Reinforcement Learning Based Approach for Flip Attack Detection

Hanxiao Liu[1], Yuchao Li[1], Jonas Mårtensson, Lihua Xie and Karl Henrik Johansson

***Abstract*— This paper addresses the detection problem of flip attacks to sensor network systems where the attacker flips the distribution of manipulated sensor measurements of a binary state. The detector decides to continue taking observations or to stop based on the sensor measurements, and the goal is to have the flip attack recognized as fast as possible while trying to avoid terminating the measurements when no attack is present. The detection problem can be modeled as a partially observable Markov decision process (POMDP) by assuming an attack probability, with the dynamics of the hidden states of the POMDP characterized by a stochastic shortest path (SSP) problem. The optimal policy of the SSP solely depends on the transition costs and is independent of the assumed attack possibility. By using a fixed-length window and suitable feature function of the measurements, a Markov decision process (MDP) is used to approximate the behavior of the POMDP. The optimal solution of the approximated MDP can then be solved by any standard reinforcement learning methods. Numerical evaluations demonstrates the effectiveness of the method.**

I. INTRODUCTION

Networked embedded sensors are widely used to monitor plants and to detect anomaly. At the same time, due to their vulnerability to malicious attacks, increasing importance has been attached to researches on the security of those systems. There are many works focusing on efficient detection frameworks in response to different kinds of attack strategies. Mo *et al.* [1] proposed an active detection scheme, physical watermarking, which introduces an authentication signal to enable the detection of replay attacks. In [2], centralized and distributed filters were designed to detect and identify various attacks. Some other detection approaches were studied in [3] with other attack approaches. Different from the above context, [4], [5] employed game-theoretic approaches to analyze the attacker's behavior for detection purpose. It is shown that the flip attack, where the attacker flips the distribution of manipulated sensors' measurements, is optimal from the attackers' perspective to a broad class of problems. Therefore, it is well-worth some attention to design detectors for flip attacks.

While the above works study the problem with the perspective of system theory, some researchers attempted to capture the properties of the detection problems by the formalism of partially observable Markov decision process (POMDP). This is achieved by assuming an attack possibility [6], [7] in various forms. With such a modeling approach, the much celebrated reinforcement learning (RL) methods [8], [9] can be applied to solve the problem. Among available options, a theoretically sound approach is to introduce the belief states and to solve in turn the induced Markov decision processes (MDPs) with the belief states as its states. However, there are two major drawbacks of this approach, one of which is generally true for all POMDP problems, while the other is particularly damaging to the problems studied here. First, regardless of the size of the state space of the POMDP, the belief state can take infinite number of possible values, which makes the induced MDP infinite dimensional and therefore challenging to solve [9]. Second, the detection problems studied here involve extraneous inputs from attackers, which cannot be captured by the POMDP framework. To address the issue, a transition probability is assumed to model the attack possibility, which is a major approximation, as the true attack probability varies due to various reasons and the transition probability used in POMDP may be different from the true attack probability. When solving a POMDP using belief states, a so-called state estimator is used to compute its value online. It relies explicitly on the transition probability, and makes the solution sensitive to the assumed transition probability. To circumvent those challenges, Kurt *et al.* [7] applied fixed-length window of observation as the state for online detection in smart grids. Similar idea also appeared in [10] for online learning and attack design. However, unlike the success of RL reported for playing games where the POMDP is given by some simulators with transition probabilities and costs enclosed in the simulator [11],they are here part of the design task. It is not clear from those works how the POMDPs shall be designed.

The focus in our work is somewhat like the quickest change detection (QCD) problem, as we aim to determine if there is an attack at every time step as quickly as possible. However, unlike those problems where the probability density functions (p.d.f.'s) before and after the change point are known [12], [13], in our work, only a set of possible p.d.f.'s are known, which is another challenge.

In this work, we focus on the detection problem of flip attacks where a group of sensors are used to measure a binary

[1] Both authors contributed equally to this work.

H. Liu is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, and the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Sweden. hanxiao001@ntu.edu.sg.

Y. Li, J. Mårtensson, and K.H. Johansson are with the Division of Decision and Control Systems, the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Sweden.{yuchao,jonas1,kallej}@kth.se.

L. Xie is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. elhxie@ntu.edu.sg.

This work is supported by the A*STAR Industrial Internet of Things Research Program, under the RIE2020 IAF-PP Grant A1788a0023, the Knut and Alice Wallenberg Foundation, the Swedish Foundation for Strategic Research, and the Swedish Research Council.

state. Attackers flip the sensors' distribution to confuse the fusion center. The contributions of our work are: 1) To the best of our knowledge, this is the first work that studies the detection problem of flip sensor attacks via RL approach. The proposed approach can be extended to other Cyber-Physical Systems. 2) Conditions for designing POMDP used to model the flip attacks are given. It is shown that the optimal behavior is independent of the assumed attack possibility. 3) It is shown that the obtained detector is robust to the assumed attack probability via numerical evaluations.

