Optimal Thresholds for Anomaly-Based Intrusion Detection in Dynamical Environments

Amin Ghafouri, Waseem Abbas, Aron Laszka,
Yevgeniy Vorobeychik, and Xenofon Koutsoukos

Institute for Software Integrated Systems,
Vanderbilt University, USA
`{firstname.lastname}@vanderbilt.edu`

Abstract. In cyber-physical systems, malicious and resourceful attackers could penetrate a system through cyber means and cause significant physical damage. Consequently, early detection of such attacks becomes integral towards making these systems resilient to attacks. To achieve this objective, intrusion detection systems (IDS) that are able to detect malicious behavior early enough can be deployed. However, practical IDS are imperfect and sometimes they may produce false alarms even for normal system behavior. Since alarms need to be investigated for any potential damage, a large number of false alarms may increase the operational costs significantly. Thus, IDS need to be configured properly, as oversensitive IDS could detect attacks very early but at the cost of a higher number of false alarms. Similarly, IDS with very low sensitivity could reduce the false alarms while increasing the time to detect the attacks. The configuration of IDS to strike the right balance between time to detecting attacks and the rate of false positives is a challenging task, especially in dynamic environments, in which the damage caused by a successful attack is time-varying.

In this paper, using a game-theoretic setup, we study the problem of finding optimal detection thresholds for anomaly-based detectors implemented in dynamical systems in the face of strategic attacks. We formulate the problem as an attacker-defender security game, and determine thresholds for the detector to achieve an optimal trade-off between the detection delay and the false positive rates. In this direction, we first provide an algorithm that computes an optimal *fixed threshold* that remains fixed throughout. Second, we allow the detector's threshold to change with time to further minimize the defender's loss, and we provide a polynomial-time algorithm to compute time-varying thresholds, which we call *adaptive thresholds*. Finally, we numerically evaluate our results using a water-distribution network as a case study.

Keywords: cyber-physical systems, security, game theory, intrusion detection system

1 Introduction

In recent years, we have seen an increasing trend of malicious intruders and attackers penetrating into various cyber-physical systems (CPS) through cyber means and causing severe physical damage. Examples of such incidents include the infamous Stuxnet worm [11], cyber attack on German steel plant [15], and Maroochy Shire water-services incident [1] to name a few. To maximize the damage, attackers often aim to remain covert and avoid getting detected for an extended duration of time. As a result, it becomes crucial for a defender to design and place efficient intrusion and attack detection mechanisms to minimize the damage. While attackers may be able to hide the specific information technology methods used to exploit and reprogram a CPS, they cannot hide their final intent: the need to cause an adverse effect on the CPS by sending malicious sensor or controller data that will not match the behavior expected by an anomaly-based detection system [7]. Anomaly-based detection systems incorporate knowledge of the physical system, in order to monitor the system for suspicious activities and cyber-attacks. An important design consideration in such detection systems is to carefully configure them in order to satisfy the expected monitoring goals.

A well-known method for anomaly-based detection is sequential change detection [9]. This method assumes a sequence of measurements that starts under the normal hypothesis and then, at some point in time, it changes to the attack hypothesis. Change detection algorithm attempts to detect this change as soon as possible. In a sequential change detection, there is a *detection delay*, that is, a time difference between when an attack occurs and when an alarm is raised. On the other hand, detection algorithms may induce *false positives*, that is, alarms raised for normal system behavior. In general, it is desirable to reduce detection delay as much as possible while maintaining an acceptable false positive rate. Nevertheless, there exists a trade-off between the detection delay and the rate of false positives, which can be controlled by changing the sensitivity of the the detector. A typical way to control detector sensitivity is through a detection threshold: by decreasing (increasing) detection threshold, a defender can decrease (increase) detection delay and increase (decrease) false positive rate. Consequently, the detection threshold must be carefully selected, since a large value may result in excessive losses due to high detection delays, while a small value may result in wasting operational resources on investigating false alarms.

Finding an *optimal threshold*, that is, one that optimally balances the detection delay-false positive trade-off, is a challenging problem [12]. However, it becomes much more challenging when detectors are deployed in CPS with dynamic behavior, that is, when the expected damage incurred from undetected cyber-attacks depends on the system state and time. As a result, an attack on a CPS which is in a critical state is expected to cause more damage as compared to an attack in a less critical state. For example, in water distribution networks and electrical grids, disruptions at a high-demand time are more problematic than disruptions at a low-demand time. Hence, defenders need to incorporate time-dependent information in computing optimal detection thresholds when facing strategic attackers.

We study the problem of finding optimal detection thresholds for anomaly-based detectors implemented in dynamical systems in the face of strategic attacks. We model rational attacks against a system that is equipped with a detector as a two-player game between a defender and an attacker. We assume that an attacker can attack a system at any time. Considering that the damage is time-dependent, the attacker's objective is to choose the optimal time to launch an attack to maximize the damage incurred. On the other hand, the defender's objective is to select the detection thresholds to detect an attack with minimum delay while maintaining an acceptable rate of false positives. To this end, first we present an algorithm that selects an optimal threshold for the detector that is independent of time (i.e., *fixed*). We call it as a *fixed threshold* strategy. Next, we allow the defender to select a time-varying threshold while associating a cost with the threshold change. For this purpose, we present a polynomial time algorithm that computes thresholds that may depend on time. We call this approach the *adaptive threshold* strategy. We present a detailed analysis of the computational complexity and performance of both the fixed and adaptive threshold strategies. Finally, we evaluate our results using a water distribution system as a case study. Since expected damage to the system by an attack is time-dependent, the adaptive threshold strategy achieves a better overall detection delay-false positive trade-off, and consequently minimize the defender's losses. Our simulations indicate that this is indeed the case, and adaptive thresholds outperform the fixed threshold.

The remainder of this paper is organized as follows. In Section 2, we introduce our system model. In Section 3, we present our game-theoretic model and define optimal fixed and adaptive detection thresholds. In Section 4, we analyze both strategies and present algorithms to obtain optimal fixed and adaptive thresholds. In Section 5, we evaluate these algorithms using numerical example. In Section 6, we discuss related work on detection threshold selection in the face of strategic attacks. Finally, we offer concluding remarks in Section 7.

2 System Model

In this section, we present the system model. For a list of symbols used in this paper, see Table 1.

2.1 Attack Model

Let the system have a finite discrete time horizon of interest denoted by $\{1, ..., T\}$. Adversaries may exploit threat channels by compromising the system through a deception attack that starts at time k_a and ends at k_e, thus spanning over the interval $[k_a, k_e]$. Deception attacks are the ones that result in loss of integrity of sensor-control data, and their corresponding danger is especially profound due to the tight coupling of physical and cyber components (see [5] for details). If an attack remains undetected, it will enable the attacker to cause physical or financial damage. In order to represent the tight relation between the CPS's dynamic behavior and the expected loss incurred from undetected attacks, we model the potential damage of an attack as a function of time.

Table 1. List of Symbols

Symbol	Description
$\mathcal{D}(k)$	expected damage caused by an attack at timestep k
$\delta(\eta)$	expected detection delay given detection threshold η
$FP(\eta)$	false positive rate given detection threshold is η
C	cost of false alarms
$\mathcal{P}(\eta, k_a)$	attacker's payoff for threshold η and attack time k_a
$\mathcal{L}(\eta, k_a)$	defender's loss for threshold η and attack time k_a
Adaptive Threshold	
$\mathcal{P}(\boldsymbol{\eta}, k_a)$	attacker's payoff for adaptive threshold $\boldsymbol{\eta} = \{\eta_k\}$ and attack time k_a
$\mathcal{L}(\boldsymbol{\eta}, k_a)$	defender's loss for adaptive threshold $\boldsymbol{\eta} = \{\eta_k\}$ and attack time k_a

Definition 1. *(Expected Damage Function): Damage function of a CPS is a function $\mathcal{D}: \{1, ..., T\} \to \mathbb{R}_+$, which represents the expected damage $\mathcal{D}(k)$ incurred to a system from an undetected attack at time $k \in \{1, ..., T\}$.*

The definition above describes instant damage at a time $k \in \{1, ..., T\}$. Following this definition, expected total damage resulting from an attack that spans over some interval is defined as follows.

Definition 2. *(Expected Total Damage): Expected total damage is denoted by a function $\bar{\mathcal{D}} : \{1, ..., T\} \times \{1, ..., T\} \to \mathbb{R}_+$, which represents the expected total damage $\bar{\mathcal{D}}(k_a, k_e)$ incurred to a system from an undetected attack in a period $[k_a, k_e]$. Formally,*

$$\bar{\mathcal{D}}(k_a, k_e) = \sum_{k=k_a}^{k_e} \mathcal{D}(k) \ . \tag{1}$$

2.2 Detector

We consider a defender whose objective is to protect the physical system, which is equipped with a detector. The detector's goal is to determine whether a sequence of received measurements generated through the system corresponds to the normal behavior or an attack. To implement a detection algorithm, we utilize a widely used method known as sequential change detection [9]. This method assumes a sequence of measurements that starts under the normal hypothesis, and then, at some point in time, changes to the attack hypothesis. Change detection algorithm attempts to detect this change as soon as possible.

Example (CUSUM). The Cumulative sum (CUSUM) is a statistic used for change detection. The nonparametric CUSUM statistic $S(k)$ is described by

$$S(k) = (S(k-1) + z(k))^+,$$

where $S(0) = 0$, $(a)^+ = a$ if $a \geq 0$ and zero otherwise, and $z(k)$ is generated by an observer, such that under normal behavior it has expected value of less than zero [7]. Assigning η as the detection threshold chosen based on a desired false alarm rate, the corresponding decision rule is defined as

$$d(S(k)) = \begin{cases} \text{Attack} & \text{if } S(k) > \eta \\ \text{Normal} & \text{otherwise} \end{cases}$$

Detection Delay and False Positive Rate. In detectors implementing change detection, there might be a *detection delay*, which is the time taken by the detector to raise an alarm since the occurrence of an attack.[1] Further, there might be a *false positive*, which means raising an alarm when the system exhibits normal behavior. In general, it is desirable to reduce detection delay as much as possible while maintaining an acceptable false positive rate. But, there exists a trade-off between the detection delay and the rate of false positives, which can be controlled by changing the detection threshold. In particular, by decreasing (increasing) the detection threshold, a defender can decrease (increase) detection delay and increase (decrease) false positive rate. Finding the optimal trade-off point and its corresponding *optimal threshold* is known to be an important problem [12], however, it is much more important in CPS, since expected damage incurred from undetected attack directly depends on detector's performance.

We represent detection delay by the continuous function $\delta : \mathbb{R}_+ \to \mathbb{N} \cup \{0\}$, where $\delta(\eta)$ is the detection delay (in timesteps) when threshold is η. Further, we denote the attainable false positive rate by the continuous function $FP : \mathbb{R}_+ \to [0, 1]$, where $FP(\eta)$ is the false positive rate when the detection threshold is η. We assume that δ is increasing and FP is decreasing, which is true for most typical detectors including the CUSUM detector.

3 Problem Statement

In this section, we present the optimal threshold selection problem. We consider two cases: 1) Fixed threshold, in which the defender selects an optimal threshold and then keeps it fixed; and 2) Adaptive threshold, in which the defender changes detection threshold based on time. We model this problems as conflicts between a defender and an attacker, which are formulated as two-player Stackelberg security games.

3.1 Fixed Threshold

Strategic Choices. The defender's strategic choice is to select a detection threshold η. The resulting detection delay and false positive rate are $\delta(\eta)$ and $FP(\eta)$, respectively. We consider the worst-case attacker that will not stop the

[1] Note that any desired definition of detection delay may be considered, for example, stationary average delay [18, 19].

attack before detection in order to maximize the damage. Consequently, the attacker's strategic choice becomes to select a time k_a to start the attack. Note that we consider damage from only undetected attacks since the mitigation of non-stealthy attacks is independent of detector.

Defender's Loss and Attacker's Payoff. As an alarm is raised, the defender needs to investigate the system to determine whether an attack has actually happened, which will cost him C. When the defender selects threshold η and the attacker starts its attack at a timestep k_a, the defender's loss (i.e., inverse payoff) is

$$\mathcal{L}(\eta, k_a) = C \cdot FP(\eta) \cdot T + \sum_{k=k_a}^{k_a+\delta(\eta)} \mathcal{D}(k) \ , \tag{2}$$

that is, the amount of resources wasted on manually investigating false positives and the expected amount of damage caused by undetected attacks.

For the strategies (η, k_a), the attacker's payoff is

$$\mathcal{P}(\eta, k_a) = \sum_{k=k_a}^{k_a+\delta(\eta)} \mathcal{D}(k) \ . \tag{3}$$

that is, the total damage incurred to the system prior to the expected detection time. The idea behind this payoff function is the assumption of a worst-case attacker that has the goal of maximizing the damage.

Best-Response Attack and Optimal Fixed Threshold. We assume that the attacker knows the system model and defender's strategy, and can thus compute the detection threshold chosen by the defender. Hence, the attacker will play a *best-response* attack to the defender's strategy, which is defined below.

Definition 3. *(Best-Response Attack): Taking the defender's strategy as given, the attacker's strategy is a best-response if it maximizes the attacker's payoff. Formally, an attack starting at k_a is a best-response attack given a defense strategy η, if it maximizes $\mathcal{P}(\eta, k_a)$.*

Further, the defender must choose his strategy expecting that the attacker will play a best-response. We formulate the defender's optimal strategy as strong Stackelberg equilibrium (SSE) [10], which is commonly used in the security literature for solving Stackelberg games.

Definition 4. *(Optimal Fixed Threshold): We call a defense strategy optimal if it minimizes the defender's loss given that the attacker always plays a best-response. Formally, an optimal defense is*

$$\underset{\substack{\eta, \\ k_a \in bestResponses(\eta)}}{\arg\min} \quad \mathcal{L}(\eta, k_a), \tag{4}$$

where bestResponses(η) is the set of best-response attacks against η.

3.2 Adaptive Threshold

Although the optimal fixed threshold minimizes the defender's loss considering attacks at critical periods (i.e., periods with high damage), it imposes a high false alarm rate at less critical periods. Adaptive threshold strategies directly address this issue. The idea of adaptive threshold is to reduce the detector's sensitivity during less critical periods (via increasing the threshold), and increase the sensitivity during more critical periods (via decreasing the threshold). As it will be shown, this significantly decreases the loss corresponding to false alarms. However, the defender may not want to continuously change the threshold, since a threshold change requires a reconfiguration of the detector that has a cost. Hence, the rational defender needs to find an *optimal adaptive threshold*, which is a balance between continuously changing the threshold and keeping it fixed.

The adaptive threshold is defined by $\boldsymbol{\eta} = \{\eta_k\}_{k=1}^{T}$. The number of threshold changes is described by $N = |S|$, where $S = \{k \mid \eta_k \neq \eta_{k+1}, k \in \{1, ..., T-1\}\}$. If the system is under an undetected attack, the detection delay for each timestep k is the delay corresponding to its threshold, i.e., $\delta(\eta_k)$. We define detection time of an attack k_a as the time index at which the attack is first detected. It is given by

$$\sigma(\boldsymbol{\eta}, k_a) = \{\min k \mid \delta(\eta_k) \leq k - k_a\} \ . \tag{5}$$

Note that the equation above represents the time index at which the attack is first detected, and not the detection delay. The detection delay for an attack k_a can be obtained by $\delta(\boldsymbol{\eta}, k_a) \coloneqq \sigma(\boldsymbol{\eta}, k_a) - k_a$.

Strategic Choices. The defender's strategic choice is to select the threshold for each time index, given by $\boldsymbol{\eta} = \{\eta_1, \eta_2, ..., \eta_T\}$. We call $\boldsymbol{\eta}$ to be the set of adaptive threshold. Since we consider a worst-case attacker that will not stop the attack before detection, the attacker's strategic choice is to select a time k_a to start the attack.

Defender's Loss and Attacker's Payoff Let C_d be the cost associated with each threshold change. When the defender selects adaptive threshold $\boldsymbol{\eta}$, and the attacker starts its attack at a timestep k_a, the defender's loss is

$$\mathcal{L}(\boldsymbol{\eta}, k_a) = N \cdot C_d + \sum_{k=1}^{T} C \cdot FP(\eta_k) + \sum_{k=k_a}^{\sigma(\boldsymbol{\eta}, k_a)} \mathcal{D}(k) \ , \tag{6}$$

that is, the amount of resources spent on changing the threshold, operational costs of manually investigating false alarms, and the expected amount of damage caused by undetected attacks.

For the strategies $(\boldsymbol{\eta}, k_a)$, the attacker's payoff is the total damage prior to the expected detection time,

$$\mathcal{P}(\boldsymbol{\eta}, k_a) = \sum_{k=k_a}^{\sigma(\boldsymbol{\eta}, k_a)} \mathcal{D}(k) \ . \tag{7}$$

Best-Response Attack and Optimal Adaptive Threshold. The definitions presented in this part are analogous to the ones discussed above for the case of optimal fixed threshold. We assume the attacker can compute the adaptive threshold, and will play a *best-response* to the defender's strategy, as defined below.

Definition 5. *(Best-Response Attack): Taking the defender's strategy as given, the attacker's strategy is a best-response if it maximizes the attacker's payoff. Formally, an attack* k_a *is a best-response given a defense strategy* $\boldsymbol{\eta}$*, if it maximizes* $\mathcal{P}(\boldsymbol{\eta}, k_a)$ *as defined in* (7)*.*

Further, the defender must choose its strategy expecting that the attacker will play a best-response.

Definition 6. *(Optimal Adaptive Threshold): We call a defense strategy* optimal *if it minimizes the defender's loss given that the attacker always plays a best-response with tie-breaking in favor of the defender. Formally, an optimal defense is*

$$\operatorname*{arg\,min}_{\substack{\boldsymbol{\eta},\\ k_a \in bestResponses(\boldsymbol{\eta})}} \quad \mathcal{L}(\boldsymbol{\eta}, k_a), \tag{8}$$

where bestResponses$(\boldsymbol{\eta})$ *is the best-response attack against* $\boldsymbol{\eta}$*.*

4 Selection of Optimal Thresholds

In this section, we present polynomial-time algorithms to compute optimal thresholds, both for the fixed and adaptive cases.

4.1 Fixed Threshold

To compute an optimal fixed threshold, we present Algorithm 1. Here, we consider that any detection delay can be achieved by selecting a specific threshold value. Therefore, the algorithm finds an optimal detection delay, from which the optimal threshold value can be selected. To find the optimal detection delay, the algorithm iterates through all possible values of detection delay and selects the one that minimizes the defender's loss considering a best-response attack. To find a best-response attack k_a, given a delay δ, the algorithm iterates through all possible values of k_a, and selects the one that maximizes the payoff.

