Temporal Attack Detection in Multimodal Cyber-Physical Systems with Sticky HDP-HMM

Andrew E. Hong[1], Peter P. Malinovsky[2] and Suresh Damodaran[2]

[1] *The MITRE Corporation, 7596 Colshire Drive, McLean, VA 22102*
[2] *The MITRE Corporation, 202 Burlington Road, Bedford, MA 01730-1420*

Abstract

Modern and legacy cyber-physical systems produce logs of operational behavior from sensors to network traffic; analyzing these heterogeneous logs to consistently identify attack signals is a difficult problem. In this work, we propose a flexible temporal non-parametric Bayesian framework for identifying these attacks based on sticky Hierarchical Dirichlet Process Hidden Markov Model (sHDP-HMM). The advantage of this approach is that it does not require detailed information on the system architecture, and it works for systems with unknown multimodal behavior, yielding interpretable inference. We demonstrate the efficacy of this framework for accurate identification of attacks from cyber and physical attack vectors on two different CPS: an avionics testbed and a consumer robot.

Keywords

cyber-physical systems, cybersecurity, machine learning, Bayesian nonparametrics

1. Introduction

Physical attacks form an important category of attacks on a cyber-physical system (CPS), for which the controller software manages the operations of a physical process using sensors and actuators [1]. Examples of CPSs include critical infrastructure components such as power plants and gas pipelines; vehicles on land, air, water, and space; building systems such as elevators; medical devices; and robots. However, many CPSs also utilize Internet of Things (IoT) devices, introducing additional vulnerabilities as well. Actors, from fraudsters to nation state actors, are conducting cyber attacks on CPSs with increased frequency and sophistication, resulting in substantial financial losses and, at times, the loss of life [2, 3].

Ruff et al. [4] refined the definition of anomaly provided by Hawkins [5] as "an observation that deviates considerably from some concept of normalcy." An attack on a CPS may be detected as an anomaly reflected in the logs from the system, sensors, actuators, or their interconnections. The automation of attack detection in CPSs is a task with many challenges. Damodaran and Rowe show that for any algorithm or anomaly detection approach, there is an inherent limitation in which certain adversarial

CAMLIS'22: Conference on Applied Machine Learning in Information Security (CAMLIS), October 20–21, 2022, Arlington, VA

ahong@mitre.org (A. E. Hong); pmalinovsky@mitre.org (P. P. Malinovsky); sdamodaran@mitre.org (S. Damodaran)

activities are fundamentally undetectable [6]. Further, irrespective of whether the normal operational behavior of a CPS is correctly and completely documented at design time, the operational behavior of the CPS may deviate after its deployment or be modified through lifecycle changes. Because many CPSs are systems of systems, composed of other CPSs, having definite and total knowledge of what "the concept of normalcy" from such deployed CPSs is frequently absent in practice.

To address this difficulty, we propose a machine learning approach that will learn this "concept of normalcy" from operational data. Because the methods and effects of adversaries are constantly evolving, the approach is unsupervised and is not fit to historical instances of attacks. Several unsupervised approaches to attack detection have been proposed previously [7]. However, we have identified through this research that many applications may require historical recordings of baseline behavior for making comparisons.

Most CPSs exhibit distinct operational behavior at different points in time that we refer to as *modes*. Whether an event is anomalous may depend entirely on the context provided by the mode of the system. For example, an aircraft may have the modes of take-off, cruising, and landing, and an action such as lowering the landing gear may be an anomaly while cruising but not while landing. The unknown of the system's mode and the dependency of the detection to these modes have motivated the approach proposed in this work. This work addresses the central challenge for attack detection where the system's current mode, number of total modes, and number of attacks are all unknown. In comparison, frameworks, such as finite state space filters, that presuppose this knowledge produce inaccurate results when these inputs are misspecified, see [8, 9].

Another motivating problem for this work is the need for flexible solutions that assimilate many heterogeneous forms of data streams. The time-series data recorded by CPSs may contain variables that are interval, e.g., location; ratio, e.g., altitude; ordinal, e.g., floor numbers; or nominal, e.g., commands [10]. Often in practice, anomalies are localized and require the examination of combinations of data streams to correctly identify them.

The primary goal of the attack detection algorithm is to segment the periods of attack from periods of normal behavior, by accurately determining the beginning and the end of each attack present [11]. This paper presents the modeling framework and inference algorithm to identify attacks in an unsupervised manner and demonstrates its efficacy on two types of systems: an avionics testbed and a consumer robot.

1.1. Contributions

Our contributions through this work are the following. First, we present a novel sticky Hierarchical Dirichlet Process Hidden Markov Model (sHDP-HMM) based framework and the associated inference algorithm for attack detection that is capable of detecting attacks reflected in multimodal and multivariate operational data of mixed continuous and categorical type.

Second, we evaluate this framework over experimental physical and cyber attacks conducted on two operational systems using a novel set of metrics. First, we analyzed

MIL-STD-1553 bus logs from an avionics testbed subject to a variety of attacks [12]. Second, we performed sensor and actuator attacks on a Roomba Create2 robot and evaluated the model's efficacy in identifying these events [13]. Our results from these evaluations are encouraging, and we are able to detect several attacks with limited false positives.

The rest of the sections are as follows. The next section discusses related work. Section 3 contains our sHDP-HMM framework for attack detection, followed by Section 4 which defines how temporal attack detection sequences are evaluated. Section 5 discusses the experiments with the avionics testbed, and Section 6 discusses the experiments with a consumer robot. Finally, Section 7 concludes this paper and discusses future work.

2. Related Work

The problem of anomaly detection in CPS under adversarial conditions has seen prolific research contributions as recent surveys indicate [1, 14, 3]. Machine learning techniques have been widely applied to the general problem of anomaly detection, and to the specific problem of attack detection. Neural networks have been applied to anomaly detection in [15, 16]. Bulusu et al. surveyed applications of deep learning for anomaly detection [17]. Ruff et al. provided a unified review of deep and shallow learning for anomaly detection [4]. The earliest published use of Hidden Markov Model (HMM) for intrusion detection was by Gao et al. in 2002 [18]. A more recent application of HMMs to anomaly detection for robotic actions was performed by Altan and Sariel [19]. Research into the problem of the lack of identifiability for the number of latent states in HMMs was conducted by [20, 21].

Teh et al. introduced the Hierarchical Dirichlet Process Hidden Markov Model (HDP-HMM) in [22]. Linderman et al. applied the HDP-HMM approach to study neural activity in rodents [23]. Blei et al. applied the HDP-HMM to application of topic modeling [24]. Fox et. al. observed that the original HDP-HMM formulation lacked stability for the latent state and introduced spurious transitions. The authors proposed remedying this problem through the introduction of an explicit self-transition parameter κ, formulating the sticky HDP-HMM (sHDP-HMM) model [25]. The sHDP-HMM was applied by Wu et al. to the application of monitoring in robotics [26].

More recently, the sHDP-HMM model has been applied to attack detection in avionics safety by Li et al. [27]. In their application, the authors analyze Automatic Dependent Surveillance - Broadcast (ADS-B) data with features such as longitude, latitude, altitude, and velocity and contrast latent state sequences against historical normal sequences to identify attacks. Our analysis also uses the sHDP-HMM approach, though we also incorporate continuous variable values and symbolic values from the captured system logs. Further, in contrast to their work, wherein the authors demonstrate the efficacy of attack detection using directly inserted anomalies into the operational logs, anomalies used to evaluate the methods in this work are captured by sensors in physical and communications layers.

Feedback control modeling of the physical dynamics of CPS has previously been

considered in [28, 1]; in particular, the authors consider the optimal stopping problem of hypothesis testing with minimal samples [29]. For these linear time invariant systems, the authors in [30, 8] treat the estimation of attack vectors as sparse recovery problems. The usage of log information for feature training of classifiers for attacks is well accepted [31]. In the context of avionics, Stan et al. demonstrated the utility log messages recorded on the MIL-STD-1553 standard conformant system bus in detecting adversarial behavior [32].

3. An sHDP-HMM Framework for Attack Detection

3.1. Convention & Notation

In this work, the notation $x|\theta$ is used to describe the distribution of the quantity x conditioned on or given a parameter θ; if x is the data, $x|\theta$ describes how the data is being generated given a set of parameters describing the observation process. In Bayesian hiearchical modeling, θ itself is considered a random quantity and in a hierarchical fashion there maybe a model describing its realization from a population $\theta|\eta$ governed by the parameter η. Wherein the quantity $\mathbf{P}(x|\theta)$ is referenced, it is the density function of the distribution $x|\theta$.

For a random time series, x_t refers to the random series at the point in time t, whereas $x_{u:v}$ refers to the entire collection between and including the points u and v and x_{-t} refers to the entire collection (from times 1 to T) excluding the point at time t. The superscript i in $x_t^{(i)}$ refers to different realizations or replications of x_t, identical in distribution.

In this work, the notion of a state is referenced, which is typically indexed over j. $\pi_{j,k}$ refers to the transition probability from j to k and $\pi_{j,\cdot}$ refers to the array of transition probabilities from state j (summing to one) and $\pi_{\cdot,k}$ refers to the array of transition probabilities from state k (not necessarily summing to one). In this work, z_t refers to the random state at time t and j denotes the specific state, for instance $\{z_t = j\}$ is the event that the system at time t is in state j.

3.2. Description

The operational behavior of a CPS may exhibit time varying modes. Our framework described in this section models operational behavior as latent states in a Hidden Markov model (HMM). The interest of this analysis is the identification of abnormal transitions between these latent states and potential identification of emergent anomalous behavior or new latent states during attack events. The latent states in normal operations may correspond to physical operational modes, but further research is needed to understand the nature of such correspondence. We hypothesize that under attack, the system may visit a non-finite number of novel latent states indicating anomalies. Figure 1 shows the transformation of a transition matrix of latent states under attacks in the avionics testbed described in Section 5.

We utilize the first order HMM model to identify these transitions between latent states. Because of the lack of identifiability of the number of latent states, we adopt the

(a) State transition probability matrix for normal operation

(b) State transition probability matrix under attack

Figure 1: Figure 1b depicts emergence of anomalous latent states, and change in transition probabilities during the attack, in contrast to normal behavior shown in Figure 1a. Recorded on the avionics testbed (see Section 5).

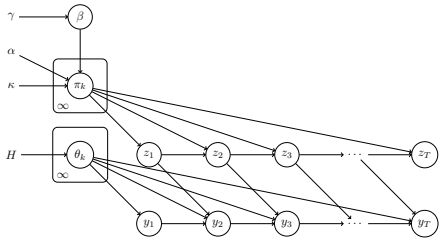


Figure 2: Graphical model with an upper layer for the latent state and a lower layer for the observed sequences.

non-parametric Bayesian approach where the latent states are informed by the data using the Dirichlet process. The data generated by the system $\{y_t\}_{t=1}^{T}$ may be a combination of continuous measurements recorded by sensors or tuples of symbolic configuration values. By assuming that the distribution and behavior of this data changes according to the latent state of the system at time t, which we label as z_t and consequently assume that the data is generated according to a member of a collection Θ - θ_{z_t}, we perform Bayesian inference to partition or cluster the data and estimate parameters.

We briefly review now the sHDP-HMM formulation by Fox et al. [25]. The HDP-HMM contains a global Dirichlet process (DP) $DP(\gamma H)$ where γ is a concentration parameter and H is a base measure. With γ, the GEM (Griffiths, Engen, and McCloskey) model may be described by the following stick-breaking procedure [33],

$$\beta'_{j'}|\gamma \sim \text{Beta}(1,\gamma) \tag{1}$$

$$\beta_i = \beta'_j \prod_{j<i}(1-\beta'_j) \tag{2}$$

Alongside this sequence $\beta = \{\beta_i\}$, θ_i may be drawn according to a base measure H over Θ and this global DP prior may be defined as $G = \sum_i \beta_i \delta_{\theta_i}$. The individual rows of

Figure 3: sHDP-HMM Framework for Attack Detection

the transition matrix between the latent states $\Pi = [\pi_{i,j}]$ have local DP priors $H(\alpha, \beta)$ where β allows for shared pooling of information.

The prior distribution for the transition probability to the next latent state given that the system is in the current state z_{t-1} is:

Global Dirichlet	$\beta\|\gamma$	$\sim \text{GEM}(\gamma)$	$i = 1, \ldots$
Process Prior:	θ_i	$\sim H$	
Transition Matrix Prior:	$\pi_{i,\cdot}$	$\sim DP(\alpha \cdot \beta + \kappa \cdot \delta_i)$	$i = 1, \ldots$
Latent State Transition:	$z_t\|z_{t-1}$	$\sim \pi_{z_{t-1},\cdot}$	$t = 1, \ldots, T$
Configuration Transition Prior:	$y_t\|y_{t-1}, z_{t-1}, p^{z_{t-1}}$	$\sim p^{z_{t-1}}_{y_{t-1},y_t}$	$t = 1, \ldots, T$

We propose a novel two-layer Markov chain model for the sequences of configuration emissions, where the transition probabilities $[p^z_{i,j}]$ varies depending on the latent state labeling of the system, allowing the pattern of sequences to vary as demonstrated in Figure 1. A graphical model representation of the proposed two-layer model is depicted in Figure 2.

Figure 3 summarizes our framework. The operational logs are pre-processed to identify unique combination of attribute values in each operational log to feed into the sHDP-HMM based inference algorithm (see Section 3.3). A detection heuristic is applied to the latent state transition matrix generated by the algorithm to output discrete temporal sequences of operation logs that correspond to attacks. The detection heuristic can vary based on the validation method used for detection, discussed in Section 3.4.

In the case where a known normal history of the system exists and is known to span the modes present in the system's behavior, the set of sufficient statistics estimated from this series $\hat{\theta}_0$ is a subset of the sufficient statistics estimated from the series containing the potential alternatives $\hat{\theta}_1$. Identifying the matching subset of $\hat{\theta}_0$ of $\hat{\theta}_1$ is a problem of matching the overlap of two high dimensional point clouds.

3.3. Inference Algorithm

In Algorithm 1, the inference implementation of the model uses the direct assignment Gibbs sample by Teh et al. [22]. The matrix N records the number of transitions between latent states, $N[j, k] = \sum_{1<t\leq T} \mathbb{1}\{z_{t-1} = j, z_t = k\}$. At each time instant, the new latent state labeling must be inferred by conditioning out the influence of its present labeling as shown in Line 8. The new labeling is sampled using the Chinese Restaurant Franchise analogy [25] and subsequently if a new state is realized the transition matrix is expanded (Line 6) and the associated summary statistics are updated (Line 8). Conversely, if a pre-existing latent state is found redundant, it is removed in Line 3, allowing the number

of latent states identified to decrease. Having fixed the latent state assignments, the remainder of the parameters in the model are conditionally updated. Algorithm 1 outputs the latent states and transition probability matrix among the latent states discovered in the operational data.

Algorithm 1: Direct Assignment Gibbs Sampler for sHDP-HMM

for $i = 1, \ldots, n$ **do**
  for $t = 1, \ldots, T$ **do**
    Decrement $N[z_{t-1}^{(i)}, z_t^{(i-1)}]$, $N[z_t^{(i-1)}, z_{t+1}^{(i-1)}]$
    Sample the state labeling $z_t^{(i)}$
    if $z_t^{(i)} = K^{(i)} + 1$ **then**
      Introduce state $K^{(i)} + 1$ into array $\beta^{(i)}$ and matrix N
      Increment $K^{(i)}$
    Increment $N[z_{t-1}^{(i)}, z_t^{(i)}]$, $N[z_t^{(i)}, z_{t+1}^{(i-1)}]$
  for $j = 1, \ldots, K^{(i)}$ **do**
    if $N_{j\cdot} = 0$ *and* $N_{\cdot j} = 0$ **then**
      Delete row and column j from N
  Update the count of unique states $K^{(i)} = |j : z_t^{(i)} = j \text{ for } t = 1, \ldots, T|$
  Sample the CRF auxiliary variable matrix $M^{(i)}$
  Sample the self-transition parameter(s)
  Sample the global weights $\beta^{(i)}$
  Sample the hyper-parameters

If $K^{(i)}$ is the number of latent states at iteration i, the number of operations required for iteration i is $\mathcal{O}(TK^{(i)})$ where T is the length of the data trace. Because each iteration requires a pass through the time series, Algorithm 1 may be expensive for massive data streams. In practice, we determine the convergence of the chain by examining the sequence of latent state labels $\{z_t^{(i)}\}$ for stability.

3.4. Detection Heuristic

Detection heuristics (see Figure 3) may be derived using the model output to contruct alarms. To determine whether an attack has occurred, there are a few possible approaches. If historical data from the system's normal operation is available, Algorithm 1 can be run on that data to identify the latent states and transition probabilities in normal behavior. This information may be contrasted against latent states and transition probabilities discovered from the attack data, as in Section 6. Alternately, if subject matter expertise is available, then specific rules-based alarms may be derived using the algorithm's output, as in Section 5. In the absence of historical data and domain knowledge, anomalies may be identified on the basis on the latent state frequency relative to the entire recording. However, the specific correspondence between these statistical anomalies and attacks and the points of differentiation between attacks and malfunctions and rare events are

subjects that require further exploration.

4. Metrics

One of the important ways of evaluating the detection framework is understanding the classification quality of detecting transitions from the "concept of normalcy" [4]. One aspect of the model output which can be used for comparison against other approaches is the ability to accurately detect the transition event from normality. This event detection may be formulated as a binary classification problem and evaluated with a confusion matrix.

However, the temporal nature of the observations is problematic for point-to-point evaluation of the classification quality. Slight delays in the time of detection should be differentiated from complete misses and the point-to-point measure for false positives may be multiplicative and overly punitive. These discrepancies and others issues relevant to other applications have been noted in [34, 16].

We present a set of metrics for evaluating detection of anomalous temporal discrete sequences. These metrics involve introducing two additional parameters h and ω, whose purpose is to partition the time series into periods of true positives and true negatives. We recommend these be selected to balance the labels present in the data set. ω is the length of time used to divide periods of non-detection into true negatives. Additionally, h serves to introduce tolerance for slight delays in detection. For detecting the beginning of attacks (similar analogues may be made towards detecting the end of attacks):

1. For a latent state transition at time t_i, if within the interval $[t_i, t_i + h]$, there are no transitions detected then the period $[t_i, t_i + h]$ is categoried as a **False Negative** (FP).
2. For a latent state transition at time t_i, if within the interval $[t_i, t_i + h]$, there is a transition detected at $\hat{t}_j \in [t_i, t_i + h]$ then the period $[t_i, t_j + h]$ is categoried as a **True Positive** (TP).
3. For a latent state non-transition at time t, if within the interval $[t, t + h]$, there are one or more estimated transitions $\{\hat{t}_j\} \in [t_i, t_i + h]$, then the period $[t, t + h]$ is categorized as **False Positive** (FP).
4. For a latent state non-transition at time t, if there are no estimated transitions, categorize the period $[t, t + h]$ as **True Negative** (TN).

5. Avionics Testbed

In a system with a bus architecture, the system components interact with each other only through a shared communication bus that employs a communication protocol. The MIL-STD-1553 standard, also referred to as *1553*, is a serial bus communication protocol standard used primarily in avionics systems [12]. This standard is used for communication among components, called remote terminals (RT) or devices. Examples of terminals are GPS receivers, auto-pilot controllers, or flight control components, such as ailerons,

elevators and rudders. In a 1553 compliant system, all devices are connected to a common wire, or multiple redundant wires, and most devices are slaves, except for a master device. The slave devices wait for queries from the master device, called the Bus Controller (BC). The master device commands a slave with its address, directionality, data sub-address, and data size to be sent [12]. This command is sent in a query called a command word, which provides specific direction for the slave to act. Our avionic testbed consists of a physical 1553 bus, and several components. Each component is a software module that used an Alta eNet-1553 interface device [35] to interact with the 1553 bus.

5.1. Attacks

Attacks are conducted using virtualized flight components interacting with the 1553 bus through the Alta eNet interface. Our BC cycles through lists of messages to transact, producing a deterministic and periodic pattern. The 1553 protocol allows any device connected to the bus masquerade as a BC, allowing a bad actor to hijack the bus. We conducted 17 attacks on the system with differing complexity, ranging from noise attacks to denial-of-service. See Table 2 for details of the attacks.

5.2. Detection

The operational data is collected from the 1553 bus using the Alta eNet-1553 interface device. In the pre-processing step, to represent the state of the bus, we used the symbolic message attributes as outlined in Table 1.

Field	Value Ranges	Variable Type
Message Type	BC to RT, RT to BC, RT to RT, mode command, mode command with transmit data, and mode command with receive data	Categorical
Remote Terminal Address	0 - 31	
Transmit/Receive	0, 1	
Subaddress/Mode	0 - 31	
Data Word Count/Mode Code	0 - 31	

Table 1
MIL-STD-1553 Message Attributes

Algorithm 1 is used to detect latent state transitions on the collected data. To derive the detection heuristic, we observe that the system exhibited only a single latent state, as shown in Figure 4, and confirmed by the experimenter on the timing of attacks. The operational data is collected only on 16 attacks, since Attack 13 is not reflected in the operational logs. Therefore, our detection heuristic in this case is simple; any latent state transition to a new latent state and return to the single stable latent state is an attack. The results are in Table 2.

In order to evaluate the detection quality of the inference for the experiment, we use the procedure described in Section 4 with $\omega = 350$. Selecting $\omega = 350$ and $h = 1$ divides the time series into 32 intervals, half of which are true positives and the other

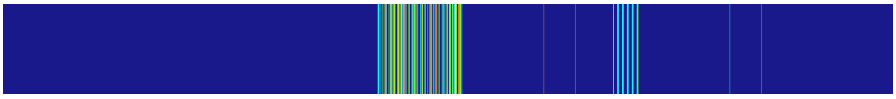


Figure 4: Latent state estimation of system recorded on the MILD-STD-1553 bus. Normal operation of the system contains only one mode - any departure from this single mode of behavior is considered to be an attack.

Attack	Attack Occurrence	Detected Occurrence	Detection	Description
Attack 0	3451 - 4248	3451 - 4248	TP	Denial of Service 1
Attack 1	4538	4538	TP	Noise Attack 1
Attack 2	4568	4568	TP	Noise Attack 2
Attack 3	4714	4714	TP	Noise Attack 3
Attack 4	4860	4860	TP	Noise Attack 4
Attack 5	5006	5006	TP	Protocol Violation 1
Attack 6	5152	5152	TP	Protocol Violation 2
Attack 7	5298	5298	TP	Protocol Violation 3
Attack 8	5444	5443 - 5445	TP	Protocol Violation 4
Attack 9	5590 - 5968	5600, 5647-5700, 5740-5745, 5773-5789, 5818-5834, 5863-5879, 5908-5911	TP	Denial of Service 2
Attack 10	6114	6114	TP	Buffer Attack 1
Attack 11	6405	6405	TP	Buffer Attack 2
Attack 12	6551	6551	TP	Anomalous Traffic 1
Attack 13	N/A	N/A	FN	Atypical Traffic
Attack 14	6726	None	FN	Anomalous Traffic 2
Attack 15	6872	6872	TP	Anomalous Traffic 3
Attack 16	7018	7018 - 7019	TP	Data Payload Attack

Table 2
Detection of attacks on the avionics testbed based on metrics in Section 4. TP is short for true positive; FN - false negative; TN - true negative; FP - false positive.

half true negatives. We conclude that there are 15 true positives, 0 false positives, 16 true negatives, and 2 false negatives. A human expert analysis of sample traffic collected during the DDoS Attack 9 revealed multiple periods of normal traffic interlacing the anomalous measurements, which may explain the fragmentation.

6. Consumer Robot

In this section, we describe detection of attacks on a consumer robot. This system is a Roomba Create2 [13], and the experiment set-up is described in detail in a previous paper [36]. This system is different from the avionics testbed described in Section 5 in the following significant way. The robot has physical sensors and actuators and is a fully functioning CPS. The attacks in this experiment are directly to the physical sensors and actuators. The log attributes used in this experiment for pre-processing are summarized in Table 3.

6.1. Attacks

As summarized in Table 4, in the sensor attack, an obstruction is introduced in front of the left optical sensor, causing the Roomba to prematurely halt traveling towards its destination. Outside intervention removes this obstruction, allowing the Roomba to continue its course. In the actuator attack, the robot is physically held in place on a slippery surface. The wheels continue to spin; however, the robot does not make any forward movement. Outside intervention removes the restraint on the robot, and it is allowed to move forwards.

Field	Value Ranges	Variable Type
Voltage	0 - 65535	Continuous
Current	-32768 - 32767	
Velocity Left	-500 - 500	
Velocity Right	-500 - 500	
Light Bumper Front Left	0 - 4095	
Light Bumper Front Right	0 - 4095	

Table 3
Roomba Attributes

6.2. Attack Detection

Experiment	Attack Vector	Data	Attack Occurrence	Detected Occurrence	Start Attack	End Attack
1	wall sensors	light bumpers, velocity	(58, 59) - (74, 75)	62 - 73	TP	TP
2	actuators	current, voltage	(171, 172) - (212, 217)	172 - 192	TP	FN

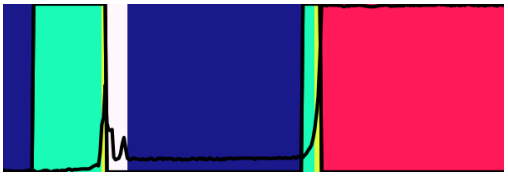
Table 4
Table summarizing the results of the experiments, attacking the position/motion sensors and the actuators of the Roomba. In both cases, novel states that emerge in the attack recordings are then compared to recordings collected during normal operation. The location of these novel states is compared against the ground truth of the attacks to determine if detection is successful with tolerance $h = 5$ (defined in Section 4) and ω unspecified as the sample size is too limited to report the accuracy statistics.

In contrast to the avionics testbed that has only a single latent state, or mode, detectable by Algorithm 1, the Roomba Create2 shows multiple modes, as shown in Figure 5. The different colors indicate different latent states, and the solid black line is the wall sensor output that measures distance to an obstruction, see Table 4.

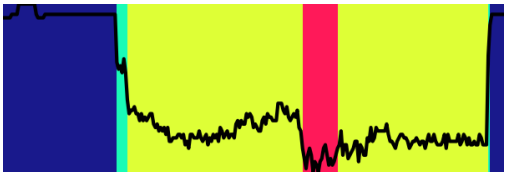
In both attack experiments, comparison between the number of latent states in recordings of normal traffic and attack traffic identified more latent states in the later. Because each latent state has associated sufficient statistics for the sensors, it is straightforward to draw correspondences between the latent states in each set of recordings and identify which latent states are novel with respect to normal behavior. Having identified the novel latent states in the attack recordings, we analyze the mode of the start of the transition to these states and compare it against expert annotation to determine the accuracy of the start/end of attack detection.

(a) Wall sensors and states for the Roomba under normal operation.

(b) Current readings and states for the Roomba under normal operation.



(c) Wall sensors and states for the Roomba under sensor attack.



(d) Current readings and states for the Roomba under actuator attack.

Figure 5: Plot of Roomba sensor readings and latent state estimates under normal operation (top row) and under attack (bottom row) during experiments. Top row displays the wall sensor readings during Experiment 1 in Table 4 in which the wall sensors are attacked, and bottom row displays the current (ampere) readings during the Experiment 2 in Table 4 in which the wheel actuators are attacked.

7. Conclusion & Future Work

We presented in this paper an unsupervised learning framework based on the sHDP-HMM model and detection heuristic to detect attacks on two types of systems using different attack approaches. In both cases, no domain knowledge of the system architecture was used to infer the location and number of attacks in the time series. This unsupervised approach for attack detection is applicable for many systems for which the only available information consists of the operations logs. We demonstrated the ability of this framework to assimilate heterogeneous data consisting of continuous sensors and categorical configurations. While these preliminary results are encouraging, more testing on a wider variety of systems is needed to better understand the limitations and wider applicability of this approach.

We surmise that the algorithm's detection is most accurate if the entire attack is contained within the data recording. It is unresolved how partial capture of the attack will impact the detection accuracy. Further, the exploration of the detection of cyberattacks on the avionics testbed has identified instances of known cyberattacks, whose effects are absent from the system bus logs.

A key challenge for machine learning approaches for attack detection in CPSs is the explainability of the results to non-experts and was a key motivator for this approach [37]. Although the model presented here as a switching process between an ensemble of simple models is more transparent in comparison to alternatives, there is potential to better integrate these outputs with other sources of data and physical laws to provide greater clarity into the causality of anomalies. Additionally, although we considered the

many common types of tabular data in this work, the framework presented here may also assimilate many forms of higher fidelity data such audio and video. Lastly, because Algorithm 1 performs multiple passes through the data, it poses an interesting challenge to scaling up the approach to the massive streams of data encountered in real-time monitoring systems[38]. Addressing this challenge of scalability is an area of research we are actively investigating.

References

[1] J. Giraldo, D. Urbina, A. Cardenas, J. Valente, M. Faisal, J. Ruths, N. O. Tippenhauer, H. Sandberg, R. Candell, A survey of physics-based attack detection in cyber-physical systems, ACM Computing Surveys 51 (2018) 1–36.

[2] R. Alguliyev, Y. Imamverdiyev, L. Sukhostat, Cyber-physical systems and their security issues, Computers in Industry 100 (2018) 212–223.

[3] A. Chowdhury, G. Karmakar, J. Kamruzzaman, Survey of recent cyber security attacks on robotic systems and their mitigation approaches, in: Cyber Law, Privacy, and Security: Concepts, Methodologies, Tools, and Applications, IGI Global, 2019, pp. 1426–1441.

[4] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, K.-R. Müller, A unifying review of deep and shallow anomaly detection, Proceedings of the IEEE 109 (2021) 756–795.

[5] D. M. Hawkins, Identification of outliers, volume 11, Springer, 1980.

[6] S. K. Damodaran, P. D. Rowe, Limitations on observability of effects in cyber-physical systems, in: Proceedings of the 6th Annual Symposium on Hot Topics in the Science of Security, 2019, pp. 1–10.

[7] A. Nisioti, A. Mylonas, P. D. Yoo, V. Katos, From intrusion detection to attacker attribution: A comprehensive survey of unsupervised methods, IEEE Communications Surveys & Tutorials 20 (2018) 3369–3388.

[8] Y. H. Chang, Q. Hu, C. Tomlin, Secure estimation based kalman filter for cyber-physical systems against sensor attacks, Automatica 95 (2018) 399–412.

[9] L. Daria, Z. Dmitry, Y. Anastasiia, Predicting cyber attacks on industrial systems using the kalman filter, in: 2019 Third World Conference on Smart Trends in Systems Security and Sustainablity (WorldS4), IEEE, 2019, pp. 317–321.

[10] A. Blázquez-García, A. Conde, U. Mori, J. A. Lozano, A review on outlier/anomaly detection in time series data, ACM Computing Surveys (CSUR) 54 (2021) 1–33.

[11] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection for discrete sequences: A survey, IEEE transactions on knowledge and data engineering 24 (2010) 823–839.

[12] R. Schuh, An overview of the 1553 bus with testing and simulation considerations, in: 1988. IMTC-88. 5th IEEE Instrumentation and Measurement Technology Conference, 1988, pp. 20–25. doi:10.1109/IMTC.1988.10811.

[13] iRobot, iRobot Create 2 open interface (OI) specification based on the iRobot Roomba 600, 2018. URL: https://edu.irobot.com/what-we-offer/create-robot, accessed July 20, 2022.

[14] D. Ding, Q.-L. Han, Y. Xiang, X. Ge, X.-M. Zhang, A survey on security control and attack detection for industrial cyber-physical systems, Neurocomputing 275 (2018) 1674–1683.
[15] Y. Wang, M. Perry, D. Whitlock, J. W. Sutherland, Detecting anomalies in time series data from a manufacturing system using recurrent neural networks, Journal of Manufacturing Systems (2020).
[16] N. Singh, C. Olinsky, Demystifying numenta anomaly benchmark, in: 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 1570–1577.
[17] S. Bulusu, B. Kailkhura, B. Li, P. K. Varshney, D. Song, Anomalous example detection in deep learning: A survey, IEEE Access 8 (2020) 132330–132347.
[18] B. Gao, H.-Y. Ma, Y.-H. Yang, Hmms (hidden markov models) based on anomaly intrusion detection method, in: Proceedings. International Conference on Machine Learning and Cybernetics, volume 1, IEEE, 2002, pp. 381–385.
[19] D. Altan, S. Sariel, What went wrong? Identification of everyday object manipulation anomalies, Intelligent Service Robotics 14 (2021) 215–234.
[20] P. Smyth, Model selection for probalistic clustering using cross-validation likelihood, Statistics and Computing 10 (2000) 63–72.
[21] G. Celeux, J.-B. Durand, Selecting the hidden markov model state number with cross-validated likelihood, Computational Statistics 23 (2008) 541–564.
[22] Y. W. Teh, M. I. Jordan, M. J. Beal, D. M. Blei, Hierarchical dirichlet processes, Journal of the American Statistical Association 101 (2006) 1566–1581.
[23] S. W. Linderman, M. J. Johnson, M. A. Wilson, Z. Chen, A bayesian nonparametric approach for uncovering rat hippocampal population codes during spatial navigation, Journal of Neuoscience Methods 263 (2016) 36–47.
[24] D. M. Blei, T. L. Griffiths, M. I. Jordan, The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies, Journal of the ACM 57 (2010).
[25] E. B. Fox, E. B. Sudderth, M. I. Jordan, A. S. Willsky, A sticky HDP-HMM with application to speak diarization, The Annals of Applied Statistics 101 (2011) 1020–1056.
[26] H. Wu, Y. Guan, J. Rojas, A latent state-based multimodal execution monitor with anomaly detection and classification for robot introspection, Applied Sciences 9 (2019).
[27] T. Li, B. Wang, F. Shang, J. Tian, K. Cao, Dynamic temporal ADS-B data attack detection based on shdp-hmm, Computers & Security 93 (2020) 101789.
[28] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, 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, 2011, pp. 355–366.
[29] T. Kaliath, H. V. Poor, Detection of stochastic processes, IEEE Transactions on Information Theory 44 (1998) 2230—-2258.
[30] H. Fawzi, P. Tabuada, S. Diggavi, Secure estimation and control for cyber-physical systems under adversarial attacks, IEEE Transactions on Automatic Control 59 (2014) 1454–1467.

[31] D. Hadžiosmanović, L. Simionato, D. Bolzoni, E. Zambon, S. Etalle, N-gram against the machine: On the feasibility of the n-gram network analysis for binary protocols, in: D. Balzarotti, S. J. Stolfo, M. Cova (Eds.), Research in Attacks, Intrusions, and Defenses, 2012, pp. 354–373.

[32] O. Stan, Y. Elovici, A. Shabtai, G. Shugol, R. Tikochinski, S. Kur, Protecting military avionics platforms from attacks on MIL-STD-1553 communication bus, CoRR abs/1707.05032 (2017). URL: http://arxiv.org/abs/1707.05032. arXiv:1707.05032.

[33] J. Pitman, Poisson–dirichlet and gem invariant distributions for split-and-merge transformations of an interval partition, Combinatorics, Probability, and Computing 11 (2002) 501—-514.

[34] X. Zhou, A. Del Valle, Range based confusion matrix for imbalanced time series classification, in: 2020 6th Conference on Data Science and Machine Learning Applications (CDMA), 2020, pp. 1–6. doi:10.1109/CDMA47397.2020.00006.

[35] Alta-dt, ENET-1553: MIL-STD-1553 REAL-TIME ETHERNET CONVERTER, ???? URL: https://www.altadt.com/product/enet-1553/, accessed July 20, 2022.

[36] A. Roque, M. Lin, S. Damodaran, Cybersafety analysis of a natural language user interface for a consumer robotic system, in: European Symposium on Research in Computer Security, Springer, 2021, pp. 107–121.

[37] A. Roque, S. K. Damodaran, Explainable ai for security of human-interactive robots, International Journal of Human–Computer Interaction (2022) 1–19.

[38] A. Bifet, R. Gavalda, G. Holmes, B. Pfahringer, Machine learning for data streams: with practical examples in MOA, MIT press, 2018.