rhode Island: Brown University, CS-95-19, 1995.

[25] J. Stuart, P. Norvig, *Artificial intelligence: A modern approach*. London: Pearson, 3rd ed., 2010.

[26] H. Kurniawati, D. Hsu, W.S. Lee, "Sarsop: Efficient point-based POMDP planning by approximating optimally reachable belief spaces," *Robotics: Science and Systems, 2008*.

[27] J. Pineau, G. Gordon, S. Thrun, "Point-based value iteration: An anytime algorithm for POMDPs," *IJCAI*, vol. 3, pp. 1025–1032, 2003.

[28] E.J. Sondik, *The optimal control of partially observable Markov processes*. Stanford, California: Stanford University, 1971.

[29] N.L. Zhang, W. Liu, *Planning in stochastic domains: Problem characteristics and approximation*, Department of Computer Science. Hong Kong: Hong Kong University of Science and Technology, HKUST-CS96-31, 1996.

[30] A.R. Cassandra, M.L. Littman, N.L. Zhang, "Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes," *arXiv preprint arXiv*:1302.1525, 2013.

[31] R.I. Brafman, "A heuristic variable grid solution method for POMDPs," *AAAI/IAAI*, 1997, pp. 727–733.