Proposition 1. *Algorithm 1 computes an optimal fixed threshold in* $\mathcal{O}(T^2)$ *steps.*

Proof. The obtained threshold is optimal since the algorithm evaluates all possible solutions through exhaustive search. Given a pair (δ, k_a), when computing the attacker's payoff $P(\delta, k_a)$ in Line 6, we use the payoff computed in previous iteration, and write $P(\delta, k_a) = P(\delta, k_a - 1) + \mathcal{D}(k_a - 1) + \mathcal{D}(k_a + \delta)$, which takes constant time. Therefore, the running time of the algorithm is subquadratic in the total number of timesteps T. □

Algorithm 1 Algorithm for Optimal Fixed Threshold

```
Input $\mathcal{D}(k)$, $T$, $C$
Initialize: $\delta \leftarrow 0$, $L^* \leftarrow \infty$
while $\delta < T$ do
    $k_a \leftarrow 1$, $P' \leftarrow 0$
    while $k_a < T$ do
        $P(\delta, k_a) \leftarrow \sum_{k_a}^{k_a+\delta} D(k)$
        if $P(\delta, k_a) > P'$ then
            $P' \leftarrow P(\delta, k_a)$
            $L' \leftarrow P' + C \cdot FP(\delta) \cdot T$
        end if
        $k_a \leftarrow k_a + 1$
    end while
    if $L' < L^*$ then
        $L^* \leftarrow L'$
        $\delta^* \leftarrow \delta$
    end if
    $\delta \leftarrow \delta + 1$
end while
return $\delta^*$
```

4.2 Adaptive Threshold

We present Algorithm 2 for finding optimal adaptive thresholds for any instance of the attacker-defender game, which is based on the SSE. The approach comprises 1) a dynamic-programming algorithm for finding minimum-cost thresholds subject to the constraint that the damage caused by a worst-case attack is at most a given value and 2) an exhaustive search, which finds an optimal damage value and thereby optimal thresholds. For ease of presentation, we use detection delays δ_k and the corresponding maximal thresholds η_k interchangeably (e.g., we let $FP(\delta_k)$ denote the false-positive rate of the maximal threshold that results in detection delay δ_k), and we let $\varDelta$ denote the set of all attainable detection delay values.

Theorem 1. *Algorithm 2 computes an optimal adaptive threshold.*

Proof (Sketch.). First, we prove that our dynamic-programming algorithm, called MinimumCostThresholds in Algorithm 2, finds minimum-cost thresholds subject to any damage constraint P. Then, we show that our exhaustive search finds an optimal damage constraint P, which minimizes the defender's loss given that the attacker plays a best response.

1) Minimum-Cost Thresholds In the first part, we assume that we are given a damage value P, and we have to find thresholds that minimize the total cost of false positives and threshold changes, subject to the constraint that any attack against these thresholds will result in at most P damage. In order to solve this

Algorithm 2 Algorithm for Optimal Adaptive Thresholds

```
Input $\mathcal{D}(k)$, $T$, $C$
SearchSpace $\leftarrow \left\{ \sum_{k=k_a}^{k_e} D(k) \,\middle|\, k_a \in \{1, \ldots, T-1\},\ k_e \in \{n+1, \ldots, T\} \right\}$
for all $P \in$ SearchSpace do
    $TC(P), \delta_1^*(P), \ldots, \delta_T^*(P) \leftarrow \textsc{MinimumCostThresholds}(P)$
end for
$P^* \leftarrow \arg\min_{P \in \text{SearchSpace}} TC(P)$
return $\delta_1^*(P^*), \ldots, \delta_T^*(P^*)$

function $\textsc{MinimumCostThresholds}(P)$
    $\forall\, m \in \{0, \ldots, T-1\},\ \delta \in \Delta :\ \textsc{Cost}(T+1, m, \delta) \leftarrow 0$
    for $n = T, \ldots, 1$ do
        for all $m \in \{0, \ldots n-1\}$ do
            for all $\delta_{n-1} \in \Delta$ do
                for all $\delta_n \in \Delta$ do
                    if $\delta_n > m$ then
                        $S(\delta_n) \leftarrow \textsc{Cost}(n+1, m+1, \delta_n) + C \cdot FP(\delta_n)$
                    else if $\sum_{k=n-m}^{n} \mathcal{D}(k) \leq P$ then
                        $S(\delta_n) \leftarrow \textsc{Cost}_P(n+1, \delta_n, \delta_n) + C \cdot FP(\delta_n)$
                    else
                        $S(\delta_n) \leftarrow \infty$
                    end if
                    if $\delta_{n-1} \neq \delta_n \wedge n > 1$ then
                        $S(\delta_n) \leftarrow S(\delta_n) + C_d$
                    end if
                end for
                $\delta^*(n, m, \delta_{n-1}) \leftarrow \arg\min_{\delta_n} S(\delta_n)$
                $\textsc{Cost}(n, m, \delta_{n-1}) \leftarrow \min_{\delta_n} S(\delta_n)$
            end for
        end for
    end for
    $m \leftarrow 0,\ \delta_0^* \leftarrow$ arbitrary
    for all $n = 1, \ldots T$ do
        $\delta_n^* \leftarrow \delta^*(n, m, \delta_{n-1}^*)$
        $m \leftarrow \min\{m+1, \delta_n^*\}$
    end for
    return $\textsc{Cost}(1, 0, \text{arbitrary}), \delta_1^*, \ldots, \delta_T^*$
end function
```

problem, we use a dynamic-programming algorithm. We will first discuss the algorithm without a cost for changing thresholds, and then show how to extend it to consider costly threshold changes.

For any two variables n and m such that $n \in \{1, \ldots, T\}$ and $0 \leq m < n$, we define $\textsc{Cost}(n, m)$ to be the minimum cost of false positives from n to T subject to the damage constraint P, given that we only have to consider attacks that start at $k_a \in \{n-m, \ldots, T\}$ and that attacks are not detected prior to n. If there are no thresholds that satisfy the damage constraint P under these conditions, we let $\textsc{Cost}(n, m)$ be ∞.[2]

We can recursively compute $\textsc{Cost}(n, m)$ as follows. For any $n < T$ and m, iterate over all possible detection delay values δ_n, and choose the one that results in the lowest cost $\textsc{Cost}(n, m)$. If $\delta_n > m$, then no attack could be detected at timestep n, and $\textsc{Cost}(n, m)$ would be the cost at timestep n plus the minimum cost for timesteps $\{n+1, \ldots, T\}$ given that attacks may start at $\{n-m, \ldots, T\} = \{(n+1)-(m+1), \ldots, T\}$. On the other hand, if $\delta_n \leq m$, then some attacks could be detected at timestep n, and the worst of these attacks would start at $n - m$. Hence, if $\sum_{k=n-m}^{n} \mathcal{D}(k) \leq P$, then $\textsc{Cost}(n, m)$ would be the cost at timestep n plus the minimum cost for timesteps $\{n + 1, \ldots, T\}$ given that attacks may start at $\{n + 1 - \delta_n, \ldots, T\}$. Otherwise, there would be an attack that could cause more than P damage, so $\textsc{Cost}(n, m)$ would be ∞ by definition since there would be no feasible thresholds for the remaining timesteps. Formally, we let

$$\textsc{Cost}(n, m) = \min_{\delta_n} \begin{cases} \textsc{Cost}(n+1, m+1) + FP(\delta_n), & \text{if } \delta_n > m \\ \textsc{Cost}(n+1, \delta_n) + FP(\delta_n), & \text{else if } \displaystyle\sum_{k=n-m}^{n} \mathcal{D}(k) \leq P \\ \infty & \text{otherwise} \end{cases} . \tag{9}$$

Note that in the equation above, $\textsc{Cost}(n, m)$ does not depend on $\delta_1, \ldots, \delta_{n-1}$, it depends only on the feasible thresholds for the subsequent timesteps. Therefore, starting from the last timestep T and iterating backwards, we are able to compute $\textsc{Cost}(n, m)$ for all timesteps n and all values m. Note that for $n = T$ and any δ_T, computing $\textsc{Cost}(T, m)$ is straightforward: if $\sum_{T-m}^{T} \mathcal{D}(k) \leq P$, then $\textsc{Cost}(T, m) = FP(\delta_T)$; otherwise, $\textsc{Cost}(T, m) = \infty$.

Having found $\textsc{Cost}(n, m)$ for all n and m, $\textsc{Cost}(1, 0)$ is by definition the minimum cost of false positives subject to the damage constraint P. The minimizing threshold values can be recovered by iterating forwards from $n = 1$ to T and again using Equation (9). That is, for every n, we select the threshold corresponding to the delay value δ_n^* that attains the minimum cost $\textsc{Cost}(n, m)$, where m can easily be computed from the preceding delay values $\delta_1^*, \ldots, \delta_n^*$.[3]

Costly Threshold Changes Now, we show how to extend the computation of $\textsc{Cost}$ to consider the cost C_d of changing the threshold. Let $\textsc{Cost}(n, m, \delta_{n-1})$ be the minimum cost for timesteps starting from n subject to the same constraints as before but also given that the detection delay at timestep $n-1$ is δ_{n-1}. Then, $\textsc{Cost}(n, m, \delta_{n-1})$ can be computed similarly to $\textsc{Cost}(n, m)$: for any $n < T$, iterate over all possible detection delay values δ_n, and choose the one that results in the lowest cost $\textsc{Cost}(n, m, \delta_{n-1})$. If $\delta_{n-1} = \delta_n$ or $n = 1$, then the cost would be computed the same way as in the previous case (i.e., similarly to Equation (9)).

[2] Note that in practice, ∞ can be represented by a sufficiently high natural number.

[3] Note that in Algorithm 2, we store the minimizing values $\delta^*(n, m)$ for every n and m when iterating backwards, thereby decreasing running time and simplifying the presentation of our algorithm.

Otherwise, the cost would have to also include the cost C_d of changing the threshold. Consequently, similarly to Equation (9), we define

$$\widehat{\textsc{Cost}}(n, m, \delta_{n-1}) = \begin{cases} \textsc{Cost}(n+1, m+1, \delta_n) + \mathit{FP}(\delta_n) & \text{if } \delta_n > m \\ \textsc{Cost}(n+1, \delta_n, \delta_n) + \mathit{FP}(\delta_n) & \text{if } \sum_{k=n-m}^{n} \mathcal{D}(k) \leq P \\ \infty & \text{otherwise} \end{cases}, \tag{10}$$

and then based on the value of δ_{n-1}, we can compute $\textsc{Cost}(n, m, \delta_{n-1})$ as

$$\textsc{Cost}(n, m, \delta_{n-1}) = \min_{\delta_n} \begin{cases} \widehat{\textsc{Cost}}(n, m, \delta_{n-1}) & \text{if } \delta_n = \delta_{n-1} \vee n = 1 \\ \widehat{\textsc{Cost}}(n, m, \delta_{n-1}) + C_d & \text{otherwise} \end{cases}. \tag{11}$$

Note that for $n = 1$, we do not add the cost C_d of changing the threshold. Similarly to the previous case, $\textsc{Cost}(1, 0, \text{arbitrary})$ is the minimum cost subject to the damage constraint P, and the minimizing thresholds can be recovered by iterating forwards.

2) Optimal Damage Constraint For any damage value P, using the above dynamic-programming algorithm, we can find thresholds that minimize the total cost $TC(P)$ of false positives and threshold changes subject to the constraint that an attack can do at most P damage. Since the defender's loss is the sum of its total cost and the damage resulting from a best-response attack, we can find optimal adaptive thresholds by solving

$$\min_P \; TC(P) + P \tag{12}$$

and computing the optimal thresholds $\boldsymbol{\eta}^*$ for the minimizing P^* using our dynamic-programming algorithm.

To show that this formulation does indeed solve the problem of finding optimal adaptive thresholds, we use indirect proof. For the sake of contradiction, suppose that there exist thresholds $\boldsymbol{\eta}'$ for which the defender's loss $\mathcal{L}'$ is lower than the loss $\mathcal{L}^*$ for the solution $\boldsymbol{\eta}^*$ of the above formulation. Let P' be the damage resulting from the attacker's best response against $\boldsymbol{\eta}'$, and let TC' be the defender's total cost for $\boldsymbol{\eta}'$. Since the worst-case attack against $\boldsymbol{\eta}'$ achieves at most P' damage, we have from the definition of $TC(P)$ that $TC' \geq TC(P')$. It also follows from the definition of $TC(P)$ that $L^* \leq TC(P^*) + P^*$. Combining the above with our supposition $L^* > L'$, we get

$$TC(P^*) + P^* \geq L^* > L' = TC' + P' \geq TC(P') + P'.$$

However, this is a contradiction since P^* minimizes $TC(P) + P$ by definition. Therefore, $\boldsymbol{\eta}^*$ must be optimal.

It remains to show that Algorithm 2 finds an optimal damage value P^*. To this end, we show that P^* can be found in polynomial time using an exhaustive search. Consider the set of damage values $\bar{\mathcal{D}}(k_a, k_e)$ from all possible attacks $k_a \leq k_e$, that is, the set

$$\left\{ \sum_{k=k_a}^{k_e} \mathcal{D}(k) \,\middle|\, k_a \in \{1, \ldots, T\}, k_e \in \{k_a, \ldots, T\} \right\}.$$

Let the elements of this set be denoted by $P_1, P_2, \ldots$ in increasing order. It is easy to see that for any i, the set of thresholds that satisfy the constraint is the same for every $P \in [P_i, P_{i+1})$. Consequently, for any i, the cost $TC(P)$ is the same for every $P \in [P_i, P_{i+1})$. Therefore, the optimal P^* must be a damage value P_i from the above set, which we can find by simply iterating over the set. □

Proposition 2. *The running time of Algorithm 2 is $\mathcal{O}(T^4 \cdot |\Delta|^2)$.*

Note that since possible detection delay values can be upper-bounded by T, the running time of Algorithm 2 is also $\mathcal{O}(T^6)$.

