Dynamic Optimization of the Level of Operational Effectiveness of a CSOC Under Adverse Conditions

ANKIT SHAH, RAJESH GANESAN, and SUSHIL JAJODIA, George Mason University
HASAN CAM, Army Research Laboratory

The analysts at a cybersecurity operations center (CSOC) analyze the alerts that are generated by intrusion detection systems (IDSs). Under normal operating conditions, sufficient numbers of analysts are available to analyze the alert workload. For the purpose of this article, this means that the cybersecurity analysts in each shift can fully investigate each and every alert that is generated by the IDSs in a reasonable amount of time and perform their normal tasks in a shift. Normal tasks include analysis time, time to attend training programs, report writing time, personal break time, and time to update the signatures on new patterns in alerts as detected by the IDS. There are several disruptive factors that occur randomly and can adversely impact the normal operating condition of a CSOC, such as (1) higher alert generation rates from a few IDSs, (2) new alert patterns that decrease the throughput of the alert analysis process, and (3) analyst absenteeism. The impact of the preceding factors is that the alerts wait for a long duration before being analyzed, which impacts the level of operational effectiveness (LOE) of the CSOC. To return the CSOC to normal operating conditions, the manager of a CSOC can take several actions, such as increasing the alert analysis time spent by analysts in a shift by canceling a training program, spending some of his own time to assist the analysts in alert investigation, and calling upon the on-call analyst workforce to boost the service rate of alerts. However, additional resources are limited in quantity over a 14-day work cycle, and the CSOC manager must determine when and how much action to take in the face of uncertainty, which arises from both the intensity and the random occurrences of the disruptive factors. The preceding decision by the CSOC manager is nontrivial and is often made in an ad hoc manner using prior experiences. This work develops a reinforcement learning (RL) model for optimizing the LOE throughout the entire 14-day work cycle of a CSOC in the face of uncertainties due to disruptive events. Results indicate that the RL model is able to assist the CSOC manager with a decision support tool to make better decisions than current practices in determining when and how much resource to allocate when the LOE of a CSOC deviates from the normal operating condition.

Categories and Subject Descriptors: C.2.3 Network Monitoring [I.2.6 Learning]: I.2.8 Scheduling

General Terms: Cybersecurity, Analysts, Reinforcement Learning, Allocate Resources, On-Call Analysts, Level of Operational Effectiveness, Average Time to Analyze Alerts

Additional Key Words and Phrases: Cybersecurity, analysts, reinforcement learning, allocate resources, on-call analysts, level of operational effectiveness, average time to analyze alerts, stochastic optimization, resource allocation, absenteeism in shift

A. Shah, R. Ganesan, and S. Jajodia were partially supported by the Army Research Office under grants W911NF-13-1-0421 and W911NF-15-1-0576 and by the Office of Naval Research under grant N00014-15-1-2007.

Authors' addresses: A. Shah, R. Ganesan, and S. Jajodia, Center for Secure Information Systems, George Mason University, 4400 University Dr., MS 5B5, Fairfax, VA 22030-4422; emails: {ashah20, rganesan, jajodia}@gmu.edu; H. Cam, Army Research Laboratory, 2800 Powder Mill Road, Adelphi, MD 20783-1138; email: hasan.cam.civ@mail.mil.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 ACM 2157-6904/2018/04-ART51 \$15.00

<https://doi.org/10.1145/3173457>

ACM Reference format:

Ankit Shah, Rajesh Ganesan, Sushil Jajodia, and Hasan Cam. 2018. Dynamic Optimization of the Level of Operational Effectiveness of a CSOC Under Adverse Conditions. *ACM Trans. Intell. Syst. Technol.* 9, 5, Article 51 (April 2018), 20 pages.

<https://doi.org/10.1145/3173457>

1 INTRODUCTION

Sensors are placed in a network to monitor the traffic flow of data in the network. The sensor data collected from this monitoring is analyzed by an intrusion detection system (IDS), which issues alerts. Next, the alerts are received at the cybersecurity operations center (CSOC) for investigation. The task of a cybersecurity analyst at a CSOC includes examining the alerts generated by an IDS such as SNORT or a security information and event management (SIEM) tool such as ArcSight (Bhatt et al. 2014), and identifying innocuous alerts and those that are considered as significant, which require further investigation. All significant alerts are categorized on a scale of 1 to 9 with categories 1, 2, 4, and 7 (incidents) being severe alerts. Category 1, 2, 4, and 7 alerts are analyzed and reported to a watch officer and then a report has to be written. Table 3 (shown later in the Electronic Appendix) provides an example of cyber incident and event categories used by the U.S. Department of Defense (CIO 2008). The focus of this article is the operational efficiency of the preceding process of examination and identification that classifies an alert as innocuous or significant. Hence, any further categorization, analysis, and mitigation of the significant alerts is beyond the scope of this work.

For the preceding process of examination and identification of innocuous or significant alert, every CSOC defines a normal operating condition in which the arrival rate of alerts and service rate of alerts are at a certain level, which guarantees a certain predetermined acceptable level of operational effectiveness (LOE). In Shah et al. (2017), the authors quantify the LOE of a CSOC by defining a new metric called *total time for alert investigation* (TTA), which is the sum of the waiting time in the queue and the analyst investigation time of an alert after its arrival in the CSOC database. For normal operating conditions, the CSOC has an established or baseline average TTA per hour (avgTTA/hr), which is calculated using the TTA values of all alerts that were investigated in an hour. The avgTTA/hr is monitored over the course of the work shift from which the LOE of a CSOC is ascertained by calculating the deviation of avgTTA/hr from its baseline value. Shah et al. [2017] use a color-coded representation of the LOE to measure the extent of deviation of avgTTA/hr from its baseline value. It is a requirement of the CSOC that the avgTTA/hr remain within a certain upper bound (e.g., 4 hours), which is referred to as the threshold value for avgTTA/hr. Different tolerance bands are created both below and above the threshold value of avgTTA to indicate a color-coded representation of the LOE status of a CSOC as shown in Figure 1.

Several real-world disruptive events affect the normal operating condition of a CSOC, and they can be classified as external or internal events. Some examples of external events include a zero-day attack, sensor downtime, broken communication link between the IDS and the CSOC, power failure, and increased attacker activity on a network that is monitored by the sensors. The internal events include equipment shutdowns at a CSOC, power disruptions, analyst absenteeism, required analyst credentials not available in a shift (e.g., secret clearances), and analyst scheduling issues that do not provide the required expertise mix for alert analysis. The preceding disruptive events affect the arrival rate of alerts and/or service rate of alert investigation. For example, when a broken communication link between an IDS and CSOC is reestablished, there could be a surge in the alert arrival rate. A zero-day attack could increase the alert arrival rate on some of the sensors. Similarly, the detection of a new vulnerability that consumes additional analyst time would increase the time

Fig. 1. Color-coded representation of the LOE (Shah et al. 2017).

to investigate the alerts, which in turn reduces the throughput (number of alerts investigated) per unit of time of the alert investigation process.

The overall impact of the preceding disruptions on the arrival or service rate of alerts is that the queue length of alerts waiting for investigation could increase, thus creating a backlog of alerts. The backlog of alerts directly affects the avgTTA/hr metric and consequently degrades the LOE status of the CSOC. The problem is compounded by the fact that the disruptions occur randomly, and the intensity of its impact is unknown until the uncertainty due to the disruption unfolds over time. The CSOC manager has a nontrivial and difficult decision-making task to bring on additional resources (over and beyond the regular workforce) to restore the LOE status of the CSOC to its normal operational level. However, additional resources are limited and must be well managed. In summary, the preceding discussion highlights the importance of researching for an intelligent decision-making tool that can be used by any CSOC and the significant gains in the LOE that can be obtained by optimally controlling the avgTTA/hr metric under the uncertainty of the disruptive events.