Notations: $\mathbb{R}$ is the set of reals. $\mathbb{R}^m$ denotes the m-dimensional Euclidean space. A^{T} is the transpose of matrix A. $\lfloor a \rfloor$ is the floored integer of the real number a. $\mathrm{supp}(\mathbf{v})$ denotes the set of indices of non-zero elements in the vector $\mathbf{v}$. $|\mathcal{S}|$ is the cardinality of finite set $\mathcal{S}$. $\mathrm{N}(\nu, \sigma^2)$ represents the Gaussian distribution with mean ν and variance σ^2. $\mathbb{1}_{\mathcal{S}}(j)$ is an indicator function of some subset $\mathcal{S}$, which equals 1 if $j \in \mathcal{S}$ and 0 otherwise.

II. PROBLEM FORMULATION

We consider a flip attack detection problem with the system diagram shown in Fig. 1. A plant P possesses a binary state $\theta \in \{0, 1\}$, measured by sensors $s_1, \cdots, s_m$, whose indices form set $\mathcal{S}$. Define the measurement from all m sensors at time k as: $\mathbf{y}(k) \stackrel{\triangle}{=} [y_1(k) \quad y_2(k) \quad \cdots \quad y_m(k)]^{\mathrm{T}} \in \mathbb{R}^m$. All sensors' measurements $\{y_i(k)\}_{j \in \mathcal{S}}$ are independently and identically distributed (i.i.d.) under normal operation. For any Borel-measurable set $B \subset \mathbb{R}$, the probability that $y_i(k) \in B$ is $\kappa_0(B)$ when $\theta = 0$ and equals $\kappa_1(B)$ when $\theta = 1$. Given the measurements of the sensors and the knowledge of the distributions κ_0 and κ_1, a fusion center FC is designed to infer the state θ. Naturally, it is assumed that the induced measures κ_0 and κ_1 are different and are absolutely continuous. However, due to the presence of a

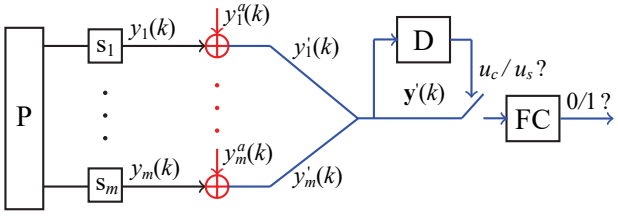

Fig. 1: The system diagram. Here P stands for the process, which possesses the binary state θ. $\mathrm{s}_1, \cdots, \mathrm{s}_m$ stand for the m sensors, which are subject to the flip attack. D and FC are the detector and the fusion center, respectively.

malicious adversary that try to compromise the performance of the fusion center by attacking some sensors, the fusion center receives the following manipulated measurements at time k:

$$\mathbf{y}'(k) = \mathbf{y}(k) + \mathbf{y}^a(k), \tag{1}$$

where $\mathbf{y}^a(k) \in \mathbb{R}^m$ is the bias vector injected by the attacker at time k. Therefore, a detection unit D forks the measurements $\mathbf{y}'(k)$ and is designed to detect possible attack given received measurements $\mathbf{y}'(k)$ and the distributions κ_0 and κ_1, without knowing the true state θ. The detector can command the FC to continue process the received measurements or to stop via the switch signal u_c (continue) and u_s (stop).

Regarding the attack type, we make following assumptions. The type of attack studied here is typical in the hypothesis testing and can be found in [4], [5].

Assumption 1 (Attacker's knowledge): The attacker has the knowledge of the probability of measures κ_0 and κ_1 and the true state θ.

Assumption 2 (l-sparse attack): There exists an index set $\mathcal{L} \subset \mathcal{S} \stackrel{\triangle}{=} \{1, 2, \ldots, m\}$ with $|\mathcal{L}| \leq l$, where $l \leq \lfloor \frac{m}{2} \rfloor$, such that $\cup_{k=1}^{\infty} \mathrm{supp}\{\mathbf{y}^a(k)\} = \mathcal{L}$. Besides, the system knows the number l, but it does not know the set $\mathcal{L}$.

Remark 1: When m is large, typically we have $l \ll \frac{m}{2}$.

Assumption 3: The compromised sensors are fixed during the whole attack period and the attack will not stop until it is detected or the detector stops detection.