Proof. In the dynamic-programming algorithm, we first compute $\textsc{Cost}(n, m, \delta_{n-1})$ for every $n \in \{1, \ldots, T\}$, $m \in \{1, \ldots, n-1\}$, and $\delta_{n-1} \in \Delta$, and each computation takes $\mathcal{O}(|\Delta|)$ time. Then, we recover the optimal detection delay for all timesteps $\{1, \ldots, T\}$, and the computation for each timestep takes a constant amount of time. Consequently, the running time of the dynamic-programming algorithm is $\mathcal{O}(T^2 \cdot |\Delta|^2)$.

In the exhaustive search, we first enumerate all possible damage values by iterating over all possible attacks (k_a, k_e), where $k_a \in \{1, \ldots, T\}$ and $k_e \in \{k_a, \ldots, T\}$. Then, for each possible damage value, we execute the dynamic-programming algorithm, which takes $\mathcal{O}(T^2 \cdot |\Delta|^2)$ time. Consequently, the running time of Algorithm 2 is $\mathcal{O}(T^4 \cdot |\Delta|^2)$. □

Finally, note that the running time of the algorithm can be substantially reduced in practice by computing $\textsc{Cost}$ in a lazy manner: starting from $n = 1$ and $m = 0$, compute and store the value of each $\textsc{Cost}(n, m, \delta_{n-1})$ only when it is referenced, and then reuse it when it is referenced again. Unfortunately, this does not change the worst-case running time of the algorithm.

5 Numerical Results

In this section, we evaluate our approach numerically using an example. In particular, we consider the anomaly-based detection of deception attacks in water distribution networks. In such networks, an adversary may compromise pressure sensors deployed to monitor the leakages and bursts in water pipes. By compromising sensors, adversary may alter their true observations, which can then result in physical damage and financial losses. Next, we present the system model and the simulations of our results.

System Model. Figure 1 presents hourly water demand for a water network during a day [8]. Since demand is time-dependent, the expected physical damage and financial loss caused by an attack on sensors is also time-dependent. That is, the expected disruptions at a high-demand time would be more problematic than the disruptions at a low-demand time. Therefore, for each timestep $k \in \{1, ..., 24\}$, we can define the expected damage as $\mathcal{D}(k) = \alpha \cdot d(k)$ where $d(k)$ is the demand at time k, and $\alpha \in \mathbb{R}_+$ is a fixed value for scaling (for example, water price rate). In our experiments, we let $\alpha = 2$.

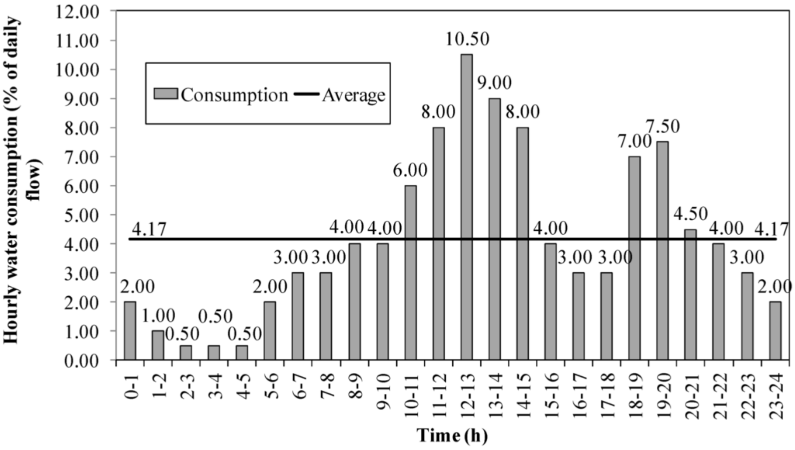

Fig. 1. Hourly water demand during a day [8].

To discover attacks, we use anomaly-based detection systems implementing sequential change detection. Based on the results presented in [7], we derive the attainable detection delays and false alarm rates for the detector as shown in Figure 2. We observe that for the detection delay $\delta = 0$, the false positive rate is $FP(\delta) = 0.95$, and for $\delta = 23$, the false positive rate is $FP(\delta) = 0.02$. As expected, the detection delay is proportional to the threshold, and the false positive rate is inversely proportional to the threshold [6].

Fixed Threshold. In the case of fixed threshold, the objective is to select the strategy that minimizes the defender's loss (2) while assuming the attacker will respond using a best-response attack. Letting $C = 7$ and using Algorithm 1, we obtain $\delta^* = 5$, and the optimal loss $L^* = 171.30$. Figure 3 shows the best-response attack corresponding to this threshold value. The best-response attack starts at $k_a^* = 10$ and attains the payoff $P^* = \sum_{k=10}^{15} \mathcal{D}(k) = 91$. Note that if the attacker starts the attack at any other timestep, the damage caused before detection is less than P^*.

Next, letting $C = 8$, we obtain $\delta^* = 6$ as the optimal defense strategy, which leads to the optimal loss $L^* = 181.86$, and best-response attack $k_a^* = 9$, with the

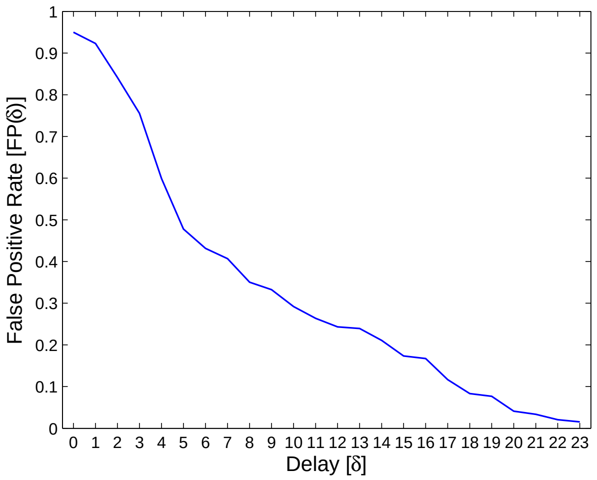

Fig. 2. Trade-off between the detection delay and the false positive rate.

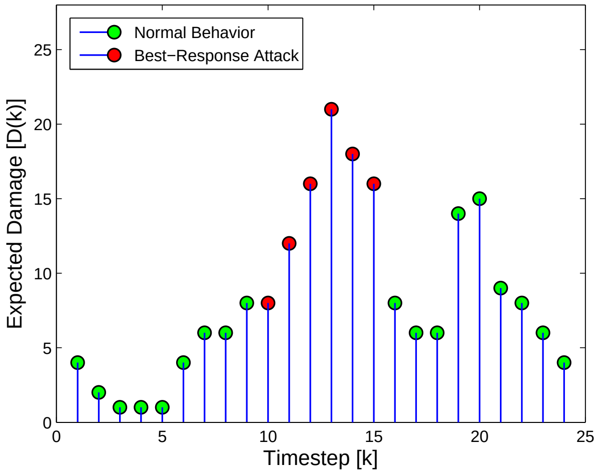

Fig. 3. Best-response attack corresponding to the optimal fixed threshold $\delta^* = 5$.

payoff $P^* = 99$. We observe that, as expected, the optimal delay is higher for the case of false alarms with higher costs.

Adaptive Threshold. Using the same setting, we use Algorithm 2 to find an optimal adaptive threshold. We let $C = 8$ and $C_d = 10$. As shown in Figure 4, we obtain the optimal adaptive threshold $\delta(k) =\ 23$ for $k \in \{1, .., 11\}$,

$\delta(k) = 1$ for $\{12, .., 15\}$, and $\delta(k) = 3$ for $\{17, ..., 23\}$. The resulting optimal loss is $L^* = 138.88$. Figure 4 shows the corresponding best-response attack, which starts at $k_a = 13$ and, attains the payoff $P^* = 39$. This figure demonstrates that the detection threshold decreases as the system experiences high-demand, so that the attacks can be detected early enough. On the other hand, as the system experiences low-demand, the threshold increases to have fewer false alarms.

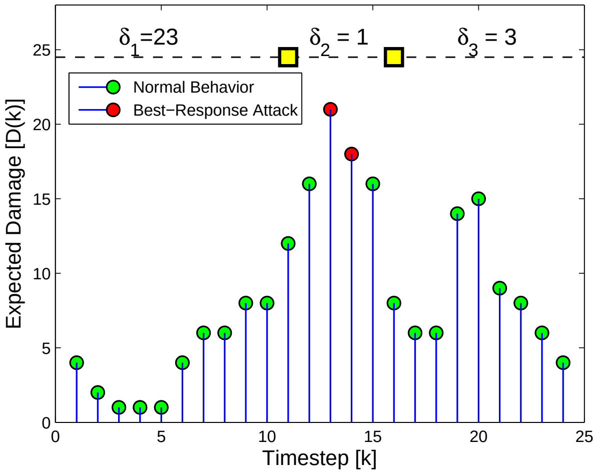

Fig. 4. Best-response attack corresponding to the optimal adaptive threshold. The yellow points indicate the times at which the threshold change occurs.

Comparison. Keeping $C = 8$ fixed, Figure 5 shows the optimal loss as a function of cost of threshold change C_d. For small values of C_d, the optimal losses obtained by the adaptive threshold strategy are significantly lower than the loss obtained by the fixed threshold strategy. As the cost of threshold change C_d increases, the solutions of adaptive and fixed threshold problems become more similar. In the current setting, the adaptive threshold solution converges to a fixed threshold when $C_d \geq 45$.

Furthermore, letting $C_d = 8$, Figure 6 shows optimal loss as a function of cost of false positives for fixed and adaptive threshold strategies. It can be seen that in both cases, the optimal loss increases as the cost of false alarms increases. However, in the case of adaptive threshold, the change in loss is relatively smaller than the fixed threshold.

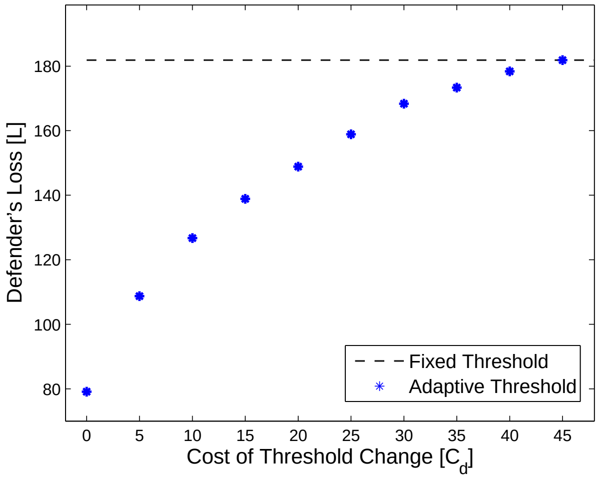

Fig. 5. The defender's loss as a function of cost of threshold change.

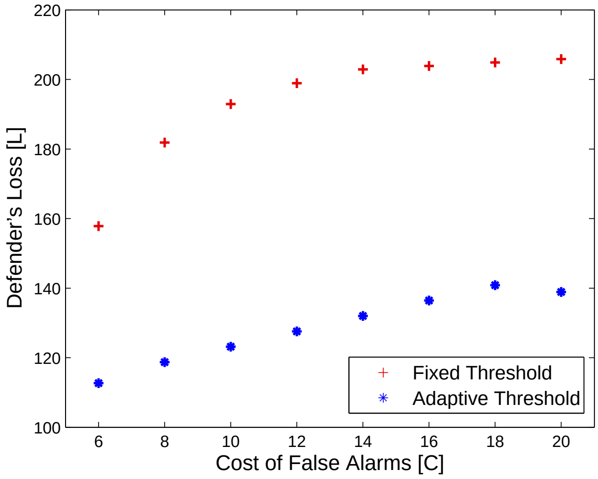

Fig. 6. The defender's loss as a function of cost of false alarms.

6 Related Work

The problem of threshold selection for anomaly detection systems has been widely studied in the literature. Nevertheless, prior work has not particularly addressed the optimal threshold selection problem in the face of strategic attacks when the damage corresponding to an attack depends on time-varying properties of the underlying physical system.

Laszka et al. study the problem of finding detection thresholds for multiple detectors while considering time-invariant damages [12]. They show that the problem of finding optimal attacks and defenses is computationally expensive, thereby, proposing polynomial-time heuristic algorithms for computing approximately optimal strategies. Cardenas et al. study the use of physical models for anomaly detection, and describe the trade-off between false alarm rates and the delay for detecting attacks [7]. Pasqualetti et al. characterize detection limitations for CPS and prove that an attack is undetectable if the measurements due to the attack coincide with the measurements due to some nominal operating condition [16].