For the convenience of personnel management and scheduling of analyst work shifts, a typical CSOC runs on a 14-day work cycle in which analysts work in 12- or 8-hour shifts. During a work shift, an analyst's time is divided into several activities, such as analyzing alerts, which constitutes about 60% to 80% of the analyst's time, and writing reports, training, and updating signatures in the IDS from new alert patterns, which constitutes the remaining 20% to 40% of the analyst's time. When there is a need to clear the backlog of alerts, the CSOC manager has the option of utilizing the following additional resources. In the order of preference, a CSOC manager could (1) divert portions of the preceding 20% to 40% of analyst time toward alert analysis by canceling a training program or delaying the report writing task, (2) spend a portion of her own time on alert analysis, and (3) bring on-call analysts for a temporary duration to supplement the current analyst workforce. Given the uncertain nature of the disruptive events and the limited availability of additional resources, deploying the additional resources too soon or having too many could result in a lack of resources toward the later part of the 14-day work cycle, which could reduce the LOE status of the CSOC when new disruptive events occur. Similarly, deploying the additional resources too little or too late could result in an elevated avgTTA/hr metric, which reduces the LOE status of the CSOC. Hence, it is imperative to determine when and how many additional resources must be deployed in the face of uncertain disruptive events and limited additional resource availability.

The objective of this work is to develop an intelligent and adaptive decision support tool for the CSOC manager to take optimal actions (when and how many) to allocate the additional resources to maintain an optimal LOE status throughout the 14-day cycle of the CSOC. Due to the dynamic and sequential decision-making framework of the 14-day CSOC operation, the article presents a reinforcement learning (RL)-based model for representing the manager's decision-making process under uncertainty. The RL model takes the continuously monitored LOE status of the CSOC

operation as one of its inputs and takes corrective actions depending on the extent of deviation of the current avgTTA/hr value from the baseline avgTTA/hr value for the CSOC system. The decisions made by the RL model are compared with greedy and rule-based uniformly distributed resource actions to demonstrate the superior decision-making ability of the RL model in the face of uncertainties due to disruptive factors. Although the rule-based actions are limited in their use of future resources in advance (not adaptive), the greedy actions are myopic in nature and respond to surges in alert backlog by allocating additional resources without any consideration of future resource needs.

Although the research builds upon the metric and the dynamic framework presented in Shah et al. (2017) to measure and monitor the LOE of a CSOC, this article focuses on optimally controlling the dynamic behavior of the metric and thereby dynamically optimizing the LOE of a CSOC under adverse conditions with limited resources. In addition, to make dynamic decisions to allocate additional resources when needed, the model in this work uses as input the queueing model described in Shah et al. (2017). In Ganesan et al. [2016] and Ganesan et al. [2017], the authors focus on optimizing the allocation of analysts (static (regular) and dynamic (on-call) workforce) to sensors and scheduling the workforce before the start of the shift to minimize the potential risk of having unanalyzed and/or not thoroughly analyzed (quantity metric) *significant alerts* at the end of the shift. This work differs from that work because (1) it dynamically manages (under uncertainty) the additional resource available to investigate all alerts to identify them as innocuous or significant, which is unlike the published work that is focused only on significant alert investigation, and (2) it optimizes the average total time to analyze (avgTTA/hr metric) for all alerts because unanalyzed alerts can be moved from one shift to another at a CSOC. Hence, the article complements the published literature by focusing on a step that is prior to significant alert investigation by stating that it is detrimental to keep alerts waiting longer than the baseline avgTTA/hr because of the potential negative effects on the CSOC if the alert is found to be significant.

The primary contribution of this article is an adaptive and dynamic decision-making model that is built using the principles of stochastic dynamic programming (SDP) and solved using RL. The model directly assists the CSOC manager to make optimal decisions in the face of uncertainties that affect the LOE status of the CSOC so that the CSOC can be maintained at its optimal LOE status over any given 14-day work cycle. Other contributions include several metaprinciples that provide deeper insights into the dynamic behavior of TTA, and its relationship to both the disruptive events and the manager's actions, which are very useful in designing an efficient CSOC wherein LOE must be optimized over time. The preceding contributions allow for a new paradigm shift in CSOC operation that allows for optimally controlling the avgTTA/hr metric, which ensures an optimum LOE status over time for a given set of CSOC operating conditions and its uncertainties.

The article is organized as follows. Section 2 presents the related literature on the metrics used to measure and monitor several real-world systems with queues. In Section 3, a discussion on the avgTTA model and the RL-based decision-making framework is presented. Additionally, model parameters are discussed. Section 4 presents the experimental setup to study the dynamic behavior of avgTTA/hr and implementation strategies for the RL model. Section 5 presents the numerical experiments that compare both the greedy and rule-based decisions to the RL-based decision-making process to allocate additional CSOC resources in the face of uncertainties. A discussion section is provided that summarizes the research findings. In Section 6, the conclusions of the research are presented.

2 RELATED LITERATURE

A CSOC continuously monitors an organization and provides critical situational awareness. CSOCs have been a focus of research efforts over the years. Equipped with an infrastructure of

several sophisticated tools, cybersecurity analysts thoroughly investigate the alerts that are generated and queued by the IDS or SIEM tools [Bhatt et al. 2014; Zimmerman 2014; Zaccaro et al. 2016]. Automated techniques for detecting malicious behavior [Northcutt and Novak 2002; Di Pietro and Mancini 2008; Subrahmanian et al. 2015] and developing automated alert reduction techniques by reducing false positives (Barbará and Jajodia 2002) have been the focus of many research studies. Understanding the human and organizational problems faced by a CSOC has been studied in Botta et al. (2011) and Furnell et al. (2010). Improving efficiency of analysts by building trust has been studied in Sundaramurthy et al. (2016).

Several queueing statistics have been studied in the published literature to measure and monitor systems with queues, especially in the cases where normal conditions are adversely impacted, resulting in congested systems such as in Helm et al. (2011) and Vansteenwegen and Van Oudheusden (2007). The LOE in the field of cybersecurity, measured by the metric avgTTA/hr, was recently presented in Shah et al. (2017). The research presented the impact of adverse conditions during a work shift on the LOE of a CSOC. Dynamic allocation of resources has been studied in various fields in the published literature. RL has been shown to provide better allocation policies compared to heuristics in cellular telephone systems with a broad variety of call traffic patterns in Singh and Bertsekas (1997). An agent-based dynamic scheduling system was proposed in Aydin and Oztemel (2000), where the agent is trained using RL to make decisions on scheduling of jobs that arrive randomly.

To the best of our knowledge, decision making under adverse conditions to optimize the LOE of a CSOC has not yet been studied or addressed in the literature. Hence, this research will add to the existing body of literature on performance monitoring in the context of a CSOC. Due to the dynamic setting of the CSOC that evolves and changes over time, several decisions must be made over time (a 14-day work cycle) in the face of uncertainty due to disruptive events. The CSOC manager's decision (when to take an action and how much action to take) in terms of allocating additional resources in the face of uncertainty is nontrivial and in current practice is often made in an ad hoc manner that maximizes only the short-term (myopic) benefits while being suboptimal in the long run. Hence, our work uses an SDP approach to model the LOE optimization problem, which maximizes the combined short-term (myopic) and long-term (overall) benefits of making several decisions in the face of uncertainty.