For the type of attacks specified by the assumptions above, the attacker may design the injected signals via various strategies and here we focus on the detection of the flip attack where the attacker flips the distribution of the corrupted sensors' measurements to confuse the fusion center [5]. This strategy has been shown to be optimal from attacker's perspective when exactly l sensors are compromised. The strategy is when $\theta = 0$, the probability measure generated by $y'_j(k)_{j \in \mathcal{L}}$ is κ_1 and when $\theta = 1$, it is κ_0. Correspondingly, the attacked signal $y^a_j(k)_{j \in \mathcal{L}}$ is derived as follows:

$$y^a_j(k) = \begin{cases} y'_j(k) - y_j(k) & \text{if } j \in \mathcal{L}, \\ 0 & \text{if } j \notin \mathcal{L}. \end{cases} \tag{2}$$

In this work, we focus on the design of flip attack detector to protect a sensor system aimed to estimate a binary state. Due to the existence of the attack, there are two operation situations: "normal" and "abnormal". When an adversarial launches the attack, the distribution of the compromised sensors' measurements flips to the distribution under the opposite binary state. The detector decides on whether to stop and declare that there is an attack or to continue receiving the observations based on the sensor measurements. The desired behavior should command "continue" if state is "normal" and "stop" if otherwise. The problem of interest is to design the detector given the scope specified here.

III. MODELING OF THE FLIP ATTACK VIA POMDP

In this section, we introduce the POMDP applied to model the flip attack. As opposed to many successful cases reported via the RL methods where there are already simulators available, here the POMDP needed to serve as the simulator is part of the challenge. This includes crafting the transition probabilities and the transition costs of the underlying MDP, and the conditional observation probabilities of obtaining various observations. The optimal policy of the designed MDP shall give u_c when the state is "normal" and u_s otherwise. We will show that this solely depends on the transition

costs and is irrelevant to the assumed attack probability. In addition, we will give some rationales on how the conditional observation probabilities are defined. The POMDP derived here will serve as the simulator used to train the detector.

A. Modeling the dynamics of hidden states as an SSP

The states of the underlying MDP shall include "normal" state, "abnormal" state, and termination, where the termination is an absorbing state. This type of problems is widely known as stochastic shortest path (SSP) problem. We will use the semicontractive models introduced in [14] and show that the optimal policy solely depends on the transition costs.

We denote by I the state space of the underlying SSP problem and by U the control space. The state space has two elements 1 and 2, standing for "normal" state and "abnormal" state, respectively, and we will use i or i' to represent the unspecified states in I. The admissible control options are to continue and to stop, denoted as u_c and u_s, respectively, the same for all $i \in I$, and we will use u to represent the unspecified control. We denote by $p_{ii'}(u)$ the transition probability from i to i' under control u and denote by $g(\cdot, u, \cdot)$ a deterministic nonnegative function which returns the transition cost. The transition graph is shown in Fig. 2. It is clear that for the detection problem, we have

$$p_{1i}(u_c) > 0,\ p_{22}(u_c) = 1,\ p_{it}(u_c) = 0,\ p_{it}(u_s) = 1, \quad (3)$$

where $t \notin I$ denotes the terminal state, and we require

$$p_{tt}(u) = 1,\ g(t, u, t) = 0,\ \forall u \in U. \quad (4)$$

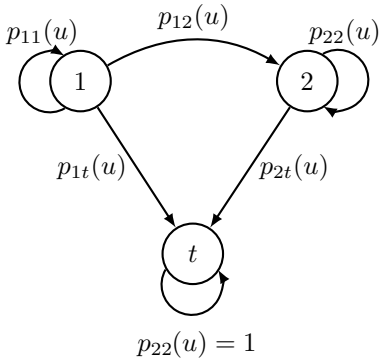

Fig. 2: The transition graph of the SSP model. There are 2 states, plus the termination state t.

A function $\mu : I \to U$ is named as a policy and the set of all policies is denoted by $\mathcal{M}$. It is easy to see that we have $|\mathcal{M}| = 4$. In the context of SSP, a policy is proper if under such a policy, the state is guaranteed to reach t regardless of the initial state; otherwise, it is improper. We denote by $\mathcal{E}(I)$ the set of functions $J : I \to \mathbb{R}^*$ where $\mathbb{R}^* = \mathbb{R} \cup \{\infty, -\infty\}$. We use the mapping $H : I \times U \times \mathcal{E}(I) \to \mathbb{R}^*$ to define the SSP problem as

$$H(i, u, J) \triangleq p_{it}(u) g(i, u, t) + \sum_{i' \in I} p_{ii'}(u) \left(g(i, u, i') + J(i')\right).$$

Then the mappings $T_\mu : \mathcal{E}(I) \to \mathcal{E}(I)$ for every $\mu \in \mathcal{M}$, and $T : \mathcal{E}(I) \to \mathcal{E}(I)$ can be defined in turn as

$$T_\mu J(i) \triangleq H(i, \mu(i), J),\ TJ(i) \triangleq \min_{\mu \in \mathcal{M}} T_\mu J(i),\ \forall i \in I.$$

In addition, the superscript of the operators means composition, viz., $(T^2 J)(i) \triangleq \big(T(TJ)\big)(i)$. Besides, we denote by $J_\mu \in \mathcal{E}(I)$ the cost function of μ defined pointwise by

$$J_\mu(i) \triangleq \limsup_{k \to \infty} (T_\mu^k \bar{J})(i),\ \forall i \in I,$$

where $\bar{J}(i) = 0$ for all i.