Further, Alpcan and Basar study distributed intrusion detection in access control systems as a game between an IDS and an attacker, using a model that represents the flow of information from the attacker to the IDS through a network [3, 4]. The authors investigate the existence of a unique Nash equilibrium and best-response strategies under specific cost functions. This work is also related to the FlipIt literature [21, 14, 13]. FlipIt is an attacker-defender game that studies the problem of stealthy takeover of control over a critical resource, in which the players receive benefits proportional to the total time that they control the resource. In [17], the authors present a framework for the interaction between an attacker, defender, and a cloud-connected device. They describe the interactions using a combination of the FlipIt game and a signaling game.

Tantawy presents a comprehensive discussion on design concerns and different optimality criteria used in model-based detection problems [20]. Srivastava presents a comparison of detection methods, including Shiryayev-Roberts, CUSUM, and EWMA, based on the stationary average delay time [19]. Alippi et al. propose a model of adaptive change detection that can be configured at run-time [2]. This is followed by [22], in which the authors present a procedure for obtaining adaptive thresholds in change detection problems.

7 Concluding Remarks

In this paper, we studied the problem of finding optimal detection thresholds for anomaly-based detectors implemented in dynamical systems in the face of strategic attacks. We formulated the problem as an attacker-defender security game that determines thresholds for the detector to achieve an optimal trade-off between the detection delay and the false positive rates. To this end, first we presented an algorithm that computes optimal fixed threshold that is independent of time. Next, we defined adaptive threshold, in which the defender is allowed to change the detector's threshold with time. We provided a polynomial time algorithm to compute optimal adaptive threshold. Finally, we evaluated our results using a case study. Our simulations indicated that the adaptive threshold strategy achieves a better overall detection delay-false positive trade-off, and consequently minimize the defender's losses, especially when the damage incurred by the successful attack varied with time.

In future work, we aim to extend this work by considering: 1) Multiple systems with different time-varying damage for each subsystem; 2) Sequential hypothesis testing detectors, in which there exits a trade-off between false alarm rate, missed detection rate, and detection delay; and 3) Moving target defense techniques based on randomized thresholds.

Acknowledgment

This work is supported in part by the the National Science Foundation (CNS-1238959), Air Force Research Laboratory (FA 8750-14-2-0180), National Institute of Standards and Technology (70NANB15H263), Office of Naval Research (N00014-15-1-2621), and by Army Research Office (W911NF-16-1-0069).

References

1. M. Abrams and J. Weiss. Malicious control system cyber security attack case study – Maroochy Water Services, Australia. `http://csrc.nist.gov/groups/SMA/fisma/ics/documents/Maroochy-Water-Services-Case-Study_report.pdf`, Jul 2008.
2. C. Alippi and M. Roveri. An adaptive CUSUM-based test for signal change detection. In *Proceedings of the 2006 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 5752–5755. IEEE, 2006.
3. T. Alpcan and T. Basar. A game theoretic approach to decision and analysis in network intrusion detection. In *Proceedings of the 42nd IEEE Conference on Decision and Control (CDC)*, volume 3, pages 2595–2600. IEEE, 2003.
4. T. Alpcan and T. Başar. A game theoretic analysis of intrusion detection in access control systems. In *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC)*, volume 2, pages 1568–1573. IEEE, 2004.
5. S. Amin, G. A. Schwartz, and A. Hussain. In quest of benchmarking security risks to cyber-physical systems. *IEEE Network*, 27(1):19–24, 2013.
6. M. Basseville, I. V. Nikiforov, et al. *Detection of abrupt changes: Theory and application*, volume 104. Prentice Hall, Englewood Cliffs, 1993.
7. A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry. Attacks against process control systems: risk assessment, detection, and response. In *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, pages 355–366. ACM, 2011.
8. B. Durin and J. Margeta. Analysis of the possible use of solar photovoltaic energy in urban water supply systems. *Water*, 6(6):1546–1561, 2014.
9. T. Kailath and H. V. Poor. Detection of stochastic processes. *IEEE Transactions on Information Theory*, 44(6):2230–2231, 1998.
10. D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. Nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *Journal of Artificial Intelligence Research*, 41:297–327, 2011.
11. D. Kushner. The real story of stuxnet. *Spectrum, IEEE*, 50(3):48–53, 2013.
12. A. Laszka, W. Abbas, S. S. Sastry, Y. Vorobeychik, and X. Koutsoukos. Optimal thresholds for intrusion detection systems. In *Proceedings of the 3rd Annual Symposium and Bootcamp on the Science of Security (HotSoS)*, pages 72–81, 2016.

13. A. Laszka, G. Horvath, M. Felegyhazi, and L. Buttyan. FlipThem: Modeling targeted attacks with FlipIt for multiple resources. In *Proceedings of the 5th Conference on Decision and Game Theory for Security (GameSec)*, pages 175–194, November 2014.
14. A. Laszka, B. Johnson, and J. Grossklags. Mitigating covert compromises: A game-theoretic model of targeted and non-targeted covert attacks. In *Proceedings of the 9th Conference on Web and Internet Economics (WINE)*, pages 319–332, December 2013.
15. R. M. Lee, M. J. Assante, and T. Conway. German steel mill cyber attack. Technical report, SANS Industrial Control Systems, December 2014.
16. F. Pasqualetti, F. Dorfler, and F. Bullo. Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, 58(11):2715–2729, 2013.
17. J. Pawlick, S. Farhang, and Q. Zhu. Flip the cloud: Cyber-physical signaling games in the presence of advanced persistent threats. In *Proceedings of the 6th International Conference on Decision and Game Theory for Security (GameSec)*, pages 289–308. Springer, 2015.
18. A. Shiryaev. The problem of the most rapid detection of a disturbance in a stationary process. *Soviet Math. Dokl*, 2(795-799), 1961.
19. M. Srivastava and Y. Wu. Comparison of EWMA, CUSUM and Shiryayev-Roberts procedures for detecting a shift in the mean. *The Annals of Statistics*, 21(2):645–670, 1993.
20. A. M. Tantawy. *Model-based Detection in Cyber-Physical Systems.* PhD thesis, Vanderbilt University, 2011.
21. M. Van Dijk, A. Juels, A. Oprea, and R. L. Rivest. Flipit: The game of stealthy takeover. *Journal of Cryptology*, 26(4):655–713, 2013.
22. G. Verdier, N. Hilgert, and J.-P. Vila. Adaptive threshold computation for cusum-type procedures in change detection and isolation problems. *Computational Statistics & Data Analysis*, 52(9):4161–4174, 2008.