Whereas control theory (Wonham 1979) deals with sequential decision-making problems with continuous-time states, dynamic programming (the Markov decision process) deals with decision-making problems with discrete states (Bellman 1957; Puterman 1994). Further, the literature cites that in the event where state transition probabilities are not available, a stochastic approximation for stochastic dynamic problems could be used to find solutions (Sutton and Barto 1998; Gosavi 2003; Powell 2007). In what follows, the LOE optimization model is presented.

3 THE LOE OPTIMIZATION MODEL

This section presents the details of the LOE optimization model, which is built using the theory of SDP and solved using RL. The detailed alert analysis process, advantages of using SDP for this research problem, and the algorithms are provided in the Electronic Appendix. The CSOC system is assumed to be operating normally, which means that the LOE status and the avgTTA/hr metric are at a predetermined acceptable level. Under normal operating conditions, the additional resources are not needed and the prescheduled analyst staffing levels for a shift are sufficient. However, when there is a disruptive event that impacts the avgTTA/hr metric and, consequently, the LOE status of the CSOC, the LOE optimization model is invoked to provide an hourly decision to the CSOC manager. The nontrivial decision determines whether or not to bring additional resources to the CSOC and the quantity of additional resources to commit if the decision is to act. The objective of the

Fig. 2. The dynamic LOE optimization model framework.

dynamic LOE optimization model is to allocate additional resources such that the LOE can be optimized over the entire 14-day work cycle period. To achieve this objective, the LOE model must consider future disruptions that might arise in the 14-day work cycle (a long-term view of the potential future need for additional resources) while making a decision to allocate additional resources to mitigate the impact of current disruptions on the avgTTA/hr metric (and the LOE status). As in any resource allocation problem, having too few or too many additional resources would make the solution trivial. However, with an adequate quantity of additional resources, the dynamic LOE optimization model is needed to manage the resource allocation optimally over the planning horizon.

3.1 Model Framework

A framework for the dynamic LOE optimization model is provided in Figure 2. The dynamic optimization model consists of two main blocks: the alert analysis process simulation block and the RL-based optimization block, which are executed one after another. As explained later, the time over the 14-day work cycle is indexed in 1-hour timesteps. As shown in Figure 2, at each timestep, the state of the CSOC system is observed and a decision is made by the RL agent. The decision is then ratified by the CSOC manager and implemented for the next hour of CSOC operation. The details of the preceding blocks are presented next.

3.2 Simulation Model of Alert Analysis Process

The simulation model of the alert analysis process consists of four main blocks as shown in Figure 2. They include the CSOC system inputs (system parameters and alert generation by the IDS), the uncertain events (both internal and external) that affect the CSOC system inputs, the alert analysis process block in which the work shift is simulated, and the performance metrics block that captures the LOE status of the CSOC at each point in time using the avgTTA/hr metric.

3.2.1 CSOC System Inputs. This section describes the system parameters and the alert generation rate from the IDS. Under normal operating conditions, all analysts reporting in a shift are

said to be 100% engaged at work, and the percentage of effort (amount of time in a shift) spent on alert analysis is given by U , which is usually 60% to 80% in reality. The rest of the time is spent on writing and submitting incident reports, attending training sessions, reporting to collaborating agencies whose sensors are being monitored, and in generating signatures to update the IDS. The preceding allows for increasing the percentage of effort spent on alert analysis in the event that alerts are backlogged, which causes a reduction in the LOE status of the CSOC. Additionally, under normal operating conditions, the alert arrival rate is less than the alert service rate, which ensures adequate staffing levels.

The CSOC inputs and queueing model described in Section B of the Electronic Appendix, which is used to simulate uncertain events for this article, has been studied in Shah et al. (2017). Figure 1 shows the color-coding scheme for the LOE status of the CSOC, which depends on the value of avgTTA/hr. The avgTTA/hr is given as follows:

$$avgTTA_t = \frac{\sum_i^n TTA_i}{n}, \quad (1)$$

where TTA_i is the individual value of TTA for alert i and n is the number of alerts that completed investigation in the previous hour of CSOC operation between time $t - 1$ and t . It is to be noted that the continuously monitored LOE status (avgTTA/hr that is derived from backlog) from the queueing system is one of the inputs to the RL model.

3.2.2 Uncertain Event Simulation. Section 1 described several disruptive events that can adversely affect the LOE status of a CSOC. They were classified as external (adversarial) and internal (related to the CSOC infrastructure) events. Several disruptive events that were obtained through our discussions with several CSOCs are simulated in this article in random order, which is explained in Section 4. Some examples of these events include an increase in alert generation for a specified number of hours, analyst absenteeism, a new vulnerability that increases alert investigation time, and a communication breakdown between sensors/IDS and CSOC for a specified number of hours. In Section 4, Table 2 shows some of the uncertain events (both internal and external) that affect the CSOC system parameters and the alert generation rate of the IDSs. Based on the literature survey, the arrival of cyber incidents are well modeled by a Poisson process. For example, the work in Kuypers and Paté-Cornell (2016) and Smith and Paté-Cornell (2017) uses the dataset that was obtained from the U.S. Department of Energy through a Freedom of Information Act request by the reporter Stephen Reilly of *USA Today* (Reilly 2015). The authors studied the arrival process of these incidents against many distributions, including gamma, logistic, normal Weibull, and negative binomial, among many others, and empirically found that exponential distribution was the best fit for describing the time between the incidents (Poisson for arrival process distribution). In Edwards et al. (2016), the authors use a dataset published by the Privacy Rights Clearinghouse. They found that the frequency of large breaches (those involving 500,000 or more records) fits a Poisson distribution best. Mathematically, the mean of several Poisson process distributions is also Poisson (Gross et al. 2008). Furthermore, according to the Palm-Khinchine theorem, even if some of the incident arrival processes are possibly non-Poisson, the aggregate arrivals from all of these processes still follow a Poisson distribution (Heyman and Sobel 2004).

The RL-based decision making framework depends on having access to a sequence of random samples, which can be obtained from real-world data, computer simulation, or sampling from a known distribution (Powell 2007). In the absence of a real-world learning dataset due to confidentiality reasons, this article uses a Poisson probability distribution for the arrival of the events shown later in Table 2 to demonstrate the performance of the RL-based approach. In practice, the real-world events at a CSOC, and the timing and intensity of their occurrences can replace the simulator. The RL framework can work directly with the real data to solve the decision-making

problem without knowing the underlying probability distribution. However, in the event where a probability model is available that describes the information process, raw data can be used to fine tune the arrival process distribution of alerts due to uncertain events. The mean of the Poisson distributed arrival process within the simulator can be updated (see Algorithm 2 in the Electronic Appendix) and an offline learning could be continued using the simulator to maintain reality and for improved decisions. The emergent properties that are exhibited due to the interactions of the system with the uncertain environment are then studied through an RL framework to learn the state-action pairs that produce the best results.

3.2.3 Alert Analysis Process: Workday Simulation. A workday at the CSOC is simulated using the simulation algorithm presented in Algorithm 1 (see the Electronic Appendix). Each simulation run corresponds to one operation day of 24 hours. Alerts are generated using a Markovian distribution. Analysts are considered as resources, and they investigate alerts from a single queue of alerts populated by the IDSs in a first-come-first-served (FCFS) manner. The time taken to investigate an alert by an analyst, T , is the average time taken based on historical statistics observed in the organization. In this work, T is maintained constant except when a new alert pattern causes an increase in T , although it could also be drawn from a probabilistic distribution for each alert. It is assumed that all analysts spend 80% of their effort in a shift toward alert analysis, and the rest of the time is spent on report writing, training, and generating signatures. Hence, an analyst could increase his effort on alert analysis up to 20% when the need arises, which will increase the service rate of alerts investigated in a day. The alert analysis process of a CSOC is considered to be in steady state under normal operating conditions, which means that the average alert arrival rate per hour, average queue length, average waiting time in the queue, and average alert investigation time are all normal. Hence, a baseline avgTTA/hr value for the CSOC system can be established using Equation (1), and the LOE status is ideal as shown in Figure 1. A threshold value for avgTTA/hr is also established. The scheduled analyst staffing levels are adequate to maintain a predetermined acceptable avgTTA/hr (and LOE status) of the CSOC. Disruptive events (uncertainties) are simulated as described later in Table 2, and they affect the arrival and service rates as shown in the simulation block in Figure 2.

3.2.4 Performance Metrics. The avgTTA is recorded from the alert analysis process for each hour of simulation over the 14-day work cycle using Equation (1). The avgTTA/hr is used to compare the LOE performance of the CSOC between the RL-based LOE optimization model and both the greedy (myopic) and rule-based uniformly distributed resource models as shown in Sections 5.2 and 5.3 respectively. The baseline and threshold avgTTA/hr that were established under normal operating conditions are used to interpret the LOE status of the CSOC (Figure 1) at any point in time under the preceding modeling strategies.

3.3 RL-Based Optimization Model for Decision Making

The past-hour performance of the CSOC from the real-world alert analysis process (simulation block in this article) presents the avgTTA/hr and LOE status to the CSOC manager. Disruptive events, if any, from the past hour are known. It is imperative that the CSOC manager considers the current avgTTA/hr metric and LOE state of the system to make a decision to add additional resources or do nothing for the next hour of CSOC operation. The decision is nontrivial because of the uncertainties in the future disruptive events and the limited additional resources available to the CSOC manager. Accurate prediction of future disruptive events, such as a zero day attack, is very hard. However, one can observe from the history of past events, such as resources that were needed to mitigate a past disruptive event and the frequency of occurrences of each type of disruptive events, and build a probability distribution that can simulate the arrival process of

real-world uncertain events. By interacting with the unknown environment via simulation and by learning from past decisions, the goal of the RL-based decision support system is to optimally plan the allocation of additional resources such that in the long run (over several 14-day cycles), the CSOC system with an adaptive RL-based decision performs far better (in terms of its LOE) than making an ad hoc or greedy or a rule-based decision.

When a disruptive event occurs, a CSOC manager, in the order of preference as determined through our discussions with CSOC managers at the Army Research Lab, would utilize the remaining 20% of analyst time on alert analysis, spend some of their own time to assist the analysts in clearing the alert backlog, and bring on-call analysts to supplement the regular analyst workforce. The RL model presented in the following manages the additional resource allocation that follows the CSOC manager's order of preference and decides the quantity of additional resources and the timing of when to allocate the additional resources. Figure 2 shows the optimization block with the inputs to the RL agent and the RL decisions as its output. The following sections describe the optimization block in detail.

3.3.1 Inputs to the RL Agent. The three main inputs to the dynamic RL-based LOE optimization model are (1) the backlog observed at time t (at the t^{th} hour in a 14-day work cycle), (2) the number of hours left in the 14-day work cycle from current time t , and (3) the additional resource that is available for use between current time t and the end of the 14-day work cycle. At the beginning of each hour t , the backlog information is obtained from the end of the past hour $t - 1$ of the CSOC operation. Under normal operating conditions, there would be an average queue length of alerts that are awaiting investigation. The backlog is the number of alerts waiting in the queue for investigation, which exceeds the preceding average queue length of alerts. The alerts that have completed the investigation process in the past hour are removed from further analysis in this article, although some of them could be further processed as significant alerts (see Section A of the Electronic Appendix), which is beyond the scope of this work. Hence, the time to analyze an alert is kept fixed in this article for normal operating conditions; however, it could also be varied by drawing from a probabilistic service time distribution with a fixed mean value. One of the disruptive events that is studied here is an increase in the mean time to analyze alerts due to the detection of a new vulnerability, which is presented in Section 5.

3.3.2 Output From the RL Agent. The primary output of RL is the decision to allocate additional resources at time t for the 1-hour look-ahead period from t to $t + 1$. Every time a decision is made by the RL model, the available additional resources are updated and the decision (output of RL) is relayed into the simulation model with the CSOC manager's approval, as one of the CSOC inputs (see Figure 2). The additional resources that a CSOC manager can commit at time t are quantified in terms of additional alerts that could be analyzed between time t and $t + 1$. Therefore, when additional resources are added to the existing workforce, the service rate of alerts is increased proportional to the amount of additional resources being added. It must be noted that the available additional resources between time t and $t + 1$ is the sum of the number of additional alerts that can be serviced by a regular analyst workforce by working an additional 20% of their time on alert analysis, any number of additional alerts that can be serviced by managers, and the number of additional alerts that can be serviced by all available on-call analyst resources at time t . As stated earlier, the RL model will seek to allocate the additional resources in the preceding order of preference.

3.3.3 SDP Formulation. SDP models for dynamic resource allocation problems exploit the fact that for complex systems with no well-defined analytical models and no closed-form solutions, their evolving properties can be studied through their interactions with the environment. For the

cybersecurity LOE optimization problem, the decision at any time t is to allocate up to the available additional resource(s) (quantified as additional alerts that could be serviced), and the uncertainty is modeled by external and internal factors that interact with the dynamic alert analysis process. The generic SDP model involves defining the following elements: (1) system state variable B_t at time t , which are all recurrent states in a Markov chain; (2) decision variable g_t to quantify the amount of resource(s) allocated at t ; (3) exogenous information from a real process or a process simulator W_{t+1} , which is observed between t and $t + 1$ after decision g_t is taken; (4) state transition function $B_{t+1} = h(B_t, g_t, W_{t+1})$, where B_{t+1} is the next system state at $t + 1$, which depends on (B_t, g_t, W_{t+1}) ; (5) contribution function $C(B_t, g_t)$ that calculates the reward/cost for taking action g_t in state B_t ; (6) B_t^g and B_{t+1}^g (post-decision states at time t and $t + 1$, respectively), which are all recurrent states in a Markov chain; and (7) objective function that minimizes (cost) or maximizes (reward) the total contribution (value of a state $V(B_t)$) over a long period (infinite horizon) of time. In the context of LOE optimization, the preceding variables are defined as follows:

- (1) *System state $B_t \in B$* : This is a 3D vector $B_t = \{X_t, G_t, Y_t\}$, where X_t indicates the backlog of alerts at time t , G_t indicates the available capacity in terms of the number of additional alerts (additional resource) that can be investigated (serviced) between time t and $t + 1$ (additional work that can be done, in order of preference, by scheduled (regular) analysts, managers, and on-call analysts), and Y_t is the number of hours left between current time t and the end of the current 14-day work cycle. Time t is indexed at the end of each hour for this work.
- (2) *Decision g_t* : This is a 1D vector ($0 \leq g_t \leq G_t$) that indicates the resource allocation at time t , which is the number of additional alerts (additional resources) that can be investigated (serviced) between time t and $t + 1$ over and above the normal service rate in that time period. It is again emphasized that g_t has two components: the additional alerts that can be serviced by the scheduled (regular) analysts and managers, and the additional alerts that can be serviced by the on-call analyst workforce. The decision on the number of additional alerts that could be serviced is relayed into the simulation block.
- (3) *Post-decision system state B_t^g* : This is a 3D vector $B_t^g = \{X_t, G_t, Y_t\}^g$, where X_t indicates the backlog of alerts at time t , G_t indicates the updated available capacity in terms of the number of additional alerts (additional resource) that can be investigated (serviced) after decision g_t is taken, and Y_t is the number of hours left between current time t and the end of the current 14-day work cycle. The value of X_t and Y_t is the same for B_t and B_t^g .
- (4) *Exogenous information (uncertainty) W_{t+1}* : The uncertain events (see Table 2) that affect the alert analysis process is captured in the exogenous information. For example, between t and $t + 1$, the alert generation rate could go up or the service rate of alerts could be reduced due to a new vulnerability or analyst absenteeism.
- (5) *State transition function $B_{t+1} = h(B_t, g_t, W_{t+1})$* : This function defines how the next system state at time $t + 1$ is evolved. For B_{t+1} , the value of G_{t+1} is the same as the G_t in B_t^g . However, the new backlog information X_{t+1} from the alert analysis process and time left in the 14-day cycle Y_{t+1} are updated. In addition, state transition probabilities are not known due to the large number of states in the system. Hence, an RL-based approach is used for the dynamic optimization model.
- (6) *Contribution function $C(B_t, g_t)$* : The contribution function is the driving force behind the decision-making process. It measures how a decision g_t when taken at state B_t is rewarded or penalized. The contribution function of taking decision g_t in state B_t is calculated as follows. The most desirable state of the system at any time t is when the backlog is zero and the additional resources have not been used, regardless of how much time is left in the 14-

Fig. 3. Graphical representation of the contribution function.

day cycle. However, when additional resources are allocated, the available resource(s) for the future must be viewed with respect to the time left in the 14-day work cycle. Hence, the ratio of (G_t/Y_t) is calculated. Both (G_t/Y_t) and X_t are normalized on a 0 to 1 scale using r_t and f_t , respectively, as follows. At the beginning of the 14-day work cycle, the ratio is $\max(G_t)/\max(Y_t)$ whose value r_t on a 0 to 1 scale is $r_t = 1$, and any numerical value for (G_t/Y_t) that exceeds $\max(G_t)/\max(Y_t)$ is given a value of $r_t = 1$. All other numerical values for (G_t/Y_t) are normalized between $0 \leq r_t \leq 1$. The minimum value of Y_t is 1, which is the beginning of the last hour of the 14-day work cycle. If the backlog X_t is zero, then the value f_t (on a 0 to 1 scale) of the backlog at $X_t = 0$ is $f_t = 1$. A threshold value of a backlog of alerts is calculated using the threshold value of avgTTA/hr, and value f_t for $X_t \geq$ backlog threshold is $f_t = 0$. Hence, the most desirable value is $(f_t, r_t) = (1, 1)$, and the least desirable value is $(f_t, r_t) = (0, 0)$. Figure 3 provides a graphical representation of the contribution function.

The contribution function is given as follows:

$$C(B_t, g_t) = (w_1 * f_t) + (w_2 * r_t), \quad (2)$$

where w_1 and w_2 are weights. For this work, the weights are kept equal; however, the preceding formulation allows for weighing one factor, such as backlog, more than the other and vice versa. Clearly, if g_t is 0, then additional resources were not used at time t for the period from t to $t + 1$, and the ratio (G_t/Y_t) would increase as Y_t is reduced at time $t + 1$. Hence, a lower value of backlog X_t and a higher value of (G_t/Y_t) is a desirable post-decision state B_t^g , and such a decision g_t from state B_t that leads to a desirable B_t^g would have a high contribution function.

- (7) The objective function for the dynamic programming is measured as the long-run total discounted value of the states $V^j(B)$ as the iteration index $j \rightarrow \infty$, which is derived using the recursive Bellman's optimality Equation (3) shown in the following (Bellman 1957). $V^j(B)$ is a cumulative sum of discounted $C(B_t, g_t)$ for the learning phase whose iterations are indexed from 1 to j . It should be noted that index t is reset to 1 after every 14-day work cycle (336 hours). The learning phase goes through several iterations (indexed with j) of 14-day work cycles. Since the value of the state is measured in terms of the cumulative sum of discounted $C(B_t, g_t)$, the objective function for RL will be to maximize the long-run total discounted value of the states $V^j(B)$. In other words, the higher the value of $V^j(B)$, the better the system state. The dynamic programming algorithm will strive to move from one good state to another by making a decision under uncertainty, which is guided by the highest value of the future states that are reachable at any given time t .

3.3.4 Phases of SDP. To achieve the objective of this article, the SDP formulation proceeds in three phases: exploration, learning, and learned. The recursive Bellman's optimality equation that updates the value of the states is given as follows:

$$V^j(B_{t-1}^g) = (1 - \alpha^j)V^j(B_{t-1}^g) + \alpha^j \eta^j, \quad (3)$$

$$\eta^j = \left[\max_{g_t \in G_t} \{C(B_t, g_t) + \beta V^j(B_{t-1}^g)\} \right], \quad (4)$$

where α^j is the learning parameter that is decayed gradually over several iterations, j is the iteration index, G_t is the set of all feasible decisions from which the dynamic programming 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 g_t taken at the beginning of hour t from state B_t , the update of the value of post-decision state B_{t-1}^g from the last hour at $t - 1$ that had put the system into state B_t is executed after reaching state B_t^g as per Equation (3) (Powell 2007):

- (1) *Exploration phase:* In this phase, the dynamic programming algorithm would explore several nonoptimal decisions and acquire the value of system states that are visited. Equation (3) is executed, but without the max operator in Equation (4), by taking random decisions $0 \leq g_t \leq G_t$ on the number of additional alerts (additional resource) that can be serviced between time t and $t + 1$. Once the decision g_t on the number of additional alerts to be serviced between time t and $t + 1$ has been determined, the order of preference is used to identify the sources for the additional resource(s). First, the 20% of analyst time that is kept for report writing and other activities is diverted toward alert analysis. If this is inadequate, then some portion of the CSOC manager's time, if available, is diverted toward alert analysis. If this is still inadequate, then the on-call workforce is brought in. In summary, depending on RL model's decision, the available resource G_t is updated. The values of $V^j(B_t^g)$ in Equation (4) and $V^j(B_{t-1}^g)$ in Equation (3) are obtained from the previously stored values if the state was visited earlier, and otherwise the value is 0. Since the algorithm begins with all $V^0(B) = 0 \forall B$ at $j = 0$, exploration helps to populate the values of some of the states that are visited. Exploration is stopped after a certain number of iterations, which depends on the size of the state space, and the number of iterations planned for the learning phase.
- (2) *Learning phase:* In this phase, the dynamic programming algorithm would take (near-) optimal decisions at time t , which is obtained by executing the right side of Equation (4) with the max operator under the assumption that better estimates of the value of the states visited during exploration are available. Using η^j from Equation (4), the value of the previous post-decision state is updated at time t as per Equation (3). After several iterations, 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 in the Electronic Appendix (Powell 2007).
- (3) *Learned phase:* This is the implementation phase of the dynamic programming for LOE optimization. The inputs to this phase include the value of the states at the time when learning was terminated. In this phase, the dynamic programming algorithm would take near-optimal decisions at each time t , which is obtained from Equation (4) with the max operator under the assumption that very good estimates of the value of the most desirable states visited during learning are available. The dynamic programming algorithm will evaluate all of its feasible actions and will choose an action that takes the system to the post-decision state with the highest value of η^j in Equation (4) (the maximization problem).

Table 1. Inputs for the Baseline Case

Number of clusters of sensors	10
Average time between alert generation (s)	Expo(18.8)
Number of analysts	10
Percentage of effort of analysts toward alert analysis	80%
Average time taken to investigate an alert T (s)	15

4 Experimental Setup

This section presents the experimental setup used in this work. A nominal (baseline) case is set up first, wherein alerts from all sensors are investigated within a predetermined reasonable amount of time. The avgTTA/hr for the baseline case is acceptable to the organization, and the LOE of the CSOC is deemed to be ideal. Table 1 shows the inputs for the baseline case.

There are 10 clusters of sensors considered for the experiment. Typically, each cluster has about 10 to 12 sensors. Based on Equation (5) (see the Electronic Appendix for equations), the average arrival rate $\lambda = 1,919$ alerts per hour, and based on Equation (6), the average service rate $\mu = 1,920$ alerts per hour for the entire system ($U = 0.8$), which combines all sensors and analysts in the system. Hence, traffic intensity is $\rho < 1$ for the baseline case scenario, where $\rho = \lambda/\mu$. It should be noted that the service rate is based on 80% effort spent on alert analysis, and at this effort value, the alert service rate is slightly greater than the alert arrival rate (prevents the formation of an infinite queue length of alerts). A larger difference between λ and μ ($\lambda \ll \mu$) at $U = 0.8$ would suggest that the analysts have idle time. Hence, it is customary at the CSOC to set the value of λ close to μ at $U = 0.8$ ($\rho < 1$) and deal with $\rho \geq 1$ situations by utilizing (1) the remaining 20% of analysts' time on alert analysis, (2) portions of the manager's time, and (3) on-call analysts if needed. The preceding parameter values were obtained through discussions with the CSOC managers of both distributed and centralized CSOCs (Zimmerman 2014); however, different values of λ and μ could be used to set the baseline value of avgTTA/hr for a CSOC with $\rho < 1$. After a few days of CSOC operations, upon reaching a steady state under normal operating conditions, the acceptable value of avgTTA/hr was determined to be 1 hour and the nominal average queue length was 1,175 alerts at any given point in time. Any increase in queue length over its nominal average value would be considered a backlog, which affects the LOE status of the CSOC. The preceding values are considered to be the baseline values for the CSOC for the given values of U , λ , and μ , which establishes the ideal LOE status of the CSOC. It is important to set a baseline value for an acceptable LOE status for the CSOC operation to detect deviations in the LOE status and to initiate corrective actions.

Based on the various conversations with CSOC managers, an upper bound (threshold) on the value of avgTTA/hr was determined to be 4 hours. Different tolerance bands are created both below and above the threshold value of avgTTA/hr to indicate a color-coded representation of the LOE status as shown in Figure 1. Several experiments were conducted by increasing the alert generation rates, which established the average value for the backlog of alerts at 4,350 alerts that corresponds to the preceding 4-hour upper bound for the avgTTA/hr. Hence, in calculating the individual score for the backlog of alerts parameter X_t in the state of the system, a backlog of 3,175 (4,350 - 1,175) alerts or more was considered to have a value of $f_t = 0$ and the value for zero backlog of alerts (queue length of 1,175 alerts) was considered to be $f_t = 1$. The value for various backlogs of alerts in between 0 and 3,175 was linearly normalized. The various values for the ratio $(\max(G_t)/\max(Y_t))$ for resource-available G_t at time t and the time remaining in the 14-day cycle ($Y_t = 336 \text{ hours} - t$) were calculated by normalizing using an exponential distribution with a mean

of 85.7 (28,800/336) whose value r_t is 1. The values of the ratio above the mean would have $r_t = 1$, and those below are exponentially distributed between 0 and 1. The value of $\max(G_t)$ is calculated as follows. In this work, a total of 10 additional on-call analysts are considered to be available for the 2-week (14-day) period. These resources are a part of the surge support team and are required to provide support as soon as possible (within 1 hour) to the CSOC when the need arises. Each analyst can work up to 12 hours in the 2-week period. An on-call analyst (resource) is committed to 4 hours of support to the CSOC each time the on-call resource is engaged by the CSOC. With the inputs from Table 1, the total additional on-call resource available G_t translates to servicing a maximum of 28,800 (10 analysts * 12 hours * (3,600 seconds/15 seconds per alert)) additional alerts over a 2-week cycle. The values of w_1 and w_2 were set to be 0.5 to give equal importance to X_t and the ratio $(G_t)/(Y_t)$ in the calculation of the contribution function.

It should be noted that when the need for additional resources arises for the next hour of CSOC operation, the *first* default action by the greedy or rule-based or RL model is to allocate an additional 20% of analyst effort toward alert analysis, which is available every hour. This is because under normal operating conditions, only 80% of analyst time is allocated toward alert analysis. With the inputs in Table 1, the 20% additional analyst effort toward alert analysis translates to servicing an additional 480 alerts per hour by the current analysts in the shift. This may include contributions from resources such as the watch officers/managers, system administrators, or system analysts. The preceding additional resource expires every hour if unused. The *second* action by the greedy or rule-based or the RL model occurs only when there is a need for more additional resources above the 480 alerts per hour, for which the on-call resource as described previously is used. Hence, the figures in the next section show only the allocation of on-call resources, although the first default action is implemented but is not plotted.

The uncertainty, with a wide range of variability in the number of stochastic events, is modeled using a Poisson distribution with a mean of seven events in a 14-day (336-hour) period. Table 2 shows the various uncertain events that are considered for the experiments. These events are simulated at random, and they affect the alert generation and/or alert service rates for the CSOC.

5 EXPERIMENTS AND ANALYSIS OF RESULTS

This section presents the results of the LOE status of the CSOC over several 14-day runs for (1) a greedy approach that has no consideration of future uncertain events in its decision making, in which the backlog is allowed to accumulate until 4 hours worth of workload is accumulated before allocating additional resources, (2) a rule-based uniformly distributed approach in which the CSOC resource is evenly distributed over the 14-day period with the option to roll over any unused resource to the following day, and (3) the RL-based SDP approach for resource allocation. Both the rule-based uniformly distributed and the RL approaches include an uncertainty model that make the approaches more realistic than the greedy approach. In the absence of an intelligent approach, the rule-based uniform distribution of resource strategy serves as a good strategy for comparing the RL-based approach.

The dynamic RL model is run in three stages: exploration, exploitation (learning), and learned (implementation or validation). The dynamic programming algorithm (Algorithm 2 in the Electronic Appendix) was tested on several types of uncertain disruptive events using 50 simulation runs of the 14-day work cycles. In addition, for the same disruptive events in a 14-day work cycle, the greedy and rule-based approaches to decision making were implemented. In each instance of the simulation run, the alert analysis process was simulated using Algorithm 1; decisions for additional resource to handle the backlog of alerts were made by using the RL-based, rule-based, and greedy approaches; and alerts were analyzed over a 14-day period. Additionally, 95% confidence intervals were calculated for the LOE performance metrics.

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

Fig. 4. Temporal patterns in uncertain events: five events (a) and eight events (b).

5.1 Temporal Patterns in Uncertain Events

This section presents the various experiments performed to address unexpected conditions such as absenteeism in a shift, unexpected increase in alert generation rates, increase in alert investigation time caused by events such as the discovery of a new vulnerability, and a surge of alerts arising from a repaired communication link between the sensors and the CSOC (see Table 2). Figure 4 represents temporal patterns in uncertain events with five and eight events over a 14-day period. The plot shows the average number of additional alerts that were generated due to one of the events in Table 2 and the duration for which the event lasted (the event number is marked over the plot). For events that decrease the alert service rate, such as analyst absenteeism, or a new vulnerability that increases the alert investigation time, the equivalent increase in alert arrival rate is calculated by maintaining the same service rate.

5.2 Case Study of Five Uncertain Events

This section presents the results from an experiment in which five uncertain disruptive events occurred over the 14-day work cycle as shown in Figure 4(a). It is reemphasized that in the case of an occurrence of an uncertain event, the additional 20% effort for alert analysis, which is available during every hour of the shift from resources such as analysts and watch officers is utilized first. The on-call analysts are called upon only if needed as a second source of additional resource. Figure 5(a) through (c) show only the depletion of on-call additional resources. Wherever the plot is horizontal, it means that the on-call resource was not utilized because either (1) the backlog was 0 (average queue length is 1,175 alerts as in the baseline case) or (2) the backlog was low and was cleared by adding the additional 20% analyst effort for alert analysis.

In the greedy approach as shown in Figure 5(a), the additional resources were utilized in a myopic manner. An additional resource is called upon only when the workload that is worth the resource's time is accumulated. Such a strategy is commonly employed in various organizations where on-call resources are limited and expensive. For example, an analyst is called upon as soon as 4 hours of workload is accumulated. As shown in Figure 5(g), since there is a waiting time for

Fig. 5. Five uncertain events: available additional desources (a–c), backlog (d–f), and AvgTTA (LOE) (g–i).

the 4 hours of workload to accumulate, the LOE (and the alert backlog as shown in Figure 5(d)) is observed to climb into the yellow zone. As soon as the additional resources are assigned, the LOE is restored into the green zone. However, since this strategy is also myopic (greedy), it can be observed from Figure 5(g) that there were no additional resources left after 10 days, and the LOE climbed into the red zone on the 11th day.

In the rule-based uniformly distributed approach, the following rules are followed. The CSOC's additional resources are evenly distributed at the start of the 14-day period. As the days progress, any unused resource is rolled over into the following day. Resources allocated to future days cannot be used in advance, which sharply differs from the greedy approach that can exhaust as many resources as needed. The decision to allocate additional resources depends on the available resource on that day, and it is a reaction to the magnitude of the uncertain event that occurs. Figure 5(b) shows that there are unused resources at the end of the 14-day period because resources were evenly distributed and no major event occurred toward the end of the 14 days that consumed all of the remaining resources. Due to the rule that future resources could not be used in advance because they are reserved for future uncertainties, the LOE was found to have higher variance (see Figure 5(h)) than the greedy and RL approaches as shown in Figures 5(g) and 5(i), respectively. Despite being better than greedy in reacting to the uncertainties, the LOE eventually crosses the red band (4-hour avgTTA threshold) with the onset of the 4th uncertain event.

There are two critical decisions learned with RL: (1) when to call the additional resources with respect to the time available in the 14-day work cycle and (2) how many additional resources to call upon such that an optimal LOE is maintained over the 14-day work cycle. It can be seen from Figure 5(c) that very few resources were utilized until the 4th event on the 10th day as compared to Figure 5(a) and (b), in which the additional on-call resources were exhausted. As a result, there were fluctuations in the avgTTA values, but the LOE was maintained in the green zone. With the event on the 10th day ($t = 240$), the majority of additional on-call resources were called upon to keep the LOE in the green zone. With resources still remaining at the end of the 14-day work cycle as shown in Figure 5(c), it can be observed from Figure 5(i) that the LOE was maintained in the green zone throughout the 14-day work cycle.

Fig. 6. Eight uncertain events: available additional resources (a–c), backlog (d–f), and AvgTTA (LOE) (g–i).

5.3 Case Study of Eight Uncertain Events

This section presents the results from an experiment in which eight uncertain disruptive events occurred over the 14-day work cycle as shown in Figure 4(b). The greedy strategy had its additional on-call resource depleted in 6 days (Figure 6(a)), and the LOE reached the red zone in the first 6 days of the 14-day work cycle (see Figure 6(g)). The rule-based uniformly distributed approach performed better than the greedy approach in terms of LOE and the ability to return the system back into the green zone (see Figure 6(h)). However, the LOE reached the red zone (4-hour avgTTA threshold) twice due to the inability of the approach to use future resources in advance.

In the RL-based approach, with more events in the first 6 days of the 14-day work cycle, there were some additional resources that were called upon to maintain the LOE in the green zone. However, the resource depletion shown in Figure 6(c) was not as steep as in the greedy case presented in Figure 6(a), and unlike the greedy case, there were leftover on-call resources at the end of the 14-day work cycle. With the discovery of a new vulnerability that reduced the throughput toward the end of the 14-day work cycle, the LOE was moved into the yellow zone (Figure 6(i)) for a few hours before being brought back into the green zone with additional resource allocation. In summary, the LOE never reached the orange or red zone. Similar results were noticed in the other 50 simulation runs of 14-day work cycles.

5.4 Discussion

One of the metaprinciples that were highlighted by the preceding study was to determine the optimal strategy for making the decision on when to increase the alert service rate, and by how much to increase it, which depends on the availability of resources at the CSOC, time left in the current 14-day work cycle, and its current LOE status. This is very critical because the analyst resource must be managed effectively when the CSOC faces several uncertain events that increase the avgTTA/hr. Given the limited nature of the analyst resource, the resource must be conserved in anticipation of future needs. At the same time, adequate action must be taken to address any reduction in LOE (increase in avgTTA/hr) of the CSOC. Hence, it is unwise to use all of the CSOC resources to reduce avgTTA/hr to its lowest value, and likewise it is unwise to let the avgTTA/hr

increase close to its threshold value. The preceding motivates the need for an intelligent and optimal decision-making strategy that maintains the LOE at a reasonable value during a given operational cycle of a CSOC (typically 14-day cycles) while managing its analyst resource effectively.

Clearly, the preceding sample result indicates that the greedy model cannot adapt to the uncertainty in disruptive events. When subjected to the same uncertain events, and the same amount of additional resources, the greedy approach allocated the limited additional on-call resource with a myopic vision that resulted in the rapid depletion of resources, which in turn caused the LOE of the CSOC to deteriorate toward the end of the 14-day work cycle when new disruptive events occurred. The rule-based uniformly distributed approach performed better than the greedy one; however, it lacked the intelligence to move resources around for making better decisions in the presence of uncertainties. The dynamic RL-based additional resource allocation model, which had learned from the uncertain events in the past, made an intelligent allocation of the additional resources such that the LOE was optimized throughout the 14-day work cycle. Thus, in the preceding example studies, the RL model with a long-term view in decision making prevented the potential danger to run out of the on-call analyst workforce toward the later part of the 14-day work cycle when there was a critical need for additional analysts due to more disruptive events. The algorithm discussed in this article can be easily implemented on a scaled version of the problem in an organization with more clusters (sensors), which may require more computing resources. The objective of the experiment performed was to provide a proof of concept using the RL model for making optimal decisions by incorporating real-world adverse conditions faced by a CSOC. More details on computational complexity are available in the Electronic Appendix.

6 CONCLUSION AND FUTURE WORK

This work developed an intelligent decision-making tool to support the CSOC manager in allocating the limited amount of additional resources over and beyond the normal resources available in a shift to mitigate the adverse effects of uncertain events (external or internal) on the LOE status of the CSOC. The decision-making tool uses artificial intelligence that was gained using the RL framework to learn optimal decisions for a given state of available resources, time left in the current work cycle, and the LOE status of the CSOC in a 14-day work cycle run. Hence, the intelligent tool developed in this work will optimize the manager's decision of when to increase the CSOC service rate and by how much when analyst resources are limited. A real-time avgTTA/hr monitoring system (software development) will allow the manager of a CSOC to observe the impact of the RL-based decision-making tool, which uses the avgTTA metric as a means to measure and monitor the LOE of a CSOC. Several questions can be answered by the dynamic avgTTA/hr monitoring and control framework. For instance, a CSOC manager could perform offline analysis to determine the impact of new uncertain events and test the efficacy of the RL model to optimize the LOE of the CSOC. Likewise, model parameters such as number of sensors or number of analysts could be changed for a CSOC, and the preceding RL model can be learned offline to make near-optimal decisions for the CSOC manager.

In this article, we have used a rule-based baseline model that evenly divides the available resources. In future work, it would be interesting to compare our method to a predictive baseline model that estimates the additional workload and spreads the resources accordingly. However, there are two challenges that must be overcome in constructing such a baseline. First, the fact that a cybersecurity environment is a dynamic environment with significant incomplete information and uncertainties makes it difficult to accurately predict the additional workload under complex scenarios (with large state and action spaces). Second, given a large state/action space and a complex system, it is not trivial to determine how much effort (resources) to use such that the LOE is maintained in the good zones throughout the time period. Another direction for future research

is to allow for allocating additional resources to investigate a false-negative alert (a real threat that was first incorrectly identified as innocuous) while minimizing false-positive alerts (minimize resource wastage). Dynamic monitoring of situational awareness and providing continuous decisions to CSOC managers for dynamic control of the LOE of a CSOC over 14-day work cycle runs is a paradigm shift in CSOC operations, which could benefit from the implementation of the preceding RL-based intelligent decision support tool developed in this work.

ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Cliff Wang of the Army Research Laboratory for suggesting this problem to us.

REFERENCES

- M. E. Aydin and E. Oztemel. 2000. Dynamic job-shop scheduling using reinforcement learning agents. *Robotics and Autonomous Systems* 33, 2, 169–178.
- Daniel Barbará and Sushil Jajodia (Eds.). 2002. *Application of Data Mining in Computer Security*. Advances in Information Security, Vol. 6. Springer.
- Richard Bellman. 1957. *Dynamic Programming*. Princeton University Press, Princeton NJ.
- Sandeep Bhatt, Pratyusa K. Manadhata, and Loai Zomlot. 2014. The operational role of security information and event management systems. *IEEE Security and Privacy* 12, 5, 35–41.
- David Botta, Kasia Muldner, Kirstie Hawkey, and Konstantin Beznosov. 2011. Toward understanding distributed cognition in IT security management: The role of cues and norms. *Cognition, Technology and Work* 13, 2, 121–134.
- CIO. 2008. *DON Cyber Crime Handbook*. U.S. Department of the Navy, Washington, DC.
- Roberto Di Pietro and Luigi V. Mancini (Eds.). 2008. *Intrusion Detection Systems*. Advances in Information Security, Vol. 38. Springer.
- Benjamin Edwards, Steven Hofmeyr, and Stephanie Forrest. 2016. Hype and heavy tails: A closer look at data breaches. *Journal of Cybersecurity* 2, 1, 3–14.
- Steven M. Furnell, Nathan Clarke, Rodrigo Werlinger, Kasia Muldner, Kirstie Hawkey, and Konstantin Beznosov. 2010. Preparation, detection, and analysis: The diagnostic work of IT security incident response. *Information Management and Computer Security* 18, 1, 26–42.
- Rajesh Ganesan, Sushil Jajodia, and Hasan Cam. 2017. Optimal scheduling of cybersecurity analyst for minimizing risk. *ACM Transactions on Intelligent Systems and Technology* 8, 4, Article 52, 33 pages.
- Rajesh Ganesan, Sushil Jajodia, Ankit Shah, and Hasan Cam. 2016. Dynamic scheduling of cybersecurity analysts for minimizing risk using reinforcement learning. *ACM Transactions on Intelligent Systems and Technology* 8, 1, Article 4, 21 pages.
- Abhijit Gosavi. 2003. *Simulation Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*. Kluwer Academic, Norwell, MA.
- Donald Gross, John Shortle, James Thompson, and Carl Harris. 2008. *Fundamentals of Queueing Theory*. Wiley InterScience, New York, NY.
- Jonathan E. Helm, Shervin AhmadBeygi, and Mark P. Van Oyen. 2011. Design and analysis of hospital admission control for operational effectiveness. *Production and Operations Management* 20, 4, 359–374.
- Daniel P. Heyman and Matthew J. Sobel. 2004. *Stochastic Models in Operations Research: Stochastic Processes and Operating Characteristics*. Vol. 1. Dover Publications.
- Marshall Kuypers and Elisabeth Paté-Cornell. 2016. *Department of Energy Cyber Security Incidents*. Center for International Security and Cooperation, Stanford, CA. https://cisac.fsi.stanford.edu/sites/default/files/doe_cyber_security_incidents.pdf.
- Stephen Northcutt and Judy Novak. 2002. *Network Intrusion Detection (3rd ed.)*. New Riders Publishing, Thousand Oaks, CA.
- Warren B. Powell. 2007. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley InterScience.
- Martin L. Puterman. 1994. *Markov Decision Processes*. Wiley InterScience, New York, NY.
- Stephen Reilly. 2015. *Records: Energy Department struck by cyber attacks*. *USA Today*. Retrieved March 16, 2018, from <https://www.usatoday.com/story/news/2015/09/09/cyber-attacks-doe-energy/71929786/>.

- Ankit Shah, Rajesh Ganesan, Sushil Jajodia, and Hasan Cam. 2017. A methodology to measure and monitor level of operational effectiveness of a CSOC. *International Journal of Information Security* 17, 2, 121–134.
- Satinder Singh and Dimitri Bertsekas. 1997. Reinforcement learning for dynamic channel allocation in cellular telephone systems. In *Advances in Neural Information Processing Systems 9 (NIPS'96)*. 974–980.
- Matthew Smith and Elisabeth Paté-Cornell. 2017. 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'17)*. 37–56.
- V. S. Subrahmanian, M. Ovelgonne, T. Dumitras, and A. Prakash. 2015. *The Global Cyber-Vulnerability Report*. Springer.
- Sathya Chandran Sundaramurthy, John McHugh, Xinming Ou, Michael Wesch, Alexandru G Bardas, and S. Raj Rajagopalan. 2016. Turning contradictions into innovations or: How we learned to stop whining and improve security operations. In *Proceedings of the 12th Symposium on Usable Privacy and Security (SOUPS'16)*.
- Richard Sutton and Andrew G. Barto. 1998. In *Reinforcement Learning*. MIT Press, Cambridge, MA.
- Pieter Vansteenwegen and Dirk Van Oudheusden. 2007. Decreasing the passenger waiting time for an intercity rail network. *Transportation Research Part B: Methodological* 41, 4, 478–492.
- W. Wonham. 1979. *Linear Multivariable Control: A Geometric Approach*. Faller-Verlag.
- Stephen J. Zaccaro, Reeshad S. Dalal, Lois E. Tetrack, and Julie A. Steinke. 2016. *Psychosocial Dynamics of Cyber Security*. Routledge.
- Carson Zimmerman. 2014. *The Strategies of a World-Class Cybersecurity Operations Center*. MITRE Corporation, McLean, VA.

Received August 2017; revised October 2017; accepted December 2017