Naturally, a desired policy μ_d would be that $\mu_d(1) = u_c$ and $\mu_d(2) = u_s$. Then a plausible choice of stage costs is

$$g(1, u_c, 1) = 0,\ g(1, u_c, 2) > 0,\ g(2, u_c, 2) > 0,$$
$$g(1, u_s, t) > 0,\ g(2, u_s, t) = 0. \quad (5)$$

The costs of other situations need not be defined as they have zero transition probability. With the specified problem data, the fundamental questions required to be answered are: 1) is there a fixed point of the corresponding Bellman equation; 2) If so, is the fixed point a cost function of certain policy μ^*; 3) what are the conditions needed in order to have $\mu_d = \mu^*$. We recall the following useful lemma for the answers.

Lemma 1 (Proposition 2, [15]): For any SSP problem defined in the form of the mapping $H(\cdot, \cdot, \cdot)$ with both I and U being finite, assume that there exists at least one proper policy, and the cost functions of all improper policies have value infinity for at least one state. Then there exists $J^* : I \to \mathbb{R}$ such that

$$J^*(i) = (TJ^*)(i),\ J^*(i) = \min_{\mu \in \mathcal{M}} J_\mu(i), \forall i \in I, \quad (6)$$

with the optimal policy that attains the value of J^* denoted as $\mu^* \in \mathcal{M}$. In addition, for every proper μ, it holds that

$$J_\mu(i) = (T_\mu J_\mu)(i),\ \forall i \in I, \quad (7)$$

which needs not to be true for the improper ones.

Aided by the above result, we have the following theorem.

Theorem 1: The SSP problem defined by (3), (4), and (5) has the following property: a) the corresponding Bellman equations fulfill (6), (7), independent of the choice of $p_{12}(u_c)$; b) the attained optimal policy μ^* is the desired policy μ_d if $g(1, u_s, t) > g(1, u_c, 2)$, regardless of the choice of $p_{12}(u_c)$.

Proof: For part a), one can verify that $\mu(i) = u_s$ is a proper policy and that all improper policies have cost function infinity for state 2. Therefore, a) follows from Lemma 1, which does not rely on the specific value of $p_{12}(u_c)$. For b), since J^* is the fixed point of T, we have

$$J^*(1) = \min\Big\{g(1, u_s, t),\ p_{11}(u_c)\big(0 + J^*(1)\big) + p_{12}(u_c)\big(g(1, u_c, 2) + J^*(2)\big)\Big\}, \quad (8)$$

$$J^*(2) = \min\Big\{0,\ g(2, u_c, 2) + J^*(2)\Big\}. \quad (9)$$

From (9), we have $J^*(2) = 0$ and $\mu^*(2) = \mu_d(2) = u_s$. To have $\mu^*(1) = \mu_d(1) = u_c$, one can see that it is required to have $g(1, u_s, t) > g(1, u_c, 2)$. ∎

From Theorem 1, by setting $g(1, u_s, t) > g(1, u_c, 2)$, the SSP can capture the assumed characteristics of the flip attack.

B. Design of the conditional observation probabilities

The SSP introduced in Section III-A is used to model the dynamics of hidden states. However, the true state $i = 1$ or 2 is not accessible to the detector. Instead, a measurement $\mathbf{y}'(k) \in \mathbb{R}^m$ defined in (1) is available, which makes the environment partially observable and in turn the problem a POMDP. Needless to say, the measurement is conditioned on the state and control of SSP, viz., i and u. However, it is also conditioned on the binary state θ and the compromised sensor index set $\mathcal{L}$. If one fixes θ and $\mathcal{L}$, then the conditional observation probabilities are fully specified by the attack type and strategy. Under Assumptions 2 and 3, one can verify that there are in total $|\mathcal{I}|$ different POMDPs induced by the same SSP where $\mathcal{I}$ is an index set defined as $\mathcal{I} \triangleq \left\{1, 2, \ldots, 2\sum_{\ell=1}^{l} \frac{m!}{(m-\ell)!}\right\}$. For every $\ell \in \mathcal{I}$, the remote state θ and compromised sensors $\mathcal{L}$ are fixed and we will name its corresponding POMDP as ℓ-POMDP. To have all those cases covered by one POMDP, we introduce a probability distribution η over $\mathcal{I}$, viz., $\eta(\ell) \geq 0\ \forall \ell \in \mathcal{I}$ and $\sum_{\ell\in\mathcal{I}} \eta(\ell) = 1$. Such a distribution indicates how likely one particular case ℓ occurs. For example, it is more likely to have one sensor get attacked than to have two, and this is reflected by η where the distribution on cases fewer sensors under attack is higher than those with more. Given the distribution η, when the hidden state is 2, the probability of certain observation $\mathbf{y}'$ is given by the sum of products between the probability of any case ℓ specified by η, and the probability that $\mathbf{y}'$ is observed in ℓ-POMDP.

IV. RL APPROACH TO THE DETECTION PROBLEM

In principle, the POMDP used to model the flip attack can be solved by introducing the belief states and solving in turn the induced MDP with belief states as its states. However, such an approach relies explicitly on the specific values of transition probabilities in the SSP and the assumed case distribution η. To obtain a detector that is robust to the change of those values, we apply a RL approach to solve the problem. We will show here how the learning problem is formulated, and sketch the procedure to train the detector.

A. The target MDP learned by RL

One major challenge of POMDPs is that the Markov property is lost. Motivated by [16], it is common to apply as an observation a sequence of past measurements and actions to infer the hidden states. Here we use as an observation at time k a stored measurement with length $w > 1$ given by

$$\mathbf{o}_k \triangleq \begin{bmatrix}\mathbf{y}^{\mathrm{T}}(k-w+1) & \cdots & \mathbf{y}^{\mathrm{T}}(k-1) & \mathbf{y}^{\mathrm{T}}(k)\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{mw}.$$

Here the control need not to be recorded as the only reasonable control is u_c.

Denote by O_ℓ the set of all possible observation $\mathbf{o} \in \mathbb{R}^{mw}$ when the POMDP index is ℓ and define O as $\cup_{\ell\in\mathcal{I}} O_\ell$. Assume that there exists a feature function $\phi : O \to X$ where $|X| < \infty$ and $X \subset \mathbb{R}^n$, and denote $X_\ell \triangleq \phi(O_\ell)$. Here the fundamental assumption we use is that for every $\ell \in \mathcal{I}$, there is a MDP characterized by a mapping $\tilde{H}_\ell : X_\ell \times U \times \mathcal{E}(X_\ell) \to \mathbb{R}^*$ given by

$$\begin{aligned}\tilde{H}_\ell(\mathbf{x}, u, V_\ell) =& \tilde{p}_{\ell,\mathbf{x}t}(u) r_\ell(\mathbf{x}, u, t) + \\ & \sum_{\mathbf{z}\in X_\ell} \tilde{p}_{\ell,\mathbf{x}\mathbf{z}}(u)\big(r_\ell(\mathbf{x}, u, \mathbf{z}) + V_\ell(\mathbf{z})\big),\end{aligned}$$

where $\tilde{p}_{\ell,..}(\cdot)$, $r_\ell(\cdot)$, and $V_\ell(\cdot)$ are defined accordingly, such that the mean cost of ℓ-POMDP is close to the mean cost of the MDP defined by the above operator after feature transformation. Then the POMDP defining the flip attack can be approximated by $\tilde{H} : X \times U \times \mathcal{E}(X) \to \mathbb{R}^*$ given by

$$\tilde{H}(\mathbf{x}, u, V) = \frac{\sum_{\ell\in\mathcal{I}_\mathbf{x}} \left(\eta(\ell)\tilde{H}_\ell(\mathbf{x}, u, V_{|X_\ell})\right)}{\sum_{\ell\in\mathcal{I}_\mathbf{x}} \eta(\ell)}, \qquad (10)$$

where $\mathcal{I}_\mathbf{x} \triangleq \{\ell \in \mathcal{I} : \mathbb{1}_{X_\ell}(\mathbf{x}) = 1\}$, and $V_{|X_\ell}$ is the restriction of V on X_ℓ. The MDP defined by $\tilde{H}$ is the target MDP to be learned by the training algorithm.

B. Training the detector

With above formulation, we obtain a standard RL problem with X as state space and U as control space. Such a problem can be solved by many different RL methods and we use Q-learning [17] as an example. The pseudocode is given in Algorithm 1. To address the exploration and exploitation trade-off, a distribution γ for initial states of SSP is specified. In addition, we denote by $i \sim \gamma(I)$ a sample from distribution γ defined on I, and similar notation is used for $\ell \sim \eta(\mathcal{I})$. We denote by $j \sim \text{SSP}(i, u)$ the sampled next state of SSP given current state and control pair (i, u), and $o \sim \ell\text{-POMDP}(i, u)$ the sampled observation of ℓ-POMDP given current state and control pair (i, u). With a slight abuse of notation, we denote by $u \sim \min_v Q_\varepsilon(\mathbf{x}, v)$ the sampled control from a greedy exploration ε policy given the current $Q(\cdot, \cdot)$, the current state $\mathbf{x}$ and exploration rate ε.

