A Survey of Intrusion Detection Techniques for Cyber-Physical Systems

ROBERT MITCHELL and ING-RAY CHEN, Virginia Tech

Pervasive healthcare systems, smart grids, and unmanned aircraft systems are examples of Cyber-Physical Systems (CPSs) that have become highly integrated in the modern world. As this integration deepens, the importance of securing these systems increases. In order to identify gaps and propose research directions in CPS intrusion detection research, we survey the literature of this area. Our approach is to classify modern CPS Intrusion Detection System (IDS) techniques based on two design dimensions: detection technique and audit material. We summarize advantages and drawbacks of each dimension's options. We also summarize the most and least studied CPS IDS techniques in the literature and provide insight on the effectiveness of IDS techniques as they apply to CPSs. Finally, we identify gaps in CPS IDS research and suggest future research areas.

Categories and Subject Descriptors: Security and Privacy [**Intrusion/Anomaly Detection and Malware Mitigation**]: Intrusion Detection Systems

General Terms: Security

Additional Key Words and Phrases: Cyber-physical systems, classification, intrusion detection, security

ACM Reference Format:
Robert Mitchell and Ing-Ray Chen. 2014. A survey of intrusion detection techniques for cyber-physical systems. ACM Comput. Surv. 46, 4, Article 55 (March 2014), 29 pages.
DOI: http://dx.doi.org/10.1145/2542049

1. INTRODUCTION

Cyber-Physical Systems (CPSs) are large-scale, geographically dispersed, federated, heterogeneous, life-critical systems that comprise sensors, actuators, and control and networking components. First responder situational awareness systems, pervasive health care systems, smart grids, and unmanned aircraft systems are some examples of CPSs. These systems have multiple control loops, strict timing requirements, predictable network traffic, legacy components, and possibly wireless network segments. CPSs fuse cyber (comprising network components and commodity servers) and physical (comprising sensors and actuators) domains.

The attack model for CPSs encompasses short and long duration attacks. A reckless adversary can enter the network and immediately disrupt the concerned processes to cause a catastrophe. On the other hand, a more sophisticated adversary may take care to not disrupt normal system operation in order to propagate and set up a distributed attack launched at one point in time. This is the brand of attack that Stuxnet used [Keizer 2010; Stuxnet 2013]. For this reason, speed of detection (detection latency) is the key challenge in CPS Intrusion Detection System (IDS) design. The focus for CPS IDS design is leveraging their unique traits and detecting unknown attacks.

Authors' addresses: Robert Mitchell, 7054 Haycock Road, Falls Church, VA 22043; email: rrmitche@vt.edu; Ing-Ray Chen, 7054 Haycock Road, Falls Church, VA 22043; email: irchen@vt.edu.

DOI: http://dx.doi.org/10.1145/2542049

This article surveys IDS design principles and techniques for CPSs. In particular, we classify existing CPS IDS techniques in the literature, discuss their merits and drawbacks, summarize strengths and weaknesses in intrusion detection research, and suggest future research areas.

The rest of the article is organized as follows. Section 2 discusses the core functionality of intrusion detection in CPSs. Section 3 provides a classification tree for organizing existing CPS IDS protocols and explains the dimensions used for CPS IDS classification. Section 4 surveys the CPS intrusion detection literature and classifies existing CPS IDS techniques grouped by the application domain. In Section 5, we first summarize advantages and drawbacks of existing CPS IDS techniques and the most and least studied CPS IDS techniques in the literature. Then, we provide insight on the effectiveness of IDS techniques as applying to CPSs and identify research gaps that are worthy of further research efforts. Section 6 presents our conclusion and suggests future research directions.

2. CPS IDS FUNCTIONS AND METRICS

2.1. Cyber-Physical Systems

Securing CPSs has emerged as a critical interest of all governments. The literature also refers to a CPS as a Distributed Control System (DCS), Networked Control System (NCS), Sensor Actuator Network (SAN), or Wireless Industrial Sensor Network (WISN) [Shin et al. 2010]. In addition, Supervisory Control And Data Acquisition (SCADA) is a subgroup of CPS. Their functions in common are sensing (acquisition) and actuation (control). These systems may have wireless segments and are heterogeneous and geographically dispersed. These systems may be federated, mobile, attended, or completely inaccessible. *Enclaves* define the edges of the segments of the federated system. Nodes that contain the sensors and actuators are called Remote Terminal Units (RTUs), Intelligent Electronic Devices (IEDs), or Programmable Logic Controllers (PLCs). RTUs may implement some limited tactical control functions. Data Acquisition Systems (DASs) aggregate readings from RTUs and adapt (bridge or tunnel) the local RTU protocol (such as Controller Area Network [CAN] [ISO 11898 2003], Distributed Network Protocol (DNP3) [DNP3 2010], or Modbus [Modbus Messaging 2006; Modbus Application 2012]) with the long-haul protocol shared with the control center (such as Transmission Control Protocol [TCP]). Data processing servers effect the business logic of the CPS; these may be high-performance computing clouds that process large datasets produced by economical nodes. Historian servers collect, store, and distribute data from sensors [Rockwell Automation Technologies, Inc. 2009]. Nodes that contain control logic and provide management services to a Human Machine Interface (HMI) are called Master Terminal Units (MTUs); in contrast with the RTUs, an MTU implements the broad strategic control functions. Figure 1 illustrates a typical CPS using these components.

Common CPS issues are availability, reconfigurability, distributed control (distributed management), real-time operation (timeliness), fault-tolerance, scalability, autonomy, reliability, security, heterogeneity, federation, and geographic dispersion [National Science Foundation 2011]. Timeliness is critical in CPSs because the situation can change quickly [Chen et al. 2011; Al-Hamadi and Chen 2013]; control loops fail if their period is longer than expected. Automatic control techniques can address CPS reliability. However, security requires distinct measures from reliability. Moreover, compromised nodes may collude to deter or disrupt the CPS functionality. An effective yet energy-efficient IDS is of great interest to detect and evict compromised nodes from a CPS whose failure can cause dire consequences.

Figure 2 illustrates a hierarchical abstraction model for a federated CPS. It represents all of the key CPS artifacts introduced: enclaves, sensors, actuators, RTUs,

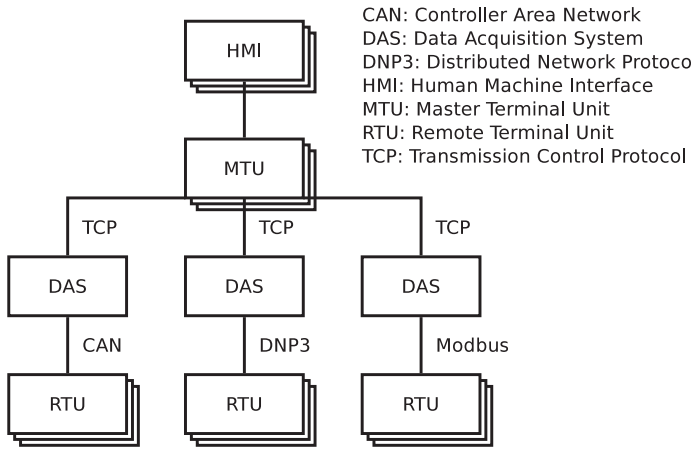

Fig. 1. A typical CPS architecture.

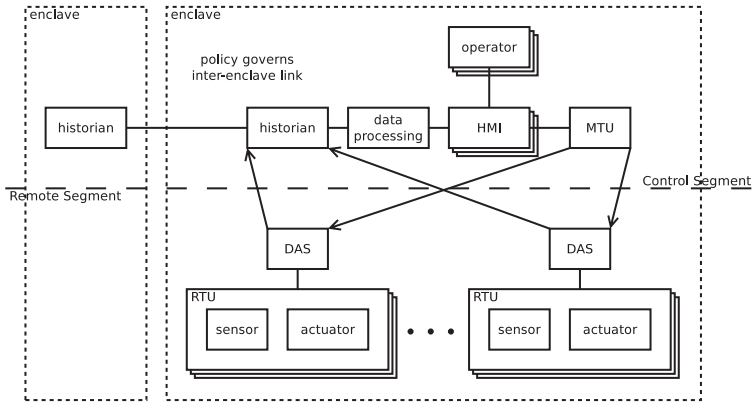

Fig. 2. Hierarchically structured CPS abstraction model.

DASs, MTUs, data processing servers, historian servers, HMIs, operators, and communications links. RTUs comprise sensors and actuators interconnected via local high speed network or bus links. In turn, they are managed by a DAS, which bridges the gap between the remote and control segments with a long distance wireless link. They escalate sensor data to the historian server and receive control messages from the MTU. Operators use HMIs to read the sensor data in the colocated historian and exploit it with the assistance of colocated data processors. Multiple enclaves compose the CPS; highly scrutinized business rules govern the exchange of data between historian servers.

2.2. Core Intrusion Detection Functions

A CPS IDS implements two core functions:

—Collecting data regarding suspects
—Analyzing the data

Data collection is the process by which a CPS accumulates audit data; the result is one or more binary or human-readable files or databases. Examples of collection are logging

system calls on the local node, recording traffic received on a network interface, and gathering hearsay reputation scores. Data analysis is the process by which a CPS audits the collected data; the result can be binary (bad/good), ternary (bad/good/inconclusive), or continuous (between 0 and 100% bad probability). Examples of analysis are pattern matching, statistical analysis, and data mining.

2.3. Intrusion Detection Performance Metrics

IDS researchers traditionally use three metrics to measure performance: False-Positive Rate (FPR), False-Negative Rate (FNR), and its complement, True Positive Rate (TPR). A false negative occurs when an IDS misidentifies a malicious node as well behaved. The literature refers to a false negative as a failure to report and refers to the inverse of FNR as completeness. On the other hand, a detection (a true positive) occurs when an IDS correctly identifies a malicious node. Finally, a false positive occurs when an IDS misidentifies a well-behaved node as an intruder. The literature also refers to a false positive as a false alarm and refers to the inverse of FPR as accuracy. In the literature, FPR is the same as false-positive probability p_{fp}, and FNR is the same as false-negative probability p_{fn}. Consequently $\text{TPR} = 1 - \text{FNR} = 1 - p_{\text{fn}}$. In this article, we will simply use the notations p_{fn} and p_{fp} to refer to FNR and FPR, respectively. When we need to refer to true positive rate, we will use the acronym TPR. It is customary to rate IDS performance by a Receiver Operating Characteristic (ROC) graph—that is, a detection rate versus FPR plot.

Some research attempts to establish effective new metrics in order to enrich IDS research. Detection latency is a rarely used but critical means to measure IDS performance [Striki et al. 2009]. This measures the time interval between an adversary penetrating the protected system (for an insider) or beginning their attack (for an outsider) and the IDS identifying the adversary. For target systems with resource limitations, power consumption, communications overhead, and processor load are important metrics as well. Packet sampling efficiency is the percentage of analyzed packets the IDS identifies as malicious; the basic idea is that it is wasteful to sample lots of packets when only a few trigger an intrusion detection [Misra et al. 2010].

Sommer and Paxson [2010] and McHugh [2000] provide extensive insight on how difficult it is to provide good measurements for IDSs.

2.4. Distinguishing Characteristics of CPS Intrusion Detection

CPS intrusion detection addresses the embedded physical components and physical environment in a CPS, which when under attacks manifest physical properties and normally require a closed control loop to react to physical manifestation of attacks. As illustrated in Table I, we summarize four major differences between CPS intrusion detection and the same function for traditional Information and Communications Technology (ICT) systems:

—*Physical Process Monitoring (PPM)*: While an ICT IDS may monitor host- or network-level user/machine activity (e.g., an HTTP request or a web server), a CPS IDS measures physical properties. In particular, a CPS IDS monitors the physical processes (and hence laws of physics) that govern behavior of physical devices that make certain behaviors more likely to be seen than others.

—*Closed Control Loops (CCL)*: The activities in a CPS environment are frequently automated and time driven in a closed-loop setting, thus providing some regularity and predictability for behavior monitoring. This is as opposed to ICT environments in which activities are user triggered, thus leading to unacceptably high FPRs due to the unpredictability of user behaviors. This CPS predictability is therefore a research opportunity to revisit behavior-based approaches.

Table I. Differences between ICT and CPS Intrusion Detection

ICT	CPS
An ICT IDS monitors host- or network-level user/machine activity (e.g., an HTTP request or a web server).	A CPS IDS monitors the physical processes (and hence laws of physics) that govern behavior of physical devices that make certain behaviors more likely to be seen more than others.
An ICT IDS monitors user-triggered activities, leading to unacceptably high false-positive rates due to the unpredictability of user behaviors.	A CPS IDS monitors activities that are frequently automated and time driven in a closed-loop setting, thus providing some regularity and predictability for behavior monitoring.
An ICT IDS deals with mostly non-zero-day attacks, rendering knowledge-based detection effective.	A CPS IDS deals with zero-day or highly sophisticated attacks, rendering knowledge-based detection ineffective.
An ICT IDS often does not have to deal with legacy components, making behavior specification of the physical processes governing legacy components unnecessary.	A CPS IDS often must deal with legacy technology, making behavior-specification–based detection an effective technique by precisely specifying the physical processes governing behavior of legacy components.

—*Attack Sophistication (AS)*: The payoff for a successful attack against a CPS is substantial. By jeopardizing the lives of hundreds of patients in a hospital or denying service to millions of utility customers, a rival state gains a strong lever to change the policy of the subject nation. By exfiltrating collected data products or operational plans of the subject military or the personally identifying information (PII) of civilians, a rival nation or group of financially motivated criminals score an intelligence victory. The high payoff would lead to an increase in attack sophistication and to the extensive use of zero-day attacks (as we have seen in Stuxnet).
—*Legacy Technology (LT)*: Many CPS environments operate with legacy hardware that is difficult to modify or physically access. Many physical components in CPSs, especially legacy physical components based on mechanical or hydraulic control, do not have software installed, and their behavior is essentially governed by the physical processes. The challenge is to identify environment variables, define environment changes in terms of environment variable changes, and incorporate the laws of physics to define acceptable behavior upon environment changes. This particularly makes behavior-specification–based detection more suitable for CPS IDS, because the physical processes can be defined more precisely by behavior specifications for individual physical components.

3. CLASSIFICATION TREE

In this section, we develop a classification tree for organizing existing CPS IDS techniques to identify research gaps in CPS IDS research based on the taxonomy established by [Debar et al. 2000]. Figure 3 shows our classification tree based on two classification dimensions:

(1) *Detection Technique*: This criterion defines "what" misbehavior of a physical component the IDS looks for to detect intrusions.
(2) *Audit Material*: This criterion defines "how" the IDS collects data before data analysis.

Next we discuss each classification dimension in detail.

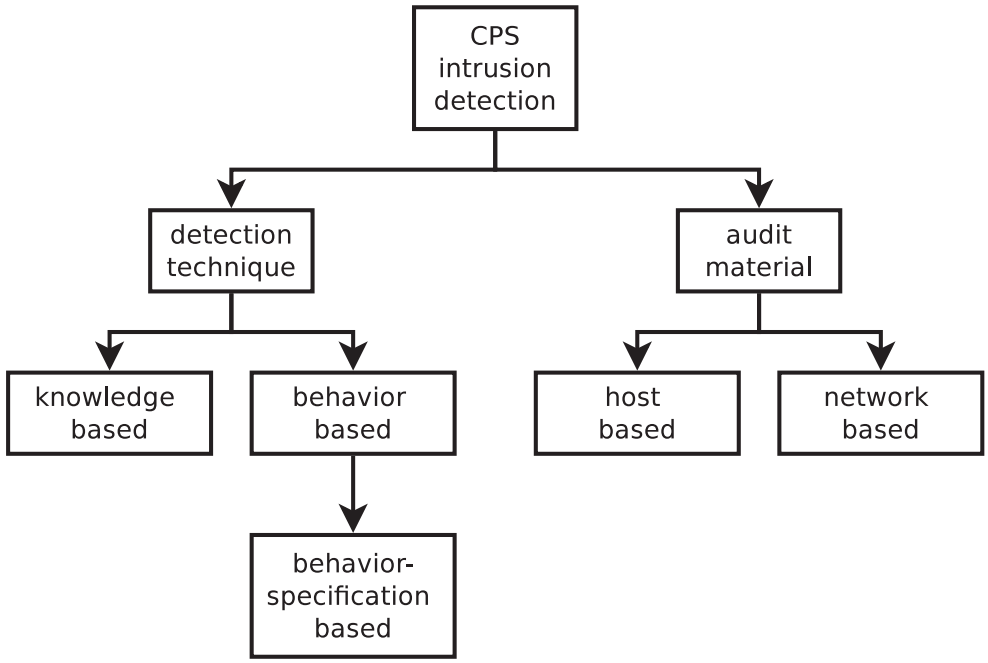

Fig. 3. A classification tree for intrusion detection techniques for CPSs.

3.1. Detection Technique

Existing CPS IDS detection techniques include knowledge and behavior-based techniques.

3.1.1. Knowledge-Based Intrusion Detection. Knowledge-based intrusion detection approaches look for runtime features that match a specific pattern of misbehavior [Whitman and Mattord 2011]. Some sources refer to this approach as misuse detection [Han et al. 2002; Foo et al. 2005; Haddadi and Sarram 2010; Ying et al. 2010], supervised detection [Zhong et al. 2005], pattern-based detection [Farid and Rahman 2008], or intruder profiling [White et al. 1996].

One major advantage of this category is a low FPR. By definition, these approaches only react to known bad behavior; the basic idea is that a good node will not exhibit the attack signature. The key disadvantage of this category is that the techniques must look for a specific pattern; a dictionary must specify each attack vector and stay current. An attack signature can be a univariate data sequence: for example, bytes transmitted on a network, a program's system call history, or application-specific information flows (e.g., sensor measurements). One sophistication is to combine simple data sequences into a multivariate data sequence. The important research problem in knowledge-based intrusion detection is creating an effective attack dictionary.

It is worth noting that knowledge and signature-based designs are not synonymous: some knowledge-based IDSs do not use a signature-based implementation, and some behavior-based IDSs do.

3.1.2. Behavior-Based Intrusion Detection. Behavior-based intrusion detection approaches look for runtime features that are out of the ordinary [Whitman and Mattord 2011]. The ordinary can be defined with respect to the history of the test signal (unsupervised)

[Hinton and Sejnowski 1999] or with respect to a collection of training data (semisupervised) [Chapelle et al. 2006]. Unsupervised approaches train with live data. Clustering is an example of unsupervised machine learning. Semisupervised approaches train with a set of truth data. Researchers take different approaches for discrete, continuous, and multivariate datasets. Examples of a discrete dataset are dialed numbers or system state; Longest Common Subsequence (LCS) can be applied to discrete data over an interval, whereas Hamming distance can be applied to discrete data instantaneously [Park et al. 2010; Cormen et al. 2001]. Position and data rate are examples of continuous datasets; this type of data calls for a system of thresholds since exact matches will be rare. An example of a multivariate dataset is a three-tuple of position, Received Signal Strength Indication (RSSI), and time; machine learning approaches (e.g., genetic programming [Gong et al. 2009], clustering [Ni and Zheng 2007], neural networks [Ali et al. 2009] and Bayesian classifiers [Luo 2010]) are useful for this brand of data.

The key advantage of behavior-based approaches is they do not look for something specific. This eliminates the need to fully specify all known attack vectors and keep this attack dictionary current. One major disadvantage of this category is the susceptibility to false positives. Another major disadvantage of this category is the training/profiling phase, during which the system is vulnerable. (This only applies to semisupervised techniques.)

We further classify behavior-based approaches into conventional statistics-based approaches and nonparametric methods. A conventional statistics-based approach may test if a sensor reading or actuator setting is within some number of standard deviations of a mean. Data clustering and support vector machines are examples of nonparametric methods [Cortes and Vapnik 1995]. A feature is a component of a multivariate dataset (e.g., start time, end time, data source, data sink, and position). The size of the feature set is a coarse indicator of efficiency for behavior-based approaches; larger feature sets suggest a larger memory requirement and higher microprocessor use. Feature selection is a key research problem with behavior-based approaches: more features do not necessarily give better results.

3.1.3. Behavior-Specification–Based Intrusion Detection. Behavior-specification–based intrusion detection [Uppuluri and Sekar 2001] is a variant of behavior-based intrusion detection, as shown in the classification tree in Figure 3. We make behavior-specification–based detection a distinctive technique since it has the potential to be the most effective technique for CPS intrusion detection. Behavior-specification–based intrusion detection approaches formally define legitimate behavior and detect an intrusion when the system departs from this model. One major advantage of behavior-specification–based intrusion detection is a low FNR. Only situations that depart from what a human expert previously defined as proper system behavior generate detections. The basic idea is that a bad node will disrupt the formal specification of the system. Another major advantage of behavior-specification–based intrusion detection is that the system is immediately effective because there is no training/profiling phase. The key disadvantage of behavior-specification–based intrusion detection is the effort required to generate a formal specification.

Behavior-specification–based intrusion detection is a form of behavior-based intrusion detection that does not leverage user, group, or data profiling. Instead, humans specify legitimate behaviors, and the IDS measures a node's misbehavior by its deviation from the specification. This allows for lightweight intrusion detection to be deployed in systems with severe resource constraints where user, group, or data profiling is not possible.

3.2. Audit Material

For CPSs, there are two ways to collect data before analysis, namely, host and network-based auditing.

3.2.1. Host-Based Audit. Many IDSs [Park et al. 2010; Mitchell and Chen 2011, 2013d, 2013c; Lauf et al. 2010; He and Blum 2011; Zhang et al. 2011b, 2011a; Asfaw et al. 2010; Carcano et al. 2011; Zimmer et al. 2010; Mitchell and Chen 2012b, 2012a, 2013b, 2013a] that use host-based auditing analyze logs maintained by a node or other audit data, such as file system details, to determine if it is compromised. One major advantage of using host-based auditing is distributed control; this is attractive for high-volume configurations like smart grids. Another major advantage of using host-based auditing is ease of specifying/detecting host-level misbehavior because one can apply well-defined host-specific knowledge to detect intruders. One major disadvantage of host-based auditing is that each node has to perform additional work to collect, if not analyze, its audit data. This is relevant in resource-constrained applications like smart grids. Another major disadvantage of this technique is that a sophisticated attacker can cover its tracks by modifying the audit data on the captured node. A third disadvantage of this technique is that it can be OS or application specific (depending on the particular content of the logs).

3.2.2. Network-Based Audit. Many IDSs [Shin et al. 2010; Tsang and Kwong 2005] that use network-based auditing study network activity to determine if a node is compromised. This audit can be general (e.g., traffic or frequency analysis) or protocol specific (e.g., deep packet inspection). The key advantage regarding resource management is that individual nodes are free of the requirement to maintain or analyze their logs. The key disadvantage regarding data collection is that the visibility of the nodes collecting audit data limits the effectiveness of a network-based technique. That is, it is challenging to arrange network-based audit sensors to get complete intracell and intercell pictures of network activity.

4. CLASSIFICATION OF CPS IDS

The current state of the art in CPS IDS design is preliminary, and not too many CPS IDSs can be found in the literature. We survey 28 CPS IDSs reported in the literature and organize them according to the classification tree in Figure 3. The intent is to examine the most and least intensive research in IDS to date and identify research gaps yet to be explored. We summarize our findings in Tables II and III. Despite our best efforts, these tables do not contain all available work.

To differentiate the 28 CPS IDSs surveyed in Tables II and III, we listed unique CPS aspects that have been considered by each CPS IDS under the CPS Aspects column so that we can compare these 28 CPS IDSs—that is, whether or not these unique CPS aspects have been explored in their CPS IDS design, as well as identify CPS IDS research opportunities/challenges. In Tables II and III, we group existing CPS IDS techniques based on the CPS application (column 2); then, for each CPS application, we group CPS IDS techniques in the format of detection technique/audit material. The Attack Type column gives a description of the attacks for which the CPS IDS is designed. The Audit Features column provides a description of what features a system is working on. The Dataset Quality column indicates the quality of the involved datasets for each CPS IDS surveyed, measured by whether the data used for the experiments are real systems operational data versus simulated data, and whether the data used are made public. In each subsection that follows, we discuss CPS IDS techniques falling into the same class in detail. The performance of each CPS IDS cited is evaluated in terms of p_{fn} and p_{fp} reported. Whenever possible, we quantify the quality of the dataset

Table II. Classification of Aerospace, Automotive, Medical, and SCADA IDSs

Existing Work in CPS IDS Design	CPS Application	Detection Technique	Audit Material	Attack Type	Audit Features	Dataset Quality	CPS Aspects
HybrIDS [Lauf et al. 2010]	aerospace	behavior	host	command injection	Automated Dependent Surveillance Broadcast (ADS-B) and distributed microrobotics protocols [NASA 2005]	unreleased, simulated	LT
UIDS [Mitchell and Chen 2012b]	aerospace	behavior specification	host	command injection, exfiltration	UAV payload, flight control, networking, and IDS state	unreleased, simulated	PPM LT
HPMIDCPS [Mitchell and Chen 2011] [Mitchell and Chen 2013d] [Mitchell and Chen 2013c]	automotive	behavior	host	data manipulation, slander, ballot stuffing	node position, sensor data, and IDS results	unreleased, simulated	AS
Asfaw Technique [Asfaw et al. 2010]	medical	behavior	host	exfiltration	user, time, location, type, network address, and patient for medical record requests	unreleased, operational	
Park Technique [Park et al. 2010]	medical	behavior	host	replay	time and duration of patient locations	public, operational	AS LT
BSID [Mitchell and Chen 2012a]	medical	behavior specification	host	command injection	vital sign monitor (VSM), patient-controlled analgesia (PCA), and cardiac device (CD) status	unreleased, simulated	PPM LT
Killourhy Techniques [Killourhy and Maxion 2010]	SCADA	behavior	host	unauthorized human	key down, key up, and return usage events	public, operational	AS
ACCM/MAS [Tsang and Kwong 2005]	SCADA	behavior	network	KDD Cup 1999	123 features present in the dataset	public, operational	AS

Continued

Table II. Continued

Existing Work in CPS IDS Design	CPS Application	Detection Technique	Audit Material	Attack Type	Audit Features	Dataset Quality	CPS Aspects
Centroid Bro [Düssel et al. 2010]	SCADA	behavior	network	18 CVE threats	n-grams passed over network connections	unreleased, operational	AS
PAYL, POSEIDON, Anagram, and McPAD [Hadžiosmanović et al. 2012]	SCADA	behavior	network	Ingham and Inoue attacks, Microsoft security bulletins, and Digital Bond attacks	n-grams passed over network connections	unreleased, operational	AS LT
Shin Technique [Shin et al. 2010]	SCADA	behavior and knowledge	network	eavesdropping, routing, and DoS	packet arrival rate, source ID, location, routing traffic, message type, and forwarding statistics for components	unreleased, operational	AS
Cheung Technique [Cheung et al. 2007]	SCADA	behavior specification	network	DoS and probing Modbus	Modbus function code and length	unreleased, operational	PPM AS LT

Table III. Classification of Smart Utility IDSs

Existing Work In CPS IDS Design	CPS Application	Detection Technique	Audit Material	Attack Type	Audit Features	Dataset Quality	CPS Aspects
Gao Technique [Gao et al. 2010]	smart utility	behavior	host	MITM, DoS, replay	water level	unreleased, operational	PPM AS LT
CLONALG and AIRS2Parallel [Zhang et al. 2011b, 2011a]	smart utility	behavior	host	NSL-KDD	41 features present in the dataset	public, operational	AS
Bigham Technique [Bigham et al. 2003]	smart utility	behavior	host	power related	power level at different points in the network	unreleased, operational	LT
LOUD, LOED, LOUD-GLR and LOED-GLR [He and Blum 2011]	smart utility	behavior	host	power related	phase, magnitude, voltage, and current at different points in the network	unreleased, simulated	
Bellettini Technique [Bellettini and Rrushi 2008]	smart utility	behavior	host	shellcode, persistent interposition	function and system calls	unreleased, operational	AS LT
IDS-NNM [Linda et al. 2009]	smart utility	behavior	network	zero day	16 features present in the dataset	unreleased, operational	AS LT
Yang Technique [Yang et al. 2005]	smart utility	behavior	network	DoS	62 kernel and I/O-related features present in the dataset	unreleased, operational	
Hadeli Technique [Hadeli et al. 2009]	smart utility	behavior	network	related to Generic Object Oriented Substation Events (GOOSE)	GOOSE metadata	no dataset	PPM CCL LT
Barbosa Technique [Barbosa and Pras 2010]	smart utility	behavior	network	greyhole, command injection	packet source, destination, and timestamp metadata	no dataset	LT
Verba Technique [Verba and Milvich 2008]	smart utility	behavior and knowledge	network	fuzzing, MITM	breaker control and status messages	no dataset	PPM AS LT

Continued

Table III. Continued

Existing Work In CPS IDS Design	CPS Application	Detection Technique	Audit Material	Attack Type	Audit Features	Dataset Quality	CPS Aspects
Oman Technique [Oman and Phillips 2007]	smart utility	knowledge	host	power related	login details, password administration, configuration management, and privilege escalation	unreleased, operational	LT
Premaratne Technique [Premaratne et al. 2010]	smart utility	knowledge	network	ARP spoof, DoS, password crack	traffic analysis and deep packet inspection for ARP, ftp, HTTP, ICMP, and telnet sessions	unreleased, operational	LT
Di Santo Technique [Di Santo et al. 2004]	smart utility	behavior specification	host	power related	power and voltage data at different points in the network	unreleased, operational	AS LT
ISML [Carcano et al. 2011]	smart utility	behavior specification	host	zero day	Modbus fields	unreleased, operational	PPM AS LT
T-Rex [Zimmer et al. 2010]	smart utility	behavior specification	host	shellcode, persistent interposition	microprocessor instruction timestamps	unreleased, operational	PPM CCL AS
SGIDS [Mitchell and Chen 2013b]	smart utility	behavior specification	host	command injection, greyhole	power production and consumption, system configuration and status, and billing rate	unreleased, simulated	PPM LT
Xiao Technique [Xiao et al. 2007]	smart utility	behavior specification	host	water related	the status of six valves, four pumps, and three sensors	no dataset	PPM LT
SCADA IDS [Carcano et al. 2010]	smart utility	behavior specification	network	multipacket Modbus	aggregate PLC and RTU status	unreleased, operational	PPM AS LT

on top of which the evaluation was carried out in order to give to the reader an idea of the reliability of the reported results.

4.1. Behavior/Host

Gao et al. [2010] study an IDS for smart utility (water) applications that uses a three-stage back propagation artificial neural network based on Modbus features. The authors' design performed poorly (42.7% p_{fn}, 45.1% p_{fp}) against replay attacks, but much better against Man-In-The-Middle (MITM) ($0 - 8.9\%$ p_{fn}, $0 - 6.2\%$ p_{fp}) and Denial of Service (DoS) ($0 - 2.0\%$ p_{fn}, $0 - 8.2\%$ p_{fp}) attacks. Gao et al. use an empirical dataset generated by the MSU SCADA testbed. The authors synthesize attacks using six modifications to the water level readings in the dataset: negative water level, water level above HH set point, water level above H set point but below HH set point, water level below L set point but above LL set point, water level below LL set point, and random water level value. The design audits sensor and actuator data (water level readings and valve settings, specifically). The threat model is sophisticated: it considers replay, MITM, and DoS attacks. This investigation considers legacy hardware by dealing with municipal infrastructure whose hardware and software are certified for safety and reliability. This paper addresses two of the unique aspects of CPS.

Zhang et al. [2011b, 2011a] propose CLONALG and AIRS2Parallel for smart utility (power) applications. CLONALG is unsupervised, whereas AIRS2Parallel is semisupervised. The authors reported that CLONALG had a detection accuracy between 80.1% and 99.7% and AIRS2Parallel had an accuracy between 82.1% and 98.7%, where the detection accuracy is the likelihood that the IDS classified a node correctly, calculated by $1 - p_{\text{fp}} - p_{\text{fn}}$. However, [Zhang et al. 2011b, 2011a] gave no ROC data (in terms of a $1 - p_{\text{fn}}$ vs. p_{fp} graph). Immunology inspired them to model immune systems, antigens, lymphocyte cells, and B-cells in their approaches. These studies use an alternate version of the KDD Cup 1999 dataset called NSL-KDD. McHugh [2000] studied the limitations of the KDD Cup 1999 dataset in studied the same topic in [Mahoney and Chan 2003]. The threat model is sophisticated: it comprises DoS, U2R, R2L, and probing attacks. This investigation does not consider legacy hardware. These papers address one of the unique aspects of CPS.

Asfaw et al. [2010] studied a behavior-based IDS for a medical CPS. The authors propose a distributed design where mobile devices collect data that they forward to a centralized audit server. The audit logs comprise location data and medical record access. Their Classification Based on Association (CBA) algorithm is a key artifact and is composed of two parts: the Rule Generator (CBA-RG) and the Classifier Builder (CBA-CB). They did not report false-negative probability p_{fn} or the false-positive probability p_{fp}. Asfaw et al. used an empirical recording of 20 normal records from a single user as their dataset. Since the authors presumed the dataset was free of misbehavior, this explains the lack of false negative results. In addition, this dataset is too small (20 records) and specific (one user) to be useful. The threat model is unsophisticated: the authors only consider exfiltration attacks. This investigation does not consider legacy hardware. This paper does not address any of the unique aspects of CPS.

Bigham et al. [2003] study an IDS for smart utility (power) applications that demonstrates promising control of detection rate and FNR. The authors generated a dataset by calculating total system loads for a six bus network for each hour over 1 year. To synthesize abnormal data, they introduced between 1 and 44 errors into some of the hourly readings. These errors included changing the sign, moving the radix, and changing one of the digits of a reading: this forms an unsophisticated attack model. This investigation considers legacy hardware by dealing with municipal

infrastructure whose hardware and software are certified for safety and reliability. This paper addresses one of the unique aspects of CPS.

The automotive IDS in Mitchell and Chen [2011, 2013d, 2013c] that relies on voting is one example of using behavior detection results in the context of multitrust. One drawback of this study is the lack of simulation to validate the probability model. Mitchell and Chen use Stochastic Petri Net (SPN) modeling techniques [Chen and Wang 1996b, 1996a; Chen et al. 1998; Gu and Chen 2005; Li and Chen 2011] to generate the dataset for their analysis. The threat model is sophisticated: it considers data manipulation, spoofing, slander, ballot stuffing, and node capture attacks by reckless, random, and insidious adversaries. This investigation does not consider legacy hardware. These papers address one of the unique aspects of CPS.

Lauf et al. [2010] propose a behavior-based approach to IDS for aerospace applications called HybrIDS. It comprises two intrusion detection methods: Maxima Detection System (MDS) and Cross-Correlative Detection System (CCDS). Specifically, these two semisupervised approaches combine in three operational phases: MDS training, MDS testing/CCDS training, and CCDS testing. MDS detects single intruders after a short training phase and conducts an in-depth training phase for CCDS. CCDS can detect cooperating intruders after the longer training phase provided by MDS. The authors chose a host-based approach rather than a network-based approach due to the time and memory constraints of an embedded system. HybrIDS is distributed for scalability. They measure the performance of HybrIDS using pervasion, which they define as the percentage of bad nodes in the system. Lauf et al. could detect intruders even with a 22% pervasion; for perspective, the Byzantine fault model establishes a theoretic limit of 33%. During the training/MDS phase, the authors collect data regarding system state. They sequence the nominal system states for use by CCDS so that the probability density function resembles a chi-squared distribution. Lauf et al. use ADS-B or distributed microrobotics protocol [NASA 2005] logs as their audit data. The authors identify two parameters to create an effective IDS for a resource-constrained application: audit collection period (Data Collection Cycle [DCC]) and audit analysis period (Data Processing Cycle [DPC]). A longer DCC increases the memory stress while increasing the detection accuracy of an intrusion detector, and a shorter DPC increases the processor stress while decreasing the detection latency of an intrusion detector. They gave no analysis regarding the trade-off between DCC and DPC. Lauf et al. did not report false-negative probability p_{fn} (i.e., missing a bad node) or the false-positive probability p_{fp} (i.e., misidentifying a good node as a bad node). The authors used a MATLAB script to generate their dataset: the script used a probability density function to produce the normal mission data, and injected emergency action and mission end commands 10% more frequently than normal to produce attack data. This forms an unsophisticated attack model. This investigation considers legacy hardware by dealing with aircraft whose hardware and software are certified for safety and reliability. This paper addresses one of the unique aspects of CPS.

He and Blum [2011] investigated a series of behavior-based IDSs for smart utility (power) applications including Locally Optimum Unknown Direction (LOUD), Locally Optimum Estimated Direction (LOED), LOUD-Generalized Likelihood Ratio (LOUD-GLR), and LOED-Generalized Likelihood Ratio (LOED-GLR). The authors' LOUD-GLR approach performed the best: the maximum detection rate (i.e., $1 - p_{\text{fn}}$) is reportedly 95%. However, He and Blum gave no ROC data. The authors run a Monte Carlo simulation 5,000 times to create a dataset. The authors do not discuss the attack model. This investigation does not consider legacy hardware. This paper does not address any of the unique aspects of CPS.

Park et al. [2010] propose a semisupervised behavior-based IDS targeted for medical CPSs (specifically, assisted living environments). Their design is host based and audits

series of events that they refer to as episodes. The authors' events are three-tuples, comprising sensor ID, start time, and duration. Park et al. test datasets using four similarity functions based on LCS, count of common events not in LCS, event start times, and event durations. They control episode length and similarity function as independent variables. The authors provide excellent ROC data. Park et al. reuse the dataset from an earlier study [Tapia et al. 2004]. They allot 70% of the dataset for presumed-normal training data and 30% for testing data. The authors use random generation and time shifting to seed the testing data with abnormal artifacts. The threat model is sophisticated: it comprises replay attacks. This investigation considers legacy hardware by dealing with medical devices whose hardware and software are certified for safety and reliability. This paper addresses two of the unique aspects of CPS.

Bellettini and Rrushi [2008] study an IDS for smart utility (power) applications that seeds the runtime stack with NULL calls, applies shuffle operations, and performs detection using product machines. The authors carry their study through to implementation on an ARM microprocessor running Linux with a Modbus stack. Bellettini and Rrushi use a semisupervised approach. Although the authors did not report false-negative probability p_{fn} or the false-positive probability p_{fp}, they did report a 6% runtime penalty for the instrumented target. Bellettini and Rrushi create their dataset empirically using an experimental testbed. The threat model is sophisticated: injected shellcode sets up a persistent interposition (rogue library) attack. This investigation considers legacy hardware by dealing with Modbus traffic from ARM-based devices. This paper addresses two of the unique aspects of CPS.

Although Killourhy and Maxion [2010] did not study a specific IDS, they did an exceptionally rigorous analysis of the impact of several parameters on the performance of anomaly detectors that audit keystroke data. These anomaly detectors are not specific to CPS, but they could be used as behavior-based IDSs that use host-based auditing applied to attended CPS nodes. The authors proposed six candidate parameters: detection algorithm, training duration, feature set, updating strategy, impostor practice, and typist-to-typist variation. The detection algorithms they consider are Nearest Neighbor (Mahalanobis), Outlier Count (z-score), and Manhattan (scaled). Impostor practice can be related to a CPS attack scenario where the adversary has surveilled the target and has recordings of legitimate sessions. Typist-to-typist variation can be related to a CPS scenario where the subject has users or processes that diverge from one another to a greater or lesser degree. Killourhy and Maxion used restricted maximum likelihood estimation to determine that detection algorithm, training duration, and updating strategy most strongly influence anomaly detection performance. The authors used the dataset from an earlier study [Killourhy and Maxion 2009] produced by 51 subjects typing a 10- character password 400 times. The attack model is sophisticated: attackers have the opportunity to practice masquerading as a legitimate user. This investigation does not consider legacy hardware. This paper addresses one of the unique aspects of CPS.

4.2. Behavior/Network

Linda et al. [2009] study a semisupervised IDS for smart utility (power) applications called Intrusion Detection System using Neural Network–based Modeling (IDS-NNM). IDS-NNM uses error-back propagation and Levenberg-Marquardt approaches with window-based feature extraction. The most significant of the 16 features that their IDS audited included were IP address count, average interval between packets, number of protocols, flag code count, number of zero window-size packets, zero length packet count, average window size, and average data length. The authors empirically recorded five 20,000 packet datasets between an Allen Bradley PLC 5 and a host workstation.

They artificially generated 100,000 intrusions using Metasploit, Nessus, and Nmap: this forms a sophisticated attack model. This investigation considers legacy hardware. This paper addresses two of the unique aspects of CPS.

Tsang and Kwong [2005] propose a multitrust IDS called Multi-Agent System (MAS) for SCADA applications. Their analysis function, Ant Colony Clustering Model (ACCM), is biologically inspired by its namesake—the ant colony. The authors intend for ACCM to reduce the characteristically high FPR of behavior-based approaches while minimizing the training period by using an unsupervised approach to machine learning. MAS is hierarchical and contains a large number of roles: monitor agents collect audit data, decision agents perform analysis, action agents effect responses, coordination agents manage multitrust communication, user interface agents interact with human operators, and registration agents manage agent appearance and disappearance. Tsang and Kwong's results indicate that ACCM slightly outperforms the detection rates and significantly outperforms the FPRs of k-means and expectation-maximization approaches. One strength of this study is the great false positive result: the ACCM FPR peaks at 6%. The authors use "recall rate" as one of their performance metrics but do not explain its meaning. This study uses the KDD Cup 1999 dataset. The threat model is sophisticated: it considers DoS, U2R, R2L, and probing attacks. This investigation does not consider legacy hardware. This paper addresses one of the unique aspects of CPS.

Düssel et al. [2010] study a semisupervised behavior-based IDS for SCADA applications that uses network-based auditing. This IDS is a centroid-based extension to Bro [Paxson 1999] and achieves a TPR of 90% and an FPR of 0.2%. They use two empirically recorded datasets: one (Web07) contains HTTP traffic from the perimeter network (demilitarized zone or DMZ) of some institution, and the other (Aut09) contains TCP traffic from a SCADA system. One strength of this study is that the authors' attack model is especially strong; it includes 18 entries comprising internal and external threats from the U.S. Government–supported Common Vulnerabilities and Exposures (CVE) database. The authors implement these threats using Metasploit, securityfocus.com, and remote-exploit.org to form a sophisticated attack model. This investigation does not consider legacy hardware. This paper addresses one of the unique aspects of CPS.

Yang et al. [2005] study an IDS for smart utility (power) applications that uses the Simple Network Management Protocol (SNMP) to drive prediction, residual calculation, and detection modules for an experimental testbed. The authors carry their study through to a MATLAB implementation. They use a semisupervised approach. The authors use prior work (Autoassociative Kernel Regression [AAKR] and Sequential Probability Ratio Test [SPRT]) for their analysis function. They claim to provision an FPR of 1% and an FNR of 10% but do not provide numerical data to demonstrate that performance matches parameterization. Of the 62 features that Yang et al.'s dataset includes, the most impactful were processor usage, processor idle time, and 1-minute load average. The authors' dataset consisted of a 1,000 observation training (normal) dataset and a 300 observation test (including intrusions) dataset. Yang et al.'s threat model is unsophisticated: it comprises only ping flood, jolt2, and bubonic DoS attacks. This investigation does not consider legacy hardware, only commodity workstations and servers. This paper does not address any of the unique aspects of CPS.

Hadeli et al. [2009] study a behavior-based IDS for smart utility (power) applications that uses network-based auditing. There is no dataset involved in this study. The design audits sensor and actuator data (water level readings and valve settings, specifically). In addition, it considers closed control loop timing (specifically, the arrival rate of GOOSE messages). The authors do not discuss the attack model. This investigation considers legacy hardware by dealing with ABB System 800xA devices. This paper addresses three of the unique aspects of CPS.

Barbosa and Pras [2010] study an IDS for smart utility (water) applications that tests state machine and Markov chain approaches that use network-based auditing on a water distribution system based on a comprehensive vulnerability assessment. The authors' investigation is incomplete: they do not provide any details on modeling, simulation, or implementation or provide any numerical results. There is no dataset involved in this study. The authors do not discuss the attack model. This investigation considers legacy hardware. This paper addresses one of the unique aspects of CPS.

Hadžiosmanović et al. [2012] compared four behavior-based IDSs that use network-based auditing. One of their datasets deals specifically with Modbus, which is widely used in and unique to CPSs. The four implementations (PAYL, POSEIDON, Anagram, and McPAD) that the authors consider use n-gram analysis. All implementations performed very well for the ICS (Modbus) dataset but struggled to achieve a high detection rate while maintaining a low FPR for the LAN (SMB/CIFS) dataset. They created datasets by fusing presumed normal data recorded from operating contemporary networks with attack data synthesized from signatures provided by public vulnerability databases: this forms a sophisticated attack model. Hadžiosmanović et al. measure false positive and detection rates. This investigation considers legacy hardware by dealing with Modbus devices. This paper addresses two of the unique aspects of CPS.

4.3. Behavior+Knowledge/Network

Shin et al. [2010] present an extension of an existing WSN technique using one hop clustering for SCADA applications; in a one hop cluster, every member falls within radio range of the cluster head. The authors combine one hop clustering for effective intrusion detection (the "second" clustering) with multihop clustering for efficient data aggregation (the "first" clustering) into a hierarchical two-level clustering approach to strike a balance between security and efficiency. This results in a four-layer hierarchy: Member Nodes (MNs) are the leaves, Cluster Heads (CHs) manage MNs, gateways bundle clusters, and a base station is the root of the hierarchy. These different roles analyze audit data the same way, but they respond differently. This heterogeneous approach has the advantage of minimizing the question of trustworthiness; the CHs need to establish trust, whereas the MNs do not. They demonstrate that one hop clustering is particularly effective when detecting spoofing attacks. One strength of this study is the numerical results; the authors report detection rates for jamming, spoofing, hello flooding, data manipulation, greyhole, eavesdropping, routing, and sinkhole attacks: this forms a sophisticated attack model. One drawback of this study is that the results are conflicting; for example, they claim a 25% to 43% detection rate for spoofing attacks in a table summarizing average detection rate and a 60% to 100% detection rate for spoofing attacks in a figure that plots average detection rate as a function of hop counts. The proposed solution is a patchwork of previously established detection techniques, and the authors do not use a single unified dataset to test all techniques working together. This investigation does not consider legacy hardware. This paper addresses one of the unique aspects of CPS.

Verba and Milvich [2008] study an IDS for smart utility (power) applications that takes a multitrust hybrid approach that uses network-based auditing. The authors' attack model includes fuzzing and MITM: this forms a sophisticated attack model. One drawback of this study is a lack of numerical results. There is no dataset involved in this study. The design audits sensor and actuator data and considers legacy hardware. This paper addresses three of the unique aspects of CPS.

4.4. Knowledge/Host

Oman and Phillips [2007] study an IDS for smart utility (power) applications that transforms data collected in Extensible Markup Language (XML) format to Snort

[2012] signatures in an electricity distribution laboratory. One drawback of this study is a lack of numerical results. The authors audit login details (time, source, success, and frequency), password administration, configuration management (software and settings), and privilege escalation. None of these items is uniquely connected to the power grid application. Oman and Phillips create their dataset empirically using an experimental testbed. The authors do not discuss the attack model. This investigation considers legacy hardware. This paper addresses one of the unique aspects of CPS.

4.5. Knowledge/Network

Premaratne et al. [2010] study a knowledge-based IDS for smart utility (power) applications that uses network-based auditing. The authors' design is specific to IEC 61850 infrastructure. Their attack model includes Address Resolution Protocol (ARP) spoofing, DoS, and password cracking: this forms an unsophisticated threat. Premaratne et al.'s design isolates the intrusion detection appliance on a separate host. This is the best practice, although many IDSs are colocated with the resource that they protect. The authors extend Snort to effect their design. They presented a couple of minor results: the timing of cyber attacks does not correspond with time of day; in contrast, traditional kinetic force-on-force attacks are typically launched shortly before dawn. In addition, Premaratne et al. discuss how to position an IDS within a smart utility CPS. The authors created the normal component of their dataset by recording network traffic for two networks for 24 hours each. They recorded traffic for an open-source ARP sniffer running on two hosts for 1 hour to create the abnormal component of their dataset. This investigation considers legacy hardware. This paper addresses one of the unique aspects of CPS.

4.6. Behavior Specification/Host

Di Santo et al. [2004] study a behavior-specification–based IDS for smart utility (power) applications that uses host-based auditing. The authors' main contribution is to propose a parallel algorithm running in a distributed system to effect the intrusion detection. They give little attention to the actual intrusion detection problem. Di Santo et al. create their dataset empirically using an experimental testbed. The attack model is sophisticated: it comprises coordinated disruption of transmission lines. This investigation considers legacy hardware by dealing with municipal infrastructure whose hardware and software are certified for safety and reliability. This paper addresses two of the unique aspects of CPS.

Carcano et al. [2011] propose a behavior-specification–based approach to intrusion detection for smart utility (power) applications called ISML. ISML uses network-based auditing. The design is an extension of Fovino et al. [2010] that distinguishes faults and attacks, describes a language to express a smart grid specification, and establishes a critical state distance metric. The authors base this work on Fovino et al., which guards against complex attacks with a Modbus/DNP3 state machine. Carcano et al. create their dataset empirically over the course of 15 days using an experimental testbed. The design audits sensor and actuator data. The attack model is sophisticated: specifically, it guards against sequences of legitimate SCADA commands that form jellyfish attacks. This investigation considers legacy hardware by dealing with Modbus nodes. This paper addresses three of the unique aspects of CPS.

Zimmer et al. [2010] propose a behavior-specification–based approach to intrusion detection for smart utility (power) applications called T-Rex. T-Rex uses host-based auditing. The design instruments the protected application and uses a scheduler to confirm timing analysis results. Zimmer et al. create their dataset empirically using an experimental testbed. The design audits sensor and actuator data and considers closed control loop timing. The threat model is sophisticated: injected shellcode sets up

a persistent interposition (rogue library) attack. This investigation does not consider legacy hardware; however, the authors experiment on resource constrained hardware (Spectrum Digital DSK6713). This paper addresses three of the unique aspects of CPS.

Mitchell and Chen [2012b, 2012a, 2013b] propose IDSs for aerospace, medical, and smart utility (power) applications, respectively. These are all behavior-specification-based approaches driven by a state machine derived from human-constructed behavior rules. In addition, they consider seven threshold monitoring approaches: binary, Hamming, Manhattan, Euclidean, LCS, Levenshtein, and Damerau-Levenshtein. The authors use a Monte Carlo simulation to create a dataset for good nodes and nodes corrupted by different kinds of attackers. This paper audits sensor and actuator data. The attack model is unsophisticated: the authors only consider reckless and random adversaries prosecuting command injection, greyhole, and exfiltration attacks. This investigation considers legacy hardware by dealing with aircraft, medical, and municipal infrastructure whose hardware and software are certified for safety and reliability. These papers address two of the unique aspects of CPS.

Xiao et al. [2007] study a behavior-specification–based IDS for smart utility (water) applications that uses host-based auditing. This is a system-level IDS that audits the collective state of all system nodes. The authors propose modeling a workflow layer for a subject CPS comprising a simulation manager and a workflow. The workflow collects audit data and performs the intrusion detection while the simulation manager predicts how the attack may propagate. Although they do not fully develop this attack propagation function, it is a great line of investigation given the foothold or island-hopping tactic that contemporary CPS attacks exhibit [Keizer 2010; Stuxnet 2013]. There is no dataset involved in this study. The design audits sensor and actuator data. The authors do not discuss the attack model. This investigation considers legacy hardware. This paper addresses two of the unique aspects of CPS.

4.7. Behavior Specification/Network

Carcano et al. [2010] study a behavior-specification–based IDS for smart utility applications that uses network-based auditing called SCADA IDS. The authors propose a language for describing a specification. Their IDS comprises three modules: a load system that initializes the CPS model using an XML file detailing the configuration of the CPS, a state controller that updates the CPS model based on network traffic (the collection function from Section 2.2), and a rules analyzer that determines if the CPS is in an unsafe state (the analysis function from Section 2.2). Carcano et al. create their dataset empirically using an experimental testbed by sending Modbus commands at 2Hz. The authors create the abnormal component by sending series of 10 commands that try to perform an invalid write to one register and 2 commands that try to perform an invalid write to a bank of coils (single-bit physical outputs). The design audits sensor and actuator data. The attack model is sophisticated: sequences of legitimate SCADA commands form jellyfish attacks. This investigation considers legacy hardware by dealing with Modbus nodes. This paper addresses three of the unique aspects of CPS.

Cheung et al. [2007] study a behavior-specification–based IDS that uses the Prototype Verification System (PVS) to transform protocol, communication pattern, and service availability specifications into a format compatible with EMERALD and Snort. The authors audit the fields of Modbus packets. Specifically, they ensure individual fields are within range (e.g., 0–127 is valid for a one byte field but 128—255 is not) and relationships between fields are preserved (e.g., field 0 is less than field 1). One drawback of this study is a lack of numerical results. Cheung et al. use an empirical dataset generated by the SNL SCADA testbed. The design audits sensor and actuator data. The threat model is sophisticated: a multistage attack penetrates the Internet-facing corporate network, propagates to the DMZ, continues to the Process Control Network

Table IV. Advantages of IDS Techniques for CPSs

Dimension	Type	Pro
Detection technique	Behavior	Detect unknown attacks
	Behavior-Specification	Detect unknown attacks, low false-positive rate
	Knowledge	Low processor demand, low false-positive rate
Audit material	Host	Distributed control and ease of specifying/detecting host-level misbehavior
	Network	Reduced load on resource-constrained nodes

Table V. Drawbacks of IDS Techniques for CPSs

Dimension	Type	Con
Detection technique	Behavior	High false-positive rate
	Behavior-Specification	Human must instrument model
	Knowledge	Attack dictionary must be stored and updated, misses unknown attacks
Audit material	Host	Increased load on resource-constrained nodes, vulnerability of audit material and limited generality
	Network	Effectiveness limited by visibility

(PCN), probes the PCN to learn its topography, and attacks Modbus nodes. This investigation considers legacy hardware by dealing with Modbus nodes. This paper addresses three of the unique aspects of CPS.

5. LESSONS LEARNED

In this section, we discuss lessons learned. We first summarize advantages and drawbacks of existing CPS IDS techniques in each design dimension's options, as evidenced by the most and least studied CPS IDS techniques in the literature. Then we provide insight on the effectiveness of IDS techniques as applying to CPSs and identify research gaps worthy of further research efforts.

5.1. Advantages and Drawbacks of IDS Techniques as Applying to CPSs

Here we discuss the suitability of IDS detection technique/audit material in terms of their advantages and drawbacks when applying to CPSs.

Table IV summarizes the advantages of various detection techniques/audit materials as they apply to CPSs, discussed in more detail as follows:

—The advantage of behavior-based detection techniques is that they detect zero-day attacks. The importance of detecting unknown attacks cannot be overstated. The most sophisticated adversaries will target the most critical systems, and these attackers will not rely on previously disclosed vulnerabilities.
—The advantages of behavior-specification–based detection techniques are that they detect zero-day attacks and yield a low FPR.
—The advantages of knowledge-based detection techniques are that they yield a low FPR and make minimal demands on the host microprocessor.
—The advantages of host-based auditing are distributed control and ease of specifying/detecting host-level misbehavior.
—The advantage of network-based auditing is that it reduces the demand for processor and memory on resource-constrained nodes.

Table V summarizes the drawbacks of various detection techniques/audit materials as they apply to CPSs, discussed in more detail as follows:

Table VI. Most and Least Studied IDS Techniques by Citations (some used more than one detection technique)

CPS Application	Detection Technique	Audit Material	Unique CPS Aspects
Smart utility (18)	Behavior (10) Behavior-Specification (6) Knowledge (3)	Host (11) Network (7)	Physical Process Monitoring (8) Closed Control Loops (2) Attack Sophistication (9) Legacy Technology (14)
SCADA (6)	Behavior (5) Behavior-Specification (1) Knowledge (1)	Network (5) Host (1)	Physical Process Monitoring (1) Closed Control Loops (0) Attack Sophistication (6) Legacy Technology (2)
Medical (3)	Behavior (2) Behavior-Specification (1) Knowledge (0)	Host (3) Network (0)	Physical Process Monitoring (1) Closed Control Loops (0) Attack Sophistication (1) Legacy Technology (2)
Aerospace (2)	Behavior (1) Behavior-Specification (1) Knowledge (0)	Host (2) Network (0)	Physical Process Monitoring (1) Closed Control Loops (0) Attack Sophistication (0) Legacy Technology (2)
Automotive (1)	Behavior (1) Behavior-Specification (0) Knowledge (0)	Host (1) Network (0)	Physical Process Monitoring (0) Closed Control Loops (0) Attack Sophistication (1) Legacy Technology (0)

—The drawback of behavior-based detection techniques is their high FPR. For unattended CPSs operating in hostile or inaccessible locations, unnecessary evictions will reduce lifetime and increase operating cost.
—The drawback of behavior-specification–based detection techniques is that a human must instrument the state machine or grammar that represents safe system behavior. This activity is expensive, slow, and prone to error.
—The drawbacks of knowledge-based detection techniques are that they are helpless against zero-day attacks and rely on an attack dictionary that must be stored and updated. The most sensitive CPSs operate on isolated networks, which obstructs attack dictionary maintenance.
—The drawbacks of host-based auditing are increased processor and memory demand on resource-constrained nodes, vulnerability of audit material, and limited generality based on OS or application.
—The drawback of network-based auditing is that the visibility of nodes collecting audit material limits the effectiveness.

5.2. Most and Least Studied IDS Techniques in the Literature

Table VI summarizes the most and least studied IDS techniques in the literature, grouped by the application type in the order of most to least.

We see that for all applications studied, the most commonly used configurations are behavior-based detection techniques and host-based auditing.

Table VI indicates that there is little research with regard to automotive applications, knowledge-based detection techniques, and network-based auditing.

Some things may not be studied because they are not relevant in the literature. This case could be made for knowledge-based detection techniques because they do not address unknown attacks; assuming that the adversary uses previously seen attacks makes for a weak, unrealistic model. In addition, this case could be made for network-based auditing; the topology has evolved (e.g., from point-to-point or star to mesh),

threatening the tractability of this approach. However, automotive applications are highly relevant as our vehicles become more intelligent (e.g., collision avoidance systems), our mobility patterns evolve (e.g., three-dimensional motion, longer commutes, and urban canyon traversals), and the human capacity to compute while commuting (due to vehicular autonomy) increases.

5.3. Effectiveness of IDS Techniques Applying to CPSs

Based on the advantages and drawbacks of existing CPS IDS techniques discussed in Section 5.1 and the most and least studies in the literature summarized in Section 5.2, in this section we provide insight on the effectiveness of IDS techniques applying to CPSs. We organize our discussion based on the two design dimensions of the classification tree—detection technique and audit material—as follows:

(1) *Detection Techniques*: Knowledge-based designs are not effective for CPSs on their own. They carry a large storage requirement, which legacy hardware or scale may preclude. Knowledge-based designs require frequent dictionary updates in order to protect the resource against the latest threats; unattended deployment or certified configurations disallow this. Even if the attack dictionary is as fresh as possible, these designs are not able to find unknown attacks, thus leaving critical infrastructure vulnerable. However, once signatures are developed, they are the most efficient method to detect attacks for resource-constrained devices common to CPSs, and even a stale attack dictionary can detect some attacks. When used, knowledge-based methods should be paired with a complementary method. Behavior-based designs are more effective than the others for highly redundant CPSs with ample processor margin. Highly redundant CPSs can tolerate wrongful eviction caused by the high FPR of behavior-based IDSs because the reserve nodes offset the aggressive eviction rate. Ample processor margin allows computationally intensive data mining techniques to run without affecting the CPS mission capability. Behavior-specification–based designs are more effective than the others for most CPSs: the channel scarcity does not accommodate dictionary updates associated with knowledge-based designs. Furthermore, storage constraints would limit the size of the attack dictionary. Although both behavior-specification and behavior-based designs can deal with unknown attacks, behavior-specification–based designs have lower FPRs than behavior-based designs in general.
(2) *Audit Material*: Network-based designs are effective for CPSs with wireless segments because these CPSs provide features that are not present in the wireline environment like RSSI and signal-to-noise ratio. For example, an IDS can check that these parameters do not change at all for a stationary node or change in accordance with the motion for a mobile node. Host-based auditing is effective for unattended CPSs: automated operation will result in stable profiles, whereas a human in the loop will yield erratic normal datasets. Although certain CPSs may favor one or the other, both network and host-based designs are important from the perspective of attack detection when there are both network and host centric attacks. The adversary chooses the attack vector; "the enemy has a vote" as warfighters say. Security appliances must organize their defense based on the threat model and not merely based on what is convenient.

5.4. Revisiting IDS Techniques and Gaps in CPS IDS Research

In this section, we identify which research gaps remain and are worthy of further research efforts. We support these findings with the trends observed in Tables II through VI.

Table VI indicates that there are more existing works on IDSs targeted for smart utilities than there are for SCADA, medical, aerospace, and automotive applications combined. However, more than half (11 of 18) of the smart utility–focused research does not provide performance data. Clearly, this focus is an active research area right now, but there is a gap when it comes to numerical results.

There are more works on CPS IDSs that use behavior-based detection (including behavior-specification–based detection) than knowledge-based detection according to Table VI. We attribute this difference to attack sophistication and a high probability of zero-day attacks for CPSs, rendering knowledge-based approaches ineffective and the use of behavior-based approaches a requirement to achieve a sufficient level of security. It remains as a challenge to be able to fully define all possible environment changes and incorporate the laws of physics to define acceptable behavior upon environment changes for behavior-based intrusion. Existing behavior-based approaches may not be the most effective, as there may be missing cases. Behavior-based detection based on specification rules, on the other hand, is emerging and, by means of specification techniques, has the potential to fully specify interactions between a physical component and the CPS environment, governed by the physical processes behind its behavior. However, most investigations have a narrowly or ill-defined attack model. For example, replay attacks seem to challenge behavior-specification–based IDSs. New studies should tie the attack model to a standard repository of vulnerabilities such as the CVE database. The robustness of an attack model could be measured in terms of CVE coverage.

Table VI shows that there has been a similar emphasis on host and network-based auditing to CPS intrusion detection. Each is specific to one of a handful of legacy protocols such as CAN [ISO 11898 2003], DNP3 [DNP3 2010], or Modbus [Modbus Application 2012; Modbus Messaging 2006]. This specificity limits the relevance of these IDSs in terms of time and scope. Table V points out that a weakness of host-based auditing is accommodating actors with erratic profiles. A related open question is how to identify erratic but good nodes and apply an alternate form of IDS to them. Table V also points out that a weakness of network-based auditing is that its effectiveness is limited by the visibility of nodes collecting audit material. Addressing this weakness is an important gap in the literature.

Four aspects of CPSs uniquely impact intrusion detection: physical process monitoring, closed control loops, attack sophistication, and legacy technology. As indicated in Table VI, of the 28 studies we surveyed, 20 considered legacy technology, 17 considered a sophisticated attack model, 11 considered physical process monitoring, and 2 considered closed control loops. None considered all. Clearly, there is a lack of CPS IDS techniques that specifically consider most or all unique aspects of CPSs that differentiate CPSs from ICT systems.

6. FUTURE RESEARCH AREAS

There are many open leads in the area of CPS IDSs.

First and foremost, research is needed to define CPS IDS performance metrics. When numerical results are reported at all, only detection rate, FPR, and FNR are usually given. However, detection latency is a critical metric that researchers rarely report on. A 100% detection rate is a great achievement, but if this IDS takes excessive time to detect intruders, the adversary may still have enough time to damage the target system. We have not found detection latency being studied in the literature, but it is clearly a critical metric. Therefore, researchers should develop detection latency as a key IDS metric.

Second, multitrust [Cho et al. 2011] is unexplored in CPS IDS research. This is the concept of using hearsay/reported information (data from witnesses or third parties).

Liu and Issarny [2004] calls this type of information a *recommendation*. In other cases, the literature calls multitrust approaches *cooperative*. Regardless of the label, multitrust can be distributed or hierarchical [Shin et al. 2010]. This hearsay information can be raw data or an analysis result. Buchegger and Le Boudec [2002] distinguish three levels of multitrust: *experienced* data is a firsthand account that has the most weight, *observed* data happens in the neighborhood (within radio range), and *reported* data is an account coming from outside the neighborhood that has less weight than experienced or observed data. *Hearsay* or *gossip* may also be used to refer to reported data. Contrast these recommendations with what Shin et al. [2010] calls *direct monitoring*. Giving weight to others' recommendations in a federated environment leads to a dilemma: on one hand, a node places enough trust in neighbors to include their hearsay in reputation calculations. On the other hand, nodes are suspicious enough of their environment to measure and respond to the reputation of their neighbors. Therefore, multitrust is better suited to increasing the security of managed/authenticated environments rather than to establishing a basic level of security. The key problem is guaranteeing that the larger dataset yields a net gain in key metrics despite the presence of bad-mouthing and ballot-stuffing attacks [Chen et al. 2010, 2013; Bao et al. 2011, 2012; Cho et al. 2009, 2012]. Multitrust deserves more attention because it expands the dataset available to an IDS.

Third, there is little network-based CPS IDS research in the literature based on our survey result listed in Table VI. However, it deserves attention because CPSs will have predictable mission-essential traffic profiles that their IDSs should leverage. In addition, CPS IDSs should be extremely frugal in the artifacts that they study to avoid increasing processor demand.

Fourth, audits should focus on application-layer data. The audit of lower layer data that is common to any application has been well studied, so adversaries expect these defensive measures. A cunning adversary will craft an attack to appear normal in every way possible to avoid widely deployed IDSs. IDSs that audit application-layer data focus on detecting the adversary where it must reveal itself to attack the system.

Fifth, model-based analysis techniques [Cho et al. 2010; Al-Hamadi and Chen 2013; Mitchell and Chen 2011, 2013d, 2013c] need to be developed and validated to analyze performance of CPS IDS protocols and identify optimal CPS IDS protocol settings to maximize CPS IDS performance based on performance metrics defined. Configuration items (e.g., number of intrusion detectors, audit interval, and detection threshold) impact the detection and FPRs of the IDS and longevity of the CPS as a whole. Researchers should identify parameters that have a local maximum and parameters that are covariant. They should establish heuristics for finding the optimal value for the former set and equations that characterize the trade-off for the latter set. To this end, closed form math models are the best tool. In their absence, investigators should establish analytical models. Furthermore, they should instrument simulations to validate the analytical models.

Sixth, not all adversaries behave the same, so researchers should identify attacker models. Key characteristics include behavior (considering timeline, degree of collusion, and sophistication) and capture rate. The literature is thin on adversary modeling [Mitchell and Chen 2013c]. There is a need for modeling and analysis of adversary behavior and intrusion detection defenses for CPSs.

Seventh, behavior-specification–based detection deserves more research attention. Knowledge-based detection techniques may not be viable for many CPS applications because they cannot detect unknown attacks. Behavior-based detection techniques may not be viable because of their high FPRs. However, more effort is needed to further refine the threshold monitoring technique coupled with behavior-specification–based

detection. In particular, existing works [Mