International Journal of Information Security (2022) 21:637–651
https://doi.org/10.1007/s10207-021-00573-4

REGULAR CONTRIBUTION

Maintaining the level of operational effectiveness of a CSOC under adverse conditions

Ankit Shah[1] · Rajesh Ganesan[2] · Sushil Jajodia[2] · Hasan Cam[3]

Published online: 22 November 2021

Abstract
The level of operational effectiveness (LOE) is a color-coded performance metric that is monitored by the cybersecurity operations center (CSOC). It is determined using the average time to analyze alerts (AvgTTA) in every hour of shift operation, where the time to analyze an alert (TTA) is the sum of waiting time in the queue and investigation time by the analysts. Ideally, the CSOC managers would set a predetermined baseline target for AvgTTA to be maintained for every hour of shift operation. However, due to adverse events, an imbalance may exist if the alert arrival rate far exceeds the service rate, resulting in high AvgTTA or low LOE. Upon exhausting all the analyst resources, the only option available to a CSOC manager is to discard alerts for restoring the LOE of the CSOC. The paper proposes two strategies: the *value-based* strategy is developed using a static optimization model while the *reinforcement learning-based* strategy is developed using a dynamic optimization model. The paper compares various strategies for discarding alerts and measures the following desiderata for comparing them: (1) minimize the number of alerts discarded, (2) ensure highest utilization of analysts, (3) determine the optimal time at which the alerts must be discarded in a shift, and (4) maintain the best possible LOE closest to the baseline target LOE. Results indicate that, overall, the RL strategy is the best performer among all strategies that guarantees the AvgTTA below the threshold value in every hour of shift operation while discarding the fewest number of alerts under adverse events.

Keywords Reinforcement learning · Discarding alerts · Level of operational effectiveness · Alert analysis · Cybersecurity operations center

1 Introduction

In recent years, the alert analysis process at a cyber security operations center (CSOC) has been thoroughly investigated by the authors of this paper. For example, the scheduling of analysts at the CSOC is presented in Ganesan et al. [10], [11]. Optimal clustering of sensors and their allocation to analysts is studied in [23]. The fair allocation of CSOC resources to clients is presented in [19]. In [20], an adaptive reallocation model to balance workload among analysts is presented. Understanding tradeoffs between throughput, quality, and cost of alert analysis in a CSOC is published in [24]. More recently, our research focused on the metric to measure the level of operational effectiveness (LOE) of the CSOC, which is presented in [22]. We have used this metric to make decisions such as when and how many on-call analysts must be added to regular analyst staff in order to maintain the LOE of a CSOC [21]. One research case study that has not been investigated in [21] is about maintaining the LOE status when the CSOC is overwhelmed by a large number of alerts that far exceeds the alert analysis capacity, and every possible analyst resource and CSOC manager's actions has already been exhausted (even all on-call analysts are used). In this situation, the LOE becomes extremely low and is said to be in the red zone as explained below. No alternative exists

✉ Sushil Jajodia
jajodia@gmu.edu

Ankit Shah
ankitshah@usf.edu

Rajesh Ganesan
rganesan@gmu.edu

Hasan Cam
hasan.cam.civ@mail.mil

[1] University of South Florida, Mail Stop ENG-030, Tampa, FL 33620, USA

[2] Center for Secure Information Systems, George Mason University, Mail Stop 5B5, Fairfax, VA 22030-4422, USA

[3] Army Research Laboratory, 2800 Powder Mill Road, Adelphi, Maryland 20783-1138, USA

in such conditions for the CSOC manager except to discard alerts and return the CSOC to its normal operating condition. This paper is focused on developing a reinforcement learning-based intelligent strategy to assist the CSOC manager in discarding alerts, and comparing it with three other discarding strategies. In what follows, a detailed background and motivation for the research is presented.

Under normal operating conditions, a CSOC's analyst staffing level (defines the service rate) is designed for a certain level of alert arrival rate. However, alerts may arrive at a much higher rate than the service rate at the CSOC due to several adverse events. Such adverse events include an external attack, lack of sufficient staffing due to absenteeism, or other network events such as a temporary breakdown in communication link between intrusion detection systems (IDSs) and the CSOC. The consequence of high alert generation rate above the service rate leads to a buildup in alert backlog, which in turn reduces the LOE of the CSOC. The LOE of a CSOC is a color-coded representation of status of the CSOC as shown in Fig. 1 [22]. The LOE status is derived using the average time to analyze alerts in an hour of operation (AvgTTA), which has both a baseline value and a threshold value as shown in the figure.

There are a few action options that a CSOC manager can deploy when LOE deteriorates. Typically, analysts spend a portion of the time investigating alerts, and the remainder on activities such as report writing, training, and updating signatures for the IDSs [2,7,9,21]. In one of the options, a CSOC manager can divert some of the above remainder time toward alert investigation. Another more expensive option is to employ on-call analysts who can support the regular CSOC analyst staff in alert investigation. However, on-call analysts are a limited resource. Shah et al. [21] provide a methodology to deploy the above on-call analysts over a 14-day work cycle (schedule) under the uncertainties of adverse events. The paper describes when and how many on-call analysts to call depending on current backlog of alerts, time left in the 14-day work cycle, and the amount of on-call resources that are still available in the 14-day work cycle. The method has been proven to be effective when compared to other greedy and rule-based options in maintaining LOE status if and only if the number of additional alerts generated due to adverse events is bounded and sufficient on-call analysts are available. However, none of the above prior work can deal with alert backlog due to adverse events as soon as the CSOC runs out of the limited additional resources and all of CSOC manager's action options have been exhausted. The consequence is that the LOE will deteriorate and reach into the red zone as shown in Fig. 1. Once in the red zone, the LOE will continue to remain in the zone and further deteriorate until the start of the next 14-day work cycle, when additional resources may be made available. The above cascading effect of transferring alert backlogs from one shift to another, and between 14-day work cycles can severely impact the overall performance of the CSOC in the long run because the LOE could have reached the red zone several times and stayed in there for several hours of CSOC operation. This would mean that the alerts would have waited for a long duration in the investigation queue, and potential malicious activities would have continued to persist in an organization's network, thereby affecting its security posture.

The current option that is exercised by CSOC managers to maintain LOE status and prevent being in the red zone is to discard alerts periodically.[1] With respect to which alerts to discard, they use different strategies: They select alerts that have waited the longest for investigation. If the alerts are prioritized based on their criticality level, they can select alerts that are not only oldest but have the lowest criticality level also. Unfortunately, this is done in a reactive and ad hoc manner. We assert that it is non-trivial to decide when, how many, and which alerts to discard because there exists a trade-off between the timing and number of alerts discarded and the LOE obtained. Discarding too many alerts would result in idling of analysts and the lack of analysis of those discarded alerts. Similarly, discarding too few would result in high alert backlogs that in time causes LOE deterioration. This paper considers 12-hour shifts over a 14-day work cycle; the decision to discard alerts is made in every hour of shift operation. The number of alerts we discard depends on LOE; our goal is to keep LOE as close to the baseline value as possible even under the uncertainty of future adverse events.

The above motivates the need for an intelligent strategy for discarding alerts when additional resources have been exhausted in the shift. The paper assumes that at the beginning of each shift the backlog is observable; the work schedule of regular analysts is static and is known prior to the beginning of the 14-day work cycle. As explained earlier, when analyst resources are exhausted in the shift, the high alert backlog due to adverse events will likely cause the LOE to reach and remain in the red zone. At this point, intelligently discarding alerts is invoked, and it becomes an important option for CSOC managers that is worthy of further research. It should be noted that discarding alerts is an action option that is independent of the amount of analyst resources available in a shift. Instead, it is driven by the baseline and threshold values of AvgTTA from which LOE status is ascertained.