V. NUMERICAL EVALUATION

In this section, the numerical evaluation of the proposed method is presented. For some given sets of probability measures κ_θ, the parameters of SSP used for modeling the attack, the fixed-length window, the feature functions, and parameters used in Q-learning are given. Via tuning the cost defined in the SSP model, the obtained detector exhibits that there is a trade-off between detecting attack early and giving false alarm. In addition, the detector, without knowing the binary state θ, has a comparable performance with the classical QCD algorithm that equips the true value θ.

Algorithm 1 Q-learning for detector training

Input: Problem data, the number of episodes N, and the learning rate $\alpha \in (0, 1]$, the exploration rate $\varepsilon \in (0, 1)$, initial state distribution γ.
Output: The optimal state-action value function $Q(\mathbf{x}, u)$

Initialize $Q(\mathbf{x}, u)$, $\forall \mathbf{x} \in X$, $u \in U$. $i \sim \gamma(I)$, $u \leftarrow u_c$.
for each $n \in N$ **do**
  $\ell \sim \eta(\mathcal{I})$, $\mathbf{o} \sim \ell$-POMDP(i, u).
  while $i \neq t$ **do**
    $\mathbf{x} \leftarrow \phi(\mathbf{o})$, $u \sim \min_v Q_\varepsilon(\mathbf{x}, v)$.
    if $u = u_s$ **then**
      $c \leftarrow g(i, u, t)$, $Q(\mathbf{x}, u) \leftarrow (1-\alpha)Q(\mathbf{x}, u) + \alpha c$
    else
      $i' \sim \text{SSP}(i, u)$, $\mathbf{o}' \sim \ell$-POMDP(i', u).
      $\mathbf{x}' \leftarrow \phi(\mathbf{o}')$, $c \leftarrow g(i, u, i')$, $u' \in \min_u Q(\mathbf{x}', u)$,
      $Q(\mathbf{x}, u) \leftarrow (1-\alpha)Q(\mathbf{x}, u) + \alpha\,[c + Q(\mathbf{x}', u')]$,
      $i \leftarrow i'$, $\mathbf{o} \leftarrow \mathbf{o}'$.
    end if
  end while
end for

A. Simulation setup and modelling parameters

The density functions under normal operation are given by $\text{N}(\nu_\theta, \sigma_\theta^2)$ where $\theta = 0, 1$, $\nu_0 = -\nu_1$, and $\sigma_0 = \sigma_1 = 1$. Various values of ν_θ have been tested. The binary state is measured by $m = 5$ sensors and by Assumption 2, at most $l = 2$ sensors are attacked. Fig. 3 illustrates the measurements of all the sensors in one trail where $\theta = 0$, $\nu_0 = 0.7$, and the attack occurred at $k = 50$ on sensor 1. The dynamics of the hidden states is modeled by SSP,

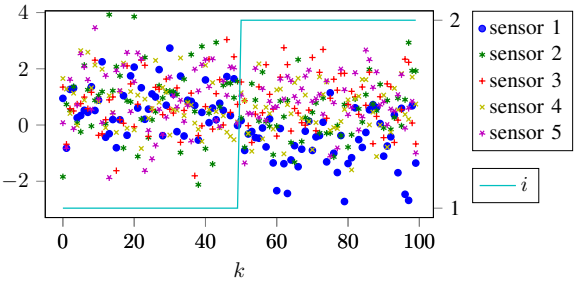

Fig. 3: Sensor measurements in one trail where $\theta = 0$, $\nu_0 = 0.7$, and the attack occurs at $k = 50$ on sensor 1.

with transition probability given by (3) where $p_{11}(u_c) = 0.95$, $p_{12}(u_c) = 0.05$. The transition probability from $i = 1$ to 2 represents attack probability and are kept fixed throughout the training process. It will be shown in the evaluation results that the trained detector is robust to this assumed transition probability. The cost per stage is given by (5), with $g(1, u_s, t) = 1$, $g(1, u_c, 2) = g(2, u_c, 2) \in (0, 1)$. Recall that it is required to have $g(1, u_c, 2) < g(1, u_s, t)$ in order to make μ_d, the desired policy, the only policy whose cost function is the fixed point of the Bellman equation (6) and at the same time an optimal policy. The specific value of $g(1, u_c, 2)$ serves as the tuning parameter of and $g(2, u_c, 2)$ is set to be the same value.

The size of the fixed-length window involves a trade-off between encoding more information and demanding more memory. In this example, window size between $w = 2$ to 6 are explored. This is by no means guaranteed to be optimal and it may be tuned to get better results. Designing the feature function of the observation is typically challenging and requires domain specific knowledge. Here we use arithmetic means as features. With a better crafted feature functions, a performance improvement may be expected. The feature used here at time k is defined as $\mathbf{x}_k \triangleq [x_1(k)\ x_2(k) \cdots\ x_m(k)]^{\mathrm{T}}$, where for $s \in \mathcal{S}$, and $\mathcal{W} = \{0, \ldots, w-1\}$, $x_s(k) = \sum_{\ell \in \mathcal{R}} \left(\ell \times \mathbb{1}_{R_\ell} \left(\frac{\sum_{j \in \mathcal{W}} y_s(k-j)}{w} \right) \right)$, with $\{R_\ell\}_{\ell \in \mathcal{R}}$ as a finite partition of the $\mathbb{R}$ whose index set is $\mathcal{R}$. The partition serves as a tuning parameter.

In addition, the distribution $\eta(\mathcal{I})$ is defined such that the chance of one sensor under attack is 80% and of two as 20% while the chances of θ being 0 and 1 are equal. In addition, the initial state i of SSP for each episode is given by a Bernoulli distribution, with probability 0.3 that $i = 1$ at the beginning of each episode and 2 otherwise.

B. Training setup and performance criteria

Q-learning algorithm is applied to obtain a MDP in form of (10) that approximates the behavior of POMDP introduced in Section III. The learning rate α is set to be constant and different values of α have also been explored. The learning rates that fulfill the Robbins-Monro conditions, which is required to have the convergent behaviors, have also been tested. It results in no clear improvement and therefore is not presented here. The number of training episodes N varies between 300 thousands to 1 million depending on the size of the state space. Once the training process is complete, a table of Q values with data size less than 0.5 MB is obtained.

To test the obtained detector, Monte Carlo simulations with 20 thousands trials are used, half of which are always in normal operation, while the other half always under attack. They corresponds to cases where the transition probability $p_{12}(u_c)$ of SSP is set to be 0 and 1 respectively. The transition probabilities are significantly different from those used to train the detector, in order to test its sensitivity to the assumed attack probability. The false alarm rate (FAR) and the average detection delay (ADD) are used as performance criteria and computed as follows: FAR $= \left(\sum_{j=1}^{10^4} \mathbb{1}_{\mathcal{H}_j}(u_s)\right)/10^4$, $i = 1$, ADD $= \left(\sum_{j=1}^{10^4} \min\{\ell : u(\ell) \in \mathcal{U}_j \wedge u(\ell) = u_s\}\right)/10^4$, $i = 2$, where $\mathcal{H}_j = \{i, u, g, \cdots\}$ is the history of j-th evaluation episode where $i = 1$ and $\mathcal{U}_j = \{u(0), u(1), \cdots\}$ is the history of control in j-th evaluation episode where $i = 2$.

C. Evaluation results

The detector has been tested with various ν_θ and some of the evaluation results are summarized here. Table I shows the effect of different window size w when $\nu_0 = 1$. The number of training episodes N is set to be 300 thousands. With the window size increases, the corresponding FAR decreases, which is expected as a better mean estimate can be obtained with w increasing. Table II lists the performance

under different costs with mean $\nu_0 = 0.7$. The number of training episodes N is set to be 1 million. It is shown that as the cost increases, the corresponding FAR increases. This is expected since the increase of the cost $g(1, u_c, 2)$ signals a bigger emphasis on minimizing detection delay during the training phase. Therefore, the FAR and ADD increases and decreases, respectively. In addition to the trade-off behavior

TABLE I: Performance under different window sizes with learning rate $\alpha = 0.005$ and cost $g(1, u_c, 2) = 0.001$.

w \ Type	1 sensor attacked		2 sensor attacked	
	FAR(%)	ADD	FAR(%)	ADD
2	14.83	14.3542	15.33	5.4918
3	4.51	8.8770	4.63	5.4022
4	1.25	7.8350	1.43	5.4208
5	0.27	8.5059	0.32	6.3201

TABLE II: Performance under different costs with window size $w = 6$ and learning rate $\alpha = 0.005$.

Cost \ Type	1 sensor attacked		2 sensors attacked	
	FAR(%)	ADD	FAR(%)	ADD
0.001	4.95	15.5038	4.98	7.8317
0.002	6.98	11.3086	6.75	7.2053
0.005	9.03	9.8902	9.71	6.8299
0.010	18.01	9.1583	18.16	6.4030

between FAR and ADD discussed above, the performance also varies with the number of sensors under attack. The FAR of 1 sensor attacked is almost similar to the one of 2 sensors attacked. The ADD of 1 sensor attacked is larger than the one of 2 sensors attacked. In other words, the detector spends longer time to judge whether there is an attack. In Fig. 4, we

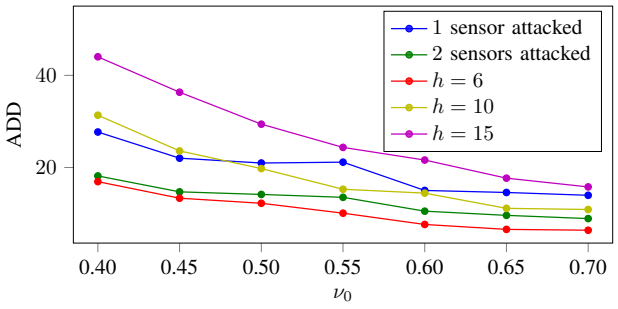

Fig. 4: ADDs of CUSUM algorithm and the resulting detector. Parameter h stands for the tuning parameter used in CUSUM.

compare the performance between our trained detector and classical quickest change detection algorithm (CUSUM) with different ν_0 [18] and we focus on comparing ADD, while all corresponding FAR are less than 5% in our methods. The blue and green lines denote ADD of 1 sensor attacked and 2 sensors attacked when choosing appropriate window size w, learning rate α and exploration parameters ε. The last three lines denote corresponding ADD when the threshold h is set as 6, 10, 15. Note that the CUSUM requires the true value of θ, which is not needed in our detector. From this figure, one can see that the resulting detector has comparable performance with classical QCD algorithm.

VI. Conclusion

In this paper, a detection problem of flip attacks is formulated as a POMDP by assuming an attack probability and a MDP in the form of SSP is employed to approximate the behavior of the POMDP by fixed-length window and state aggregation of observations. Then a standard Q-learning algorithm is applied to derive the optimal solution of the approximated MDP. Numerical results are provided to illustrate that the resulting detector exhibits promising behavior.

ACKNOWLEDGMENT

The authors are grateful to Prof. Xiaoqiang Ren from Shanghai University for the insightful comments.

References

[1] Y. Mo, R. Chabukswar, and B. Sinopoli, "Detecting integrity attacks on scada systems," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1396–1407, 2013.

[2] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE transactions on automatic control*, vol. 58, no. 11, pp. 2715–2729, 2013.

[3] D. Ding, Q.-L. Han, Y. Xiang, X. Ge, and X.-M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems," *Neurocomputing*, vol. 275, pp. 1674–1683, 2018.

[4] X. Ren and Y. Mo, "Secure detection: Performance metric and sensor deployment strategy," *IEEE Transactions on Signal Processing*, vol. 66, no. 17, pp. 4450–4460, 2018.

[5] Z. Li, Y. Mo, and F. Hao, "Game theoretical approach to sequential hypothesis test with byzantine sensors," *arXiv preprint arXiv:1909.02909*, 2019.

[6] M. Burmester, E. Magkos, and V. Chrissikopoulos, "Modeling security in cyber–physical systems," *International journal of critical infrastructure protection*, vol. 5, no. 3-4, pp. 118–126, 2012.

[7] M. N. Kurt, O. Ogundijo, C. Li, and X. Wang, "Online cyber-attack detection in smart grid: A reinforcement learning approach," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5174–5185, 2018.

[8] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, 2nd ed. MIT press, 2018.

[9] D. P. Bertsekas, *Reinforcement learning and optimal control*. Athena Scientific, 2019.

[10] Y. Chen, S. Huang, F. Liu, Z. Wang, and X. Sun, "Evaluation of reinforcement learning-based false data injection attack to automatic voltage control," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 2158–2169, 2018.

[11] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[12] V. V. Veeravalli and T. Banerjee, "Quickest change detection," in *Academic Press Library in Signal Processing*. Elsevier, 2014, vol. 3, pp. 209–255.

[13] A. G. Tartakovsky and V. V. Veeravalli, "General asymptotic bayesian theory of quickest change detection," *Theory of Probability & Its Applications*, vol. 49, no. 3, pp. 458–497, 2005.

[14] D. P. Bertsekas, *Abstract dynamic programming*, 2nd ed. Athena Scientific, 2018.

[15] D. P. Bertsekas and J. N. Tsitsiklis, "An analysis of stochastic shortest path problems," *Mathematics of Operations Research*, vol. 16, no. 3, pp. 580–595, 1991.

[16] S. Whitehead, "Reinforcement learning for the adaptive control of perception and action," Ph.D. dissertation, University of Rochester, 1992.

[17] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Cambridge, 1989.

[18] M. Basseville, I. V. Nikiforov *et al.*, *Detection of abrupt changes: theory and application*. prentice Hall Englewood Cliffs, 1993, vol. 104.