The paper has the following desiderata. 1) minimize the number of alerts discarded, 2) ensure highest utilization of analysts, 3) determine the optimal time at which the alerts must be discarded in a shift, and 4) maintain the best possible LOE closest to the baseline target LOE. The objective of the paper is to compare four strategies for discarding

[1] This observation is based on our numerous conversations with cybersecurity analysts and the CSOC managers.

alerts, including the design and evaluation of an intelligent strategy. These strategies include 1) threshold-based 2) analyst resource utilization-based, 3) value-based (static optimization), and 4) reinforcement learning-based (dynamic optimization) strategies. The first two strategies (threshold-based and analyst resource utilization-based) are currently being used by CSOC managers to bring the CSOC to normal operating conditions. The threshold-based strategy discards alerts when a certain threshold value is reached for the AvgTTA. The utilization-based strategy ensures that analysts are fully utilized but does not consider maintaining the AvgTTA below the threshold. In this work, we propose two strategies (value-based and reinforcement learning-based) using optimization techniques. While the former strategy is developed using a static optimization model, the latter strategy is developed using a dynamic optimization model. The value-based strategy is myopic and this static strategy does not consider future uncertainties while making decisions to discard alerts. Attaining all of the aforementioned desiderata requires sequential decision-making that takes into consideration future uncertainties (adverse events) that may occur at a CSOC. This uncertainty changes the state of the CSOC environment, where the future dynamics of the system state depend on the current state. The randomly evolving system state and alert discarding decision-making can be formulated as a Markov decision process (MDP). Reinforcement learning (RL) is a solution method for this MDP. In an RL approach, an autonomous agent prescribes (near-) optimal actions by learning through experience over time. A reward signal guides the agent in attaining its objective over the long run. Results indicate that the RL strategy discards the lowest number of alerts and determines the right timing to discard alerts among all strategies when simulated with any number of uncertain events during the 14-day work cycle.

The contributions of this research are as follows. It presents two optimization models (value-based and RL-based) to solve the problem of discarding alerts and compares them with the ad hoc rule-based strategies utilized by the CSOCs. The reward function in the RL-based strategy takes into account multiple conflicting objectives of maintaining a low AvgTTA value and a long alert queue length. Moreover, this reward function is individualized to allow an organization to assign preferences among the objectives. This decision-making framework has the capability to incorporate future uncertainties into its decision-making, therefore, resulting in the minimum number of alerts discarded. It also maintains the LOE below the threshold set by the CSOC for every shift operation. Other contributions include several meta-principles that provide insights into alert backlog formation and its impact on LOE, and decision-making to discard alerts such that the desiderata given above are maintained.

The rest of the paper is organized as follows. Section 2 describes literature related to this work. In Sect. 3, the four strategies investigated in this research are presented along with their algorithms. Section 4 describes the experimental setup comparing the above strategies. In Sect. 5, the results and analysis of experiments are presented along with computational complexity of the reinforcement learning-based strategy. Meta-principles derived from the study are presented in the same section. Finally, in Sect. 6, conclusions of the research are presented.

2 Related literature

Organizations rely on a unique combination of security personnel and technology to protect against cyber threats. Such an entity with defined goals, responsibilities, and processes is identified by various names: a cybersecurity operations center (CSOC or SOC), a computer security incident response team (CSIRT), network operations center (NOC), network and security operations center (NSOC), or a managed security service provider (MSSP). Optimal management of CSOC resources have been an active topic of interest by many researchers ([13], [10], [6], [26]), and more recently by [22]. D'Amico and Whitley [6] studied the analytical process that transforms data into security situation awareness by conducting a cognitive task analysis to baseline the state of the practice in the Department of Defense CSOCs. Sundaramurthy et al. [27] studied the burnout of analysts using an anthropological approach and proposed methods to improve the CSOCs.

Alert analysis process at a CSOC is shown in Fig. 2 [10]. Sensors are clustered together based on types, or to uniformly balance the expected alert workload among them and are then assigned to the analysts [23]. The sensor data are processed through automated filters such as the intrusion detection systems (IDSs) [18] or Secure Information and Event Management (SIEM) systems [3], which use techniques such as pattern recognition ([5], [2]) or anomaly detection [17] to identify suspicious activities. These suspicious activities, in the form of alerts, are then picked up by the analysts for further investigation. The initial or first level of investigation is a fast decision-making process [6], where the analysts distinguish among the innocuous and significant alerts. The number of alerts generated by the IDSs is far more than the number of alerts that could be investigated by the available analysts at a CSOC. Recently, Ganesan et al. [10] have investigated optimal scheduling of analysts to maximize alert investigations using genetic programming. Furthermore, Ganesan et al. [11] proposed a dynamic model using an agent to schedule on-call analysts to assist with additional alert investigations during adverse events. The significant alerts identified by the analysts during the initial analysis are further investigated as a part of the secondary level of investigation, which takes hours to days to complete

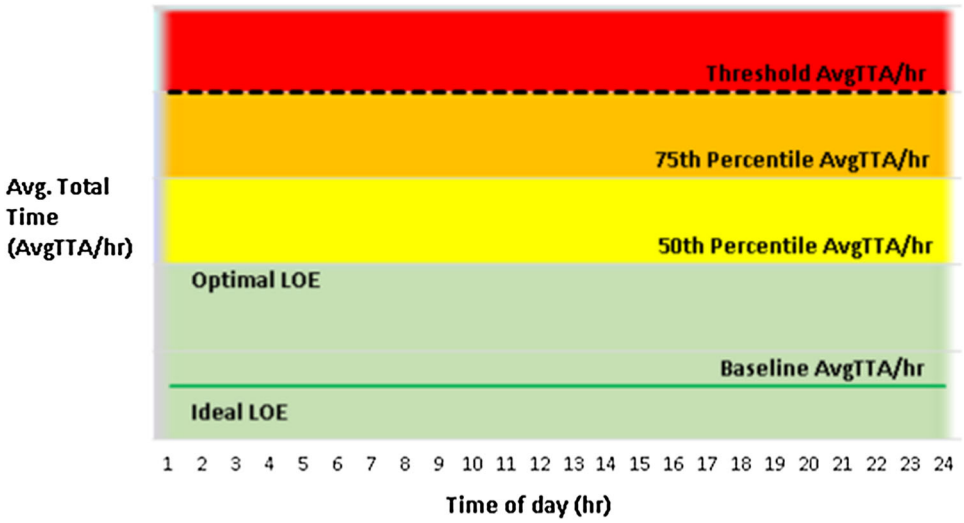

Fig. 1 Color-coded level of operational effectiveness status [22]

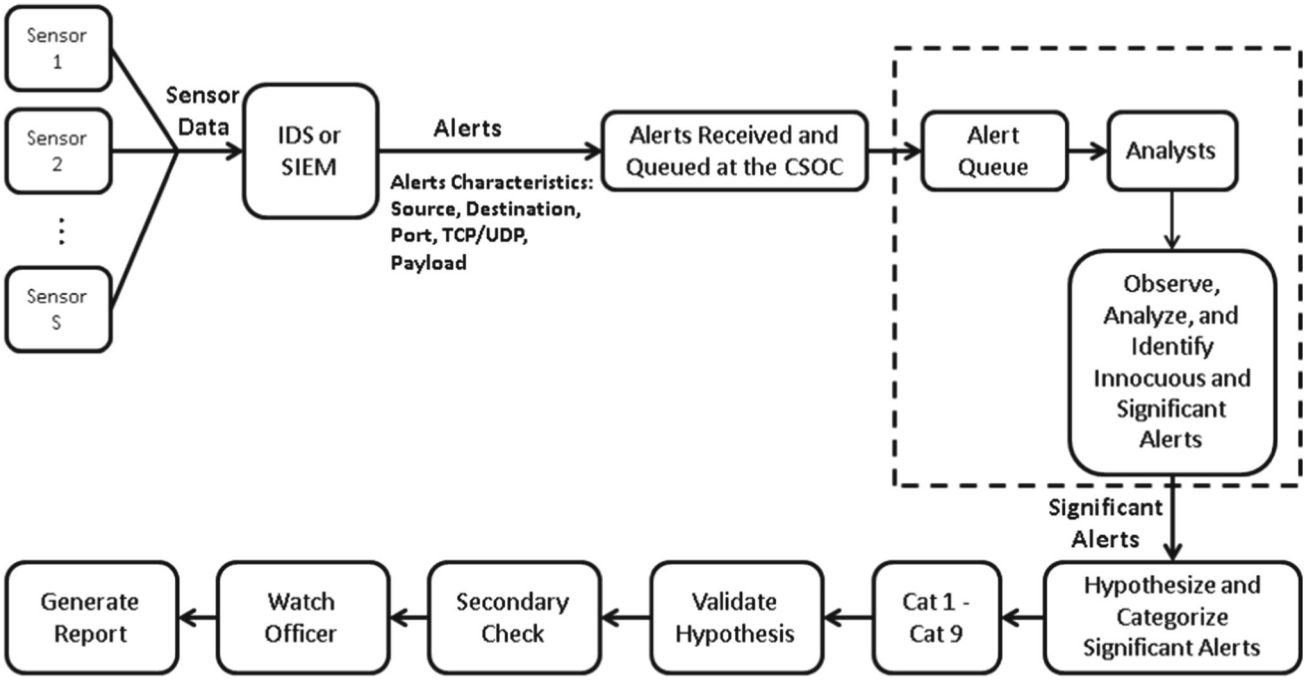

Fig. 2 Alert analysis process [10]

investigation. Reports are written for the significant alerts, which are identified as incidents [4].

Delays in initial alert investigation is a major security concern for an organization as a malicious activity will remain in the network for longer, while waiting for an analyst to identify and mitigate it. Shah et al. [22] quantified the delays in alert investigations and modeled the alert analysis process at a CSOC as a M/D/c/FCFS queueing model. A queueing metric that measures the average time an alert spends in the CSOC system (AvgTTA) is used to indicate the performance (LOE) of the CSOC. Furthermore, Shah et al. [21] proposed a method to optimize this queueing metric (AvgTTA) by allocating additional resources at a CSOC. However, with a limited number of additional analyst resource available at a CSOC during any given time period, a strategy that relies only on the analyst resource will not be able to optimize the performance of the CSOC in the event where the number of alert investigations required is far more than the number of alert investigations that could be performed. Integer programming formulations are used to model CSOC processes to take optimal actions in [9] and [24], whereas a stochastic dynamic programming framework is used to model sequential decision-making processes under uncertainty in [11] and [21].

Reinforcement learning is a method for solving Markov decision processes. It is a computational approach to learning optimal actions from interactions with an environment [28]. A reinforcement learning agent is guided by the reward signal generated by the environment in response to the agent's action in a given state. The objective of the reinforcement learning agent is to maximize the total reward over the long run. Reinforcement learning approaches have produced (near-) optimal policies for stochastic lot scheduling [15] and job-shop scheduling [1] problems and have attained human-level control in computer games [14].

To the best of the authors' knowledge, there exists no literature on taking corrective actions by optimally discarding alerts to improve the effectiveness of alert management at a CSOC using a reinforcement learning approach.

3 Strategies for discarding alerts

The paper investigates four strategies for discarding alerts. We describe them next.

3.1 Threshold-based strategy

In the threshold-based strategy, a corrective action of discarding alerts is taken only when the AvgTTA crosses the threshold value, which is pre-determined by the CSOC. Once the AvgTTA crosses into the red (unacceptable) zone, enough number of alerts are discarded to bring the AvgTTA below the threshold. It is to be noted that this is a myopic decision, which is reactive and does not take into consideration any variance in the AvgTTA value due to the underlying stochastic queueing process and future adverse events.

3.2 Utilization-based strategy

The (analyst resource) utilization-based strategy takes into account the utilization of the analyst resource for the entire duration of the work-shift, i.e., during a work-shift, alerts are accumulated until the capacity for alert investigation is reached. This strategy does not take into consideration the AvgTTA value during the work-shift and rather focuses on maintaining the queue length as long as possible such that the resources are fully utilized for the duration of the work-shift. It is to be noted that at any given time t during a work-shift, the number of alerts that could be investigated can be calculated in a deterministic manner. Any additional alert on top of this number at time t, which is added to this waiting-for-investigation queue will result in removing the longest waiting alert from this queue. This strategy ensures maximum analyst utilization for the work-shift.

3.3 Value-based strategy: static optimization model

A value-based static optimization model observes the AvgTTA and the alert queue length waiting for investigation at a CSOC at any given time t and selects an optimal number of alerts to discard. The corresponding backlog numbers for the baseline and the threshold AvgTTA values, which are set by the CSOC, are determined using simulation. There are two functions, f_a and f_q, which are used to obtain the normalized values of the AvgTTA and the alert queue length, respectively. These functions are explained as follows. The parameter a_t is given a value of 0, if the backlog number corresponding to the AvgTTA is at or above the threshold backlog number, and a value of 1, if the backlog number corresponding to the AvgTTA is at or below the baseline backlog number. All other numerical values for the backlog number are linearly normalized between $0 \leq a_t \leq 1$. This normalization function is called f_a. The parameter q_t is given a value of 1, if the alert queue length at time t is greater than or equal to the total number of alerts that could be investigated by the total number of analysts between time t and $t+1$, and a value of 0 if the alert queue length is 0. All other numerical values for the queue length number are exponentially normalized between $0 \leq q_t \leq 1$. This normalization function is called f_q. The goal is to keep the analyst utilization to the maximum for the time between t and $t+1$.

The objective of the value-based static optimization model is to maximize the cumulative score of the above two normalized values by selecting an optimal number of alerts to discard at each time-stamp. It is to be noted that this optimization model is myopic and does not take future uncertainties into consideration while making the decision to discard alerts. The model parameters and mathematical formulation are described next.

3.3.1 Index

- t is the time index, $1 \leq t \leq T$.

3.3.2 Inputs

- b_t is the number of alerts backlogged at time t.
- f_a is the function that returns the normalized value of the AvgTTA.
- f_q is the function that returns the normalized value of the alert queue length.

3.3.3 Decision variable

- x_t is an integer variable indicating the number of alerts to discard at time t.

3.3.4 Mathematical model

The objective of the static model is to maximize the cumulative score of the normalized values for the AvgTTA and the alert queue length, $\forall\, t$.

$$z = Max \;\; (a_t + q_t) \tag{1}$$

Subject to the following constraints:

$$a_t = f_a(b_t - x_t), and \tag{2}$$
$$q_t = f_q(b_t - x_t) \tag{3}$$

It is to be noted that a_t and q_t are calculated using functions f_a and f_q, respectively.

3.3.5 Algorithm for the value-based static optimization model

The algorithm for the value-based static optimization model is presented in Algorithm 1.

Algorithm 1: Value-Based Static Optimization Algorithm

Input: Number of alerts backlogged b_t, function that returns the normalized value of the AvgTTA f_a, and function that returns the normalized value of the alert queue length f_q
Output: Total number of alerts to discard x_t.
Step 1: Initiate the solution search for the objective of maximizing the cumulative score of normalized values for the AvgTTA and the alert queue length in the solver $Max \;\; (a_t + q_t)$
Step 2a: Verify the following constraint within the solver
for *each value of* x_t, **do**
  check the constraints (Equations 2 and 3);
end
Step 2b: Perform optimality check within solver
if *an optimal solution is found;*
then
  stop the solution search in the solver;
end
return *Total number of alerts to discard* x_t.

Next, the dynamic optimization model is explained.

3.4 Reinforcement learning-based strategy: dynamic optimization model

The autonomous decision support tool for taking corrective actions under adverse conditions to maintain the LOE of a CSOC is built using the principles of stochastic dynamic programming (SDP) and solved using reinforcement learning (RL). Figure 3 shows the dynamic optimization model framework. We first present the simulation model, and then explain the key elements of the SDP formulation and the RL algorithm.

3.4.1 Simulation model

As shown in Fig. 3, the simulation model consists of four components: (i) the CSOC system inputs, which include system parameters and alert generation by the IDS, (ii) the adverse events that affect the CSOC, (iii) the alert analysis process, in which the work shift is simulated, and (iv) the performance metric (AvgTTA) that gets monitored every hour of shift operation. The alert analysis process and the adverse events are simulated using the algorithm in [22]. The number of sensors, alert arrival rate, and alert service rate are input parameters for the simulator. The arrivals of the adverse events are modeled using a Poisson probability distribution [8,25]. Analysts are considered as resources, and they investigate the IDS alerts in a first-come, first-served (FCFS) manner. The time taken to investigate an alert by an analyst is the average time taken based on historical statistics observed in the CSOC. This time could be maintained constant unless a new alert pattern causes it to change or drawn from a probabilistic distribution for each alert in the simulated environment.

In a system implementation, the adverse events component will be replaced by the historical real-world data with uncertain events observed in the CSOC, along with the timing of their occurrences and their respective intensities. The RL agent will then interact with the environment without knowing the underlying probability distribution to learn actions that produce the best results. However, if there is raw data available to obtain the distribution of the arrival process, then the CSOC can use this information within the simulator to continue training the RL agent offline and improve its decision-making in real time.

3.4.2 Stochastic dynamic programming formulation

In the context of the research problem, the elements of the SDP formulation are explained below.

- *System State*: $s_t \in S$ is a tuple $s = \langle a_t, q_t, t \rangle$, where a_t is the normalized value of the AvgTTA and q_t is the normalized value of the alert queue length at the beginning of time t. The normalized values of the aforementioned state variables are calculated as described in the prior section.
- *Action (Decision)*: The corrective action is to discard x_t number of alerts at the beginning of time t.
- *Uncertainty*: The uncertain events (see examples in Table 2), which affect the alert analysis process are captured as exogenous information W_{t+1}. The impact of these adverse events is an increase in the alert generation rate or a decrease in the alert service rate, which further increases the alert queue length between time t and $t + 1$.

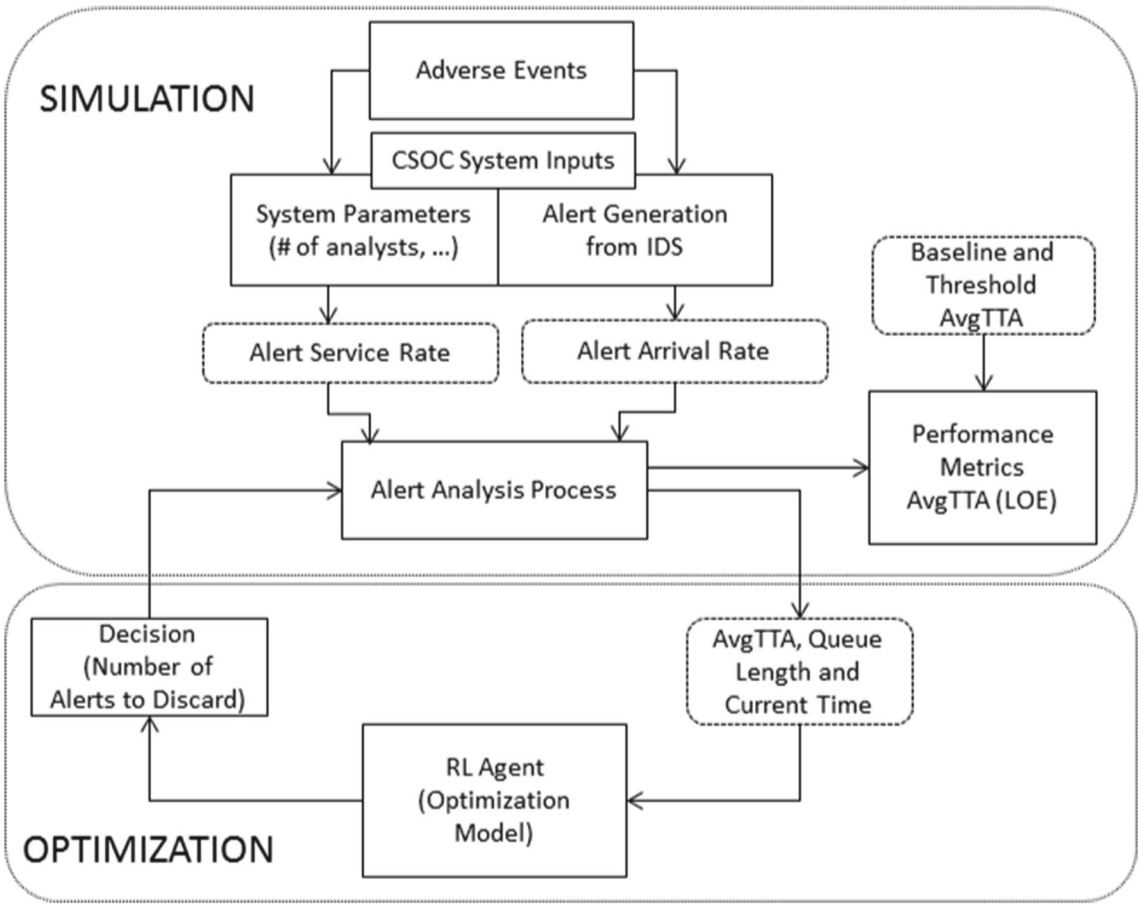

Fig. 3 Dynamic optimization model framework

- *State Transition Function*: $s_{t+1} = h(s_t, x_t, W_{t+1})$: This function defines how the next system state at time $t+1$ is evolved. The state transition probabilities are unknown for this research problem. The concept of a post-decision state variable [16] is used in this approach, which is explained next.
- *Post-decision State (PDS)* variable represents the state of the system after a decision is made and is represented by s_t^x. It is to be noted that the PDS variable represents the state of the system before the exogenous information, W_{t+1} arrives.
- *Contribution Function*: $C(s_t, x_t) = w_a * a_t + w_q * q_t$. The immediate reward has two terms. The first term gives a high reward if the AvgTTA is at or below the baseline number and the second term gives a high reward for maximizing the analyst utilization by keeping the alert queue length long enough. The values for a_t and q_t are obtained using the functions f_a and f_q, respectively. w_a and w_q are the weights assigned to the first and the second term, respectively, by the stakeholders at the CSOC. These weights denote the preference for each of the objectives. It is to be noted that $w_a + w_q = 1$.
- *Objective Function*: The objective of the dynamic optimization model is to maximize the rewards over the entire time horizon of the problem. It is to be noted that the dynamic optimization model differs from the static optimization model by learning the long-run total discounted values of the states $V(S)$ and making decisions such that the system moves from one good state to another. The recursive Bellman's optimality equation is used to achieve this objective, which is given as follows:

$$V^j(s_{t-1}^x) = (1 - \alpha^{j-1})V^{j-1}(s_{t-1}^x) + \alpha^{j-1}\eta^j \quad (4)$$

$$\eta^j = \max_{x_t \in X_t} \{C(s_t, x_t) + \beta V^{j-1}(s_t^x)\} \quad (5)$$

where α^j is the learning parameter that is decayed gradually over several iterations, j is the iteration index, X_t is the set of all feasible actions from which the SDP algorithm will choose a decision at every iteration, and β is the fixed discount factor that allows the state values to converge in the long run. It should be noted that for a decision x_t taken at the beginning of hour t from state s_t, the update of the value of post-decision state s_{t-1}^x from the last hour at $t-1$ that had put the system into state

s_t is executed after reaching state s_t^x as per Equation (4) [16].

3.4.3 Phases of RL-based dynamic optimization algorithm

The RL-based optimization consists of three phases as follows:

1. Exploration Phase: In this phase, the SDP algorithm would explore several suboptimal actions, and acquire the value of system states that are visited. Equation (4) is executed, but without the max operator in Equation (5), by taking random actions $0 \leq x_t \leq X_t$ on the number of alerts that can be discarded between time t and $t+1$. Since the algorithm begins with all $V^0(s) = 0\,\forall s$ at $j = 0$, exploration helps to populate the values of some of the states that are visited. Exploitation (next phase) starts after a certain number of iterations, which depends on the size of the state-space, and number of iterations planned for the entire learning phase (exploration and exploitation).
2. Exploitation Phase: In this phase, the SDP algorithm would take (near-) optimal decisions at time t, which is obtained by executing the right side of Equation (5) with the max operator after attaining better estimates of the value of the states visited during exploration. Using η^j from Equation (5), the value of the previous post decision state is updated at time t as per Equation (4). Learning is stopped when convergence of the value of the states is achieved, as measured in terms of the mean-squared error (MSE) of the stochastic gradient as described below:

$$MSE_j = \frac{\sum_{a=1}^{j}(V^a(s_{t-1}^x) - \eta^a)^2}{j} \qquad j \neq 0 \qquad (6)$$

3. Implementation Phase: In the implementation phase of the SDP algorithm, the value of the states at the time when learning was terminated are used as inputs to make near-optimal decisions at each time t. This is obtained from Equation (5) with the max operator by evaluating all the feasible actions and choosing an action that takes the system to the post-decision state with the highest value of η^j in Equation (5).

The algorithm for the RL-based dynamic optimization model is presented in Algorithm 2.

Algorithm 2: RL-based Dynamic Optimization Algorithm

Input: Number of iterations for learning J, % of iterations for exploration phase m, discount parameter β, initial learning parameter α^0, and time at the end of horizon T.
Output: Long-run state values $V(s), \forall s$
Initialize $V^0(s) = 0, \forall s$
$M = m * J$ /* number of iterations in the exploration phase */
for $j = 1, 2, \ldots, J$ **do**
  for $t = 1, 2, \ldots, T$ **do**
    if $(j \leq M)$ /* *Exploration Phase* */, **then**
      Pick an arbitrary action x_t
      Compute $C(s_t, x_t)$
      $\eta^j = C(s_t, x_t) + \beta V^{j-1}(s_t^x)$
      else if $(j > M)$ /* *Exploitation Phase* */, **then**
        $\eta^j = \max_{x_t \in X_t}\{C(s_t, x_t) + \beta V^{j-1}(s_t^x)\}$
      end
    end
    $V^j(s_{t-1}^x) = (1 - \alpha^{j-1})V^{j-1}(s_{t-1}^x) + \alpha^{j-1}\eta^j$ /* PDS value */
    Generate W_{t+1} /* uncertainty through simulation or real-world */
    $s_{t+1} = h^W(s_t, x_t, W_{t+1})$ /* state transition function */
  end
  $MSE_j = \frac{\sum_{a=1}^{j}(V^a(s_{t-1}^x) - \eta^a)^2}{j}$ /* MSE */
  Decay the learning parameter, $\alpha^j = \frac{\alpha^{j-1}}{1+e}$, where
  $e = \frac{j^2}{6 \cdot 10^{15} + j}$ /* alpha decay scheme [12] */
  $V^j(s) = V^{j-1}(s), \forall s$
end
return $V(s), \forall s$

Table 1 Inputs for experiments

Number of clusters of sensors	10
Average time between alert generation (s)	Expo(18.8)
Number of available analysts	10
Average time taken to investigate an alert (s)	15
Baseline AvgTTA	1
Threshold AvgTTA	4

4 Experimental setup

The following section presents the experimental setup that is used in this work. The inputs used in the experiments, as shown in Table 1, are taken from [21]. A baseline performance is established for the CSOC using the simulation algorithm in [22] with an exponential distribution governing the time between alert generation and a deterministic alert service rate for the available analysts (see Table 1 for the input values). Clusters of sensors are formed to uniformly balance the expected number of alerts (based on historical data) between them. The alerts are investigated by the available analysts in a first-come, first-served (FCFS) manner. An analyst spends 80% of time analyzing alerts while the remainder of 20% is spent on tasks such as updating signatures for the IDS and report writing. It is to be noted that the analysts are staffed such that the service rate of alerts (μ) is kept higher than the arrival rate of alerts (λ), i.e., traffic intensity, $\rho = (\lambda/\mu) < 1$. The alert analysis process is modeled

Table 2 Uncertain events for experiments

Event 1	30% increase in alert generation for 8 hours
Event 2	40% increase in alert generation for 8 hours
Event 3	1 Analyst absent in a shift (12 hours)
Event 4	2 Analysts absent in a shift (12 hours)
Event 5	New vulnerability that increases alert investigation time by 5 times for 8 hours
Event 6	New vulnerability that increases alert investigation time by 10 times for 12 hours
Event 7	Communication breakdown between sensors/IDS and CSOC for 12 hours

as a M/D/c/FCFS queueing model, as given in [22]. The LOE of the CSOC is deemed to be ideal at the respective baseline AvgTTA value. The CSOC is deemed to be operating with an unacceptable LOE for any value for AvgTTA which is equal to or higher than the threshold value set by the CSOC. The average alert queue length for the baseline AvgTTA value (1 hour) is established at 1175 alerts upon reaching the steady state conditions in the simulation. Similarly, the average alert queue length for the threshold AvgTTA value (4 hours) is established at 4350 alerts. The number of alerts that could be investigated per hour by 10 analysts with 80% of time spent in alert analysis work in a CSOC is 1920 (10 analysts*80% of effort towards alert analysis*3600/15 alerts per hour).

As described earlier, the RL-based (dynamic optimization) model is executed in three phases: exploration, exploitation, and implementation. It is to be noted that there are weights assigned to the reward terms in the contribution function. There are three cases considered for the experiments: Case I considers equal weights assigned to both the reward terms, i.e., $w_a = w_q = 0.5$. In case II, w_q is set higher than w_a, indicating a preference for maintaining a longer alert queue length. In case III, w_a is set higher than w_q, indicating a preference for maintaining a lower AvgTTA value. The values of discount parameter β and initial learning parameter α^0 are typically chosen close to (but less than) 1. We used the following values for the parameters in Algorithm 2: $\beta = 0.9, \alpha^0 = 0.8$, and $m = 10\%$. The SDP learning (exploration and exploitation) phase achieves convergence for the values of the states over 1000 iterations of the 14-day work cycles (i.e., with $J = 1000 * 336$ in Algorithm 2). Once good estimates of the values of the states are achieved, the RL-based strategy is executed in the implementation phase for the experiments (sample realizations).

The proposed optimization models, value-based and RL-based, are evaluated against the commonly employed methods at a CSOC, namely the threshold-based and the (analyst resource) utilization-based strategies for maintaining the LOE under adverse conditions.

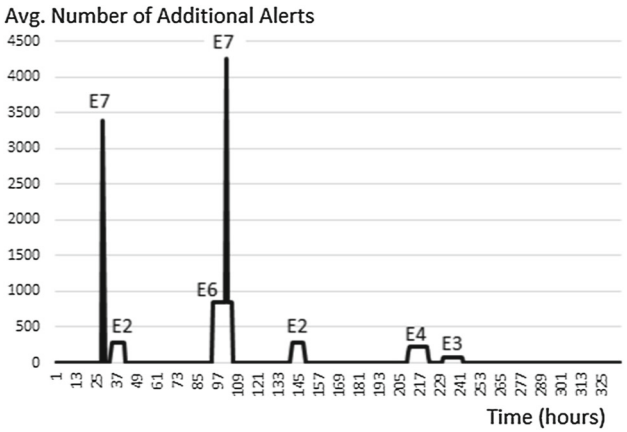

Fig. 4 Temporal patterns in a sample realization of adverse events

5 Experiments and analysis of results

This section presents the experiments conducted using the simulated framework presented in [22]. Several adverse events (see Table 2) are generated in a 14-day work cycle and various corrective strategies are evaluated, namely, the threshold-based, the utilization-based, the value-based (static optimization), and the RL-based (dynamic optimization). The algorithms for the value-based and the RL-based (all three cases) optimization models are tested on 50 simulation runs of the 14-day work cycle, where several types of disruptive events (from Table 2) are simulated for each run. The threshold-based and the utilization-based strategies are also tested on the same set of adverse events as experienced by the CSOC using the value-based and the RL-based strategies (by using the same random seed in the simulator for generating adverse events). The weights considered for the three cases in the RL-based strategy are: $w_a = w_q = 0.5$ for case I, $w_a = 0.2$ and $w_q = 0.8$ for case II, and $w_a = 0.8$ and $w_q = 0.2$ for case III. The dynamic behavior of AvgTTA throughout the 14-day work cycle is captured and reported from each of these strategies in the results described below.

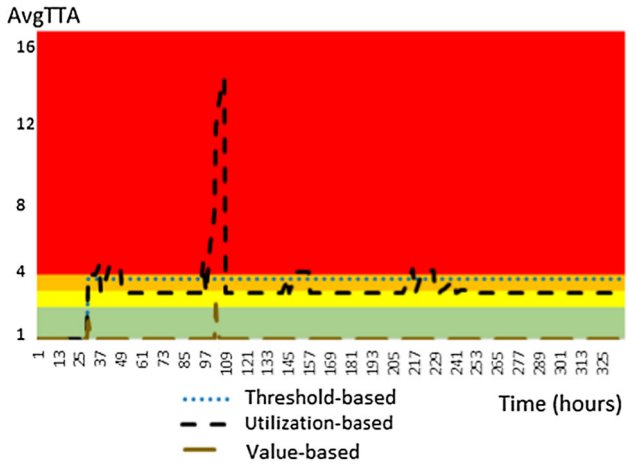

Fig. 5 Dynamic behavior of AvgTTA

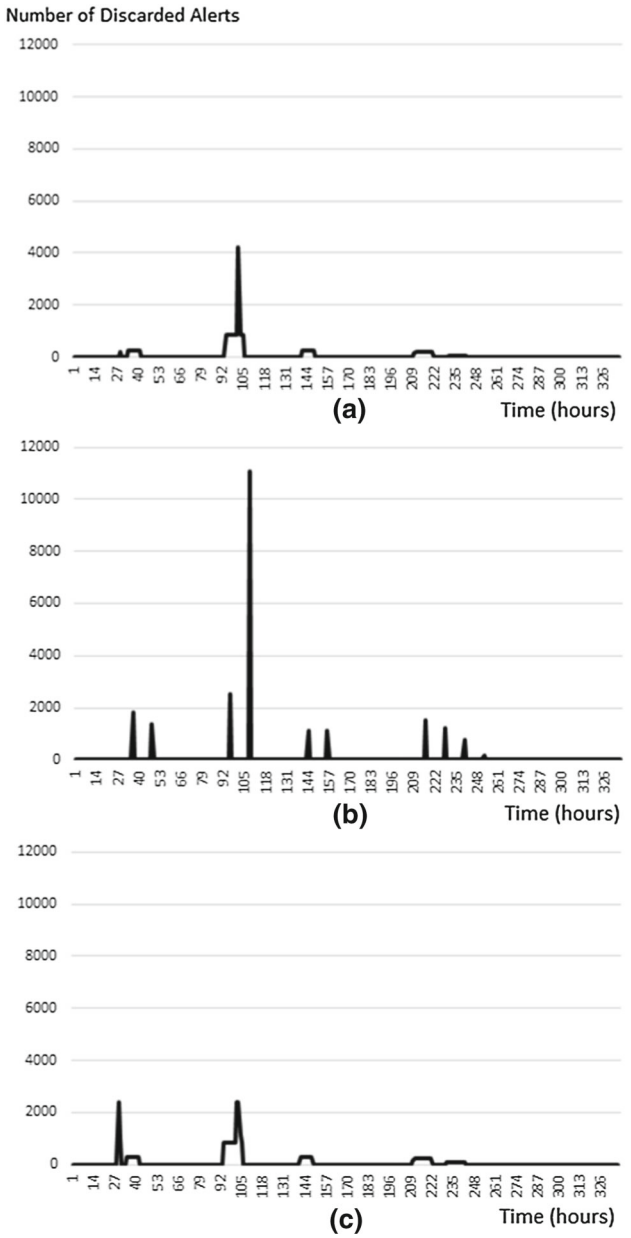

Fig. 6 Timing and amount of alerts discarded in **a** threshold-based strategy, **b** utilization-based strategy and **c** value-based strategy

Figure 4 shows a sample realization for the temporal patterns observed in the adverse events during a 14-day work cycle. The same random seed for the generation of these adverse events is used for the evaluation of all the strategies. First, a comparison between the performances of the threshold-based, the utilization-based, and the value-based optimization strategies is presented.

Figure 5 shows the AvgTTA observed per hour using each of the aforementioned strategies plotted against the background of the color-coded LOE representation. The temporal patterns in the decision-making of the corrective actions for each of these strategies are shown in Fig. 6a–c. It is to be noted that the AvgTTA values are captured after the corrective actions are taken at each time stamp (hourly). It can be seen from Fig. 5 that the AvgTTA is maintained in the orange zone, just under the threshold value, using the threshold-based strategy. With the utilization-based strategy, the AvgTTA is observed to cross the threshold value many times during the 14-day work cycle. The utilization-based strategy keeps on accumulating additional alerts until the investigation capacity for the work-shift is reached. Only upon reaching the full capacity, this strategy deploys the corrective action of discarding alerts. Hence, the AvgTTA is seen to climb high into the red zone with the occurrence of the fourth event (E7 in Fig. 4), while the alerts are being accumulated. And once this capacity is reached and new alerts are still being generated, this strategy starts discarding the longest waiting alerts, as shown with a spike in the number of alerts discarded in Fig. 6b. The value-based optimization strategy is observed to maintain the AvgTTA closest to the baseline value throughout the 14-day work cycle. As described earlier, the value-based strategy is optimized to attain the highest score possible from a cumulative normalized score for AvgTTA and the alert queue length at any given time. Hence, in order to attain this maximum score, the value-based optimization strategy takes more number of minor corrective actions (less number of alerts discarded at each action), as shown in Fig. 6c. However, in order to maintain the AvgTTA close to the baseline, the value-based optimization strategy is observed to have discarded the most number of alerts when compared to the threshold-based and the utilization-based strategies (see Fig. 7).

The above results show that while a value-based strategy succeeds in keeping the AvgTTA as close to the baseline

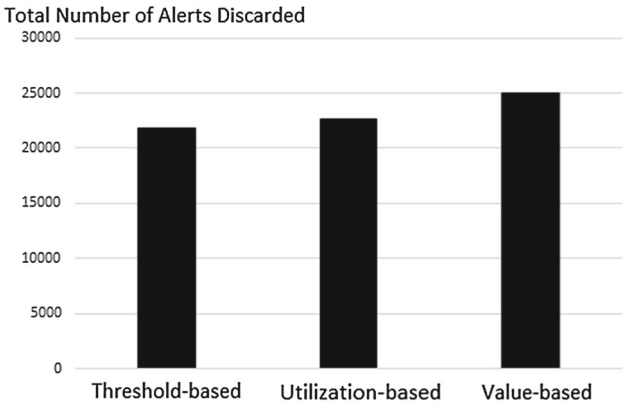

Fig. 7 Total number of alerts discarded during the 14-day work cycle

value as possible, it also discards the maximum number of alerts in the process. As described earlier, this is due to the fact that this optimization strategy does not take into account the future uncertainties arising from the adverse events at a CSOC. The RL-based strategy takes into consideration the values of the future states that could be reached by simulating uncertainty and learning (near-) optimal actions to take at all times (hourly, in this work). Next, the performance of the value-based strategy is compared with the three cases of the RL-based strategy.

Figure 8a–d shows the AvgTTA per hour for the optimization strategies (static and dynamic). It is to be noted that the plot shown in Fig. 8a is the same as shown for the value-based strategy in Fig. 5 (represented by the brown line.) The largest AvgTTA values are observed for case I (Fig. 8b) and case II (Fig. 8c) of the RL-based strategy, where a lower (or an equal) weight was assigned to the first reward term of the contribution function representing the need for maintaining a lower AvgTTA value compared to that assigned to the second reward term representing the need for maintaining a longer alert queue length. Hence, the LOE of the CSOC reaches into the orange zone, while discarding the lowest total number of alerts in these two cases among the static and dynamic optimization strategies (Fig. 10). The amount of time the LOE of the CSOC stays in the yellow zone is similar among the value-based and the RL-based case III strategies (Fig. 8a and d). However, the alert queue length is maintained relatively higher in the RL-based case III strategy than in the value-based strategy, and hence, it can be observed that the number of times the action to discard alerts and the number of alerts discarded are lower using the RL-based strategy than that found using the value-based strategy (Fig. 9a and d).

Figure 10 shows the total number of alerts discarded for the given sample realization in the 14-day work cycle. The RL-based strategy (cases I, II, and III) discards less number of alerts compared to the value-based strategy. It is also noted that the RL-based strategy discards the fewest number of alerts compared to the currently employed strategies (threshold-based and utilization-based) by the CSOC. Similar observations were made in other simulation runs (50) where the RL-based strategy outperformed, on an average, the other 3 strategies in terms of maintaining the AvgTTA in the acceptable zones and the alert queue length as long as possible.

5.1 Computational complexity and scalability

The system state vector, as described in Sect. 3, is 3-dimensional in the stochastic dynamic programming formulation for the research problem. The normalized values for the first two variables in the system state, representing the AvgTTA and the alert queue length, respectively, can go up to two decimal places between 0 and 1, i.e., each variable yields 100 values. The third variable represents the current time (hour) in the 14-day work cycle and yields 336 different values. Hence, the system space consists of under three and a half million ($100 \times 100 \times 336$) states. By defining the state of the system with normalized values between 0 and 1 for the first two terms, we are able to avoid the curse of state space dimensionality, which makes the algorithm scalable for any range of alert backlog numbers at a CSOC. This representation is able to guide the RL agent in taking (near-) optimal actions in any CSOC environment.

5.2 Meta-principles derived from the study

A summary of results is given in Table 3. It is evident from Table 3 that while other strategies may achieve at par performance with the RL case III strategy in only one among the four performance metrics, the RL case III strategy performs the best in all four performance metrics. It should be noted that the performance metrics of RL case I and case II strategies excelled in only a few of the four performance metrics and, therefore, are not included in Table 3. The meta-principles derived from the preceding study are as follows.

- The RL-based strategy presented in case III is able to keep the LOE of the CSOC in the best zones possible throughout the 14-day work cycle, while discarding the fewest number of alerts compared to the threshold-based, utilization-based, and value-based strategies.
- The threshold-based strategy, by design, is able to keep the AvgTTA under the threshold value in all of the simulation runs. However, it is found to discard the alerts too soon in the cases where adverse events occur early on in the 14-day work cycle, and thereby reducing the analyst utilization (i.e., analysts are idling) later in the work cycle.

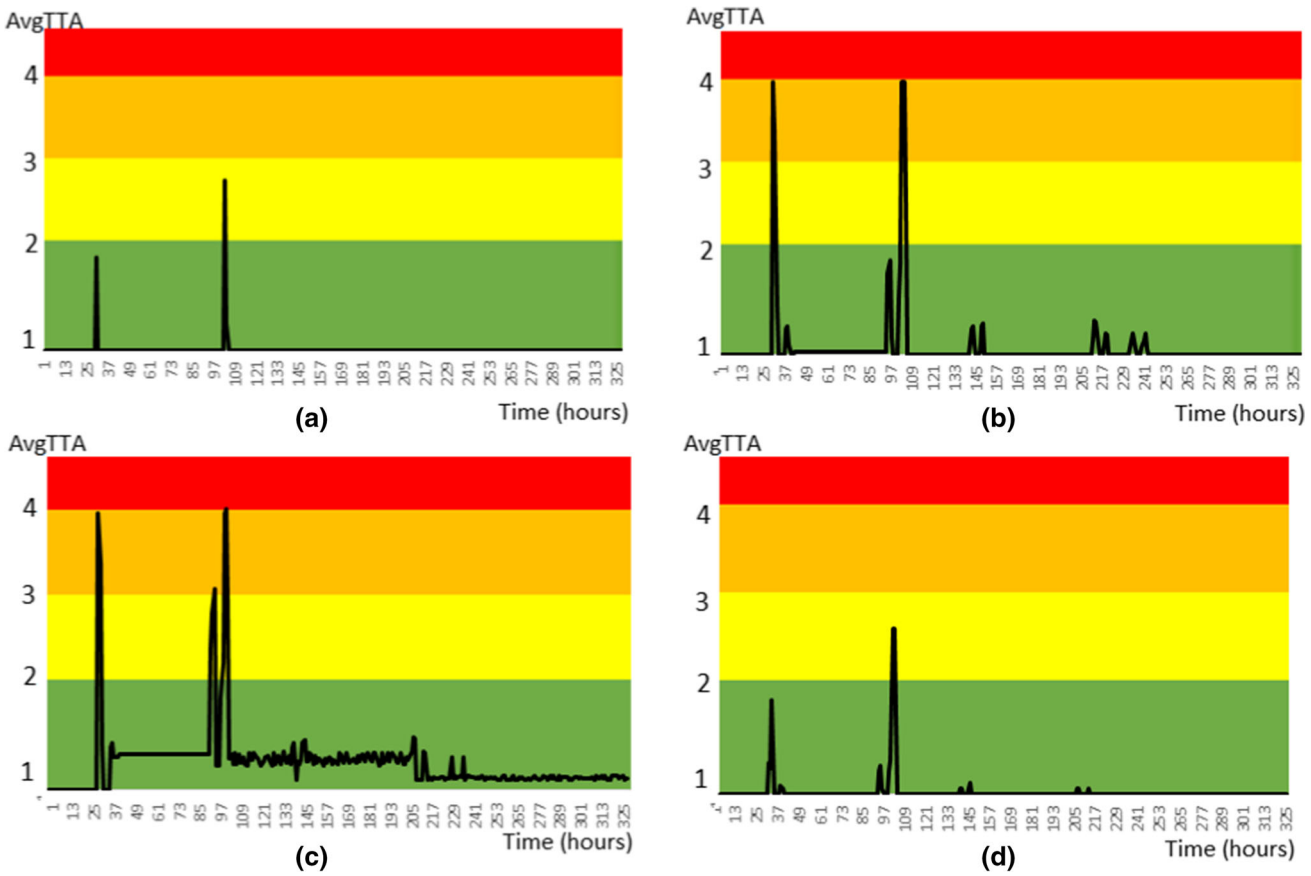

Fig. 8 Dynamic behavior of AvgTTA using **a** value-based Strategy, **b** RL-based strategy: case I, **c** RL-based strategy: case II, and **d** RL-based strategy: case III

– The (analyst resource) utilization-based strategy is unable to keep the AvgTTA under the threshold value across the 14-day work cycle, though it attains the maximum analyst utilization among all of the strategies.
– The value-based strategy discarded more alerts than the other strategies in order to maintain the AvgTTA close to the baseline. This strategy runs the risk of discarding too many alerts, when the desiderata are to discard the minimum.
– The RL agent with the unique representation for the system state variable increases the practical aspect of the proposed research work and makes the model deployable in any real-world CSOC environment. In practice, a CSOC must set the weights in the contribution function such that a higher preference is given to maintaining a lower AvgTTA value compared to a longer alert queue length (i.e., case III with $w_a > w_q$) to be able to meet all the desiderata of this paper.

6 Conclusions and future work

The paper presented four strategies for discarding alerts in order to maintain the LOE of a CSOC under adverse conditions. One of the strategies is an intelligent strategy that uses reinforcement learning to make decisions on when and how many alerts to discard during the shift operation. It discarded the lowest number of alerts over several experimental

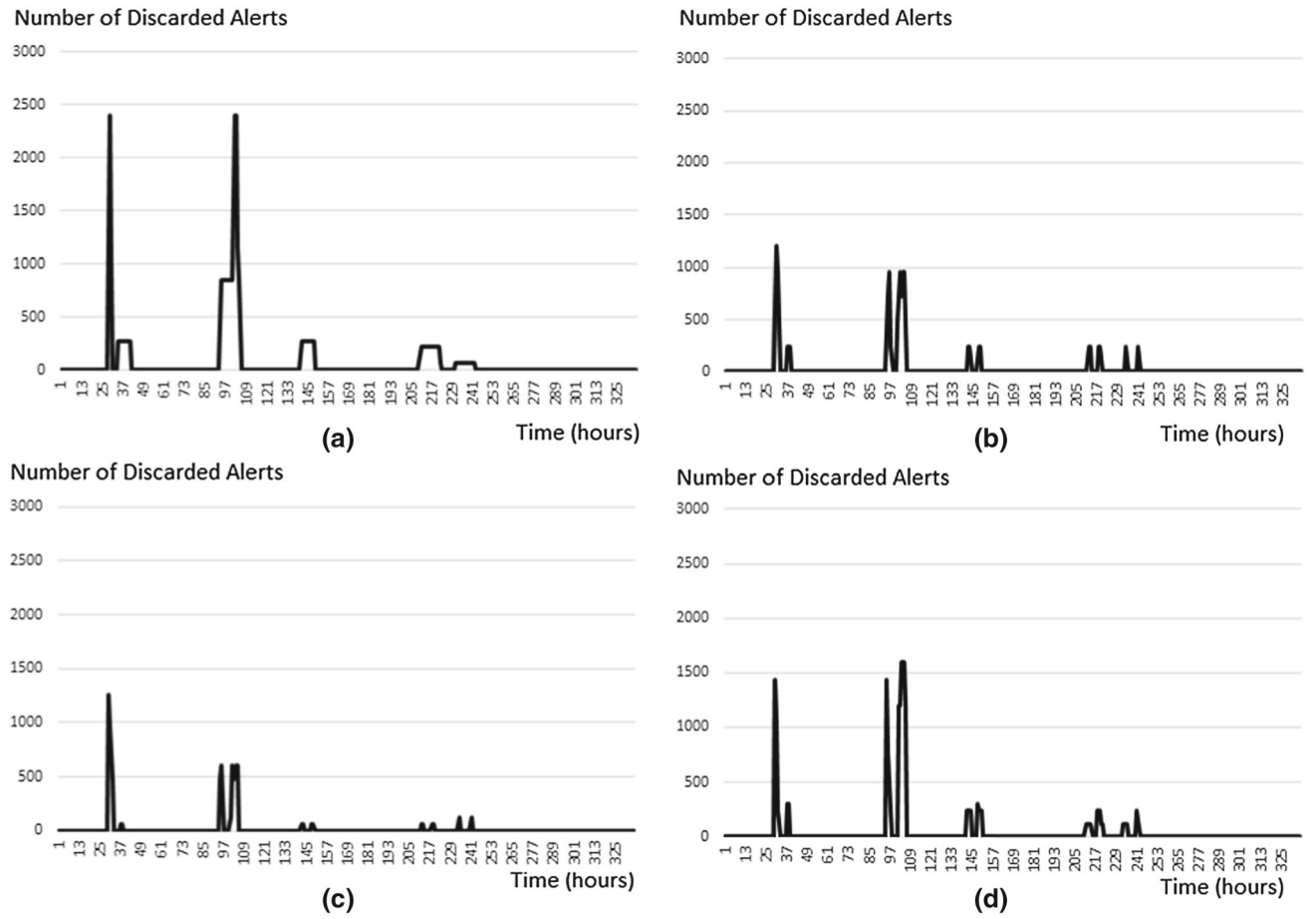

Fig. 9 Timing and amount of alerts discarded in **a** value-based strategy, **b** RL-based strategy: case I, **c** RL-based strategy: case II, and **d** RL-based strategy: case III

simulation runs, determined the right timing to discard the alerts, and maintained the AvgTTA below the threshold set by the CSOC. Under adverse conditions and with limited analyst resource, a CSOC manager can use the RL-based strategy as a decision-support system that guarantees the AvgTTA below the threshold value in every hour of shift operation. Such a guarantee with minimum number of alerts discarded is a paradigm shift in how CSOCs with limited resources (both regular and on-call analysts combined) can efficiently operate in the face of uncertainties that affect the length of the alert backlog for investigation.

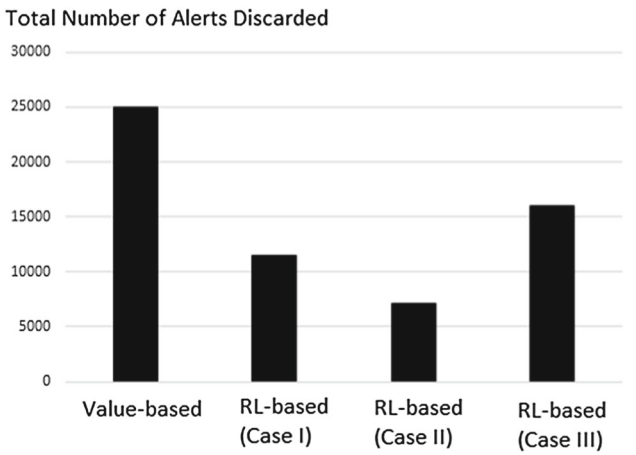

Fig. 10 Total number of alerts discarded during the 14-day work cycle

Table 3 Summary of experiments: √ : best performance, blank: intermediate performance, and X: worst performance

Desiderata	Threshold	Utilization	Value	RL (Case III)
Minimizing number of alerts discarded			X	√
Ensuring highest utilization of analysts	X	√		√
Determining optimal time to discard alerts			X	√
Maintaining LOE closest to the baseline		X	√	√

Acknowledgements The authors would like to thank Dr. Cliff Wang of the Army Research Office for the many discussions which served as the inspiration for this research.

Funding This study was funded by the Army Research Office (grant number W911NF-13-1-0421), by the Office of Naval Research (grant numbers N00014-18-1-2670 and N00014-20-1-2407), and by the National Science Foundation under (grant number CNS-1822094).

Declarations

Conflicts of interest Authors Shah, Ganesan, Jajodia, and Cam declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Aydin, M.E., Oztemel, E.: Dynamic job-shop scheduling using reinforcement learning agents. Robot. Auton. Syst. **33**(2), 169–178 (2000)
2. Bejtlich, R.: The Tao of Network Security Monitoring: Beyond Intrusion Detection. Pearson Education Inc., London (2005)
3. Bhatt, S., Manadhata, P.K., Zomlot, L.: The operational role of security information and event management systems. IEEE Secur. Privacy **12**(5), 35–41 (2014)
4. CIO: DON cyber crime handbook. Dept. of Navy, Washington, DC (2008)
5. Crothers, T.: Implementing Intrusion Detection Systems. Wiley Publishing Inc., New York (2002)
6. D'Amico, A., Whitley, K.: The real work of computer network defense analysts. In: VizSEC 2007: Proceedings of the Workshop on Visualization for Computer Security (2008)
7. D'Amico, A., Whitley, K.: The real work of computer network defense analysts: The analysis roles and processes that transform network data into security situation awareness. In: Proceedings of the Workshop on Visualization for Computer Security, pp. 19–37 (2008)
8. Edwards, B., Hofmeyr, S., Forrest, S.: Hype and heavy tails: a closer look at data breaches. J. Cybersecur. **2**(1), 3–14 (2016)
9. Farris, K.A., Shah, A., Cybenko, G., Ganesan, R., Jajodia, S.: VULCON–a system for vulnerability prioritization, mitigation, and management. ACM Trans. Privacy Secur. **21**(4), 16:2-16:28 (2018)
10. Ganesan, R., Jajodia, S., Cam, H.: Optimal scheduling of cybersecurity analyst for minimizing risk. ACM Trans. Intell. Syst. Technol. **8**(4), 1–32 (2017)
11. Ganesan, R., Jajodia, S., Shah, A., Cam, H.: Dynamic scheduling of cybersecurity analysts for minimizing risk using reinforcement learning. ACM Trans. Intell. Syst. Technol. **8**(1), 1–21 (2016). https://doi.org/10.1145/2882969
12. Gosavi, A.: Simulation Based Optimization: Parametric Optimization Techniques and Reinforcement Learning. Kluwer Academic, Norwell, MA (2003)
13. Killcrece, G., Kossakowski, K.P., Ruefle, R., Zajicek, M.: State of the practice of computer security incident response teams (csirts). Tech. Rep. CMU/SEI-2003-TR-001, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA (2003)
14. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013)
15. Paternina-Arboleda, C.D., Das, T.K.: A multi-agent reinforcement learning approach to obtaining dynamic control policies for stochastic lot scheduling problem. Simul. Modell. Pract. Theory **13**(5), 389–406 (2005)
16. Powell, W.B.: Approximate Dynamic Programming: Solving the Curses of Dimensionality. Wiley-Interscience, New York (2007)
17. Rasoulifard, A., Bafghi, A.G., Kahani, M.: Incremental hybrid intrusion detection using ensemble of weak classifiers. In: Advances in Computer Science and Engineering, pp. 577–584. Springer, Cham (2008)
18. Scarfone, K., Mell, P.: Guide to intrusion detection and prevention systems (IDPS). Special Publication 800-94, NIST (2007)
19. Shah, A., Ganesan, R., Jajodia, S.: A methodology for ensuring fair allocation of CSOC effort for alert investigation. Int. J. Inf. Secur. (2018). https://doi.org/10.1007/s10207-018-0407-3
20. Shah, A., Ganesan, R., Jajodia, S., Cam, H.: Adaptive reallocation of cybersecurity analysts to sensors for balancing risk between sensors. Serv. Orient. Comput. Appl. (2018). https://doi.org/10.1007/s11761-018-0235-3
21. Shah, A., Ganesan, R., Jajodia, S., Cam, H.: Dynamic optimization of the level of operational effectiveness of a CSOC under adverse conditions. ACM Trans. Intell. Syst. Technol. **9**(5), 51:1-51:20 (2018). https://doi.org/10.1145/3173457
22. Shah, A., Ganesan, R., Jajodia, S., Cam, H.: A methodology to measure and monitor level of operational effectiveness of a CSOC. Int. J. Inf. Secur. **17**(2), 121–134 (2018). https://doi.org/10.1007/s10207-017-0365-1
23. Shah, A., Ganesan, R., Jajodia, S., Cam, H.: Optimal assignment of sensors to analysts in a cybersecurity operations center. IEEE Syst. J. (2018). https://doi.org/10.1109/JSYST.2018.2809506
24. Shah, A., Ganesan, R., Jajodia, S., Cam, H.: Understanding tradeoffs between throughput, quality, and cost of alert analysis in a CSOC. IEEE Trans. Inf. Forens. Secur. **14**(5), 1155–1170 (2019)
25. Smith, M., Paté-Cornell, E.: Cyber risk analysis for a smart grid: How smart is smart enough? a multi-armed bandit approach. In: Proceedings of the 2nd Singapore Cyber-Security R&D Conference (SG-CRC 2017), pp. 37–56 (2017)
26. Sundaramurthy, S.C., Bardas, A.G., Case, J., Ou, X., Wesch, M., McHugh, J., Rajagopalan, S.R.: A human capital model for mitigating security analyst burnout. In: Eleventh Symposium on Usable Privacy and Security (SOUPS 2015), pp. 347–359. USENIX Association (2015)
27. Sundaramurthy, S.C., McHugh, J., Ou, X., Wesch, M., Bardas, A.G., Rajagopalan, S.R.: Turning contradictions into innovations or: How we learned to stop whining and improve security oper-

ations. In: Twelfth Symposium on Usable Privacy and Security (SOUPS 2016), pp. 237–250. USENIX Association (2016)
28. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT press, Cambridge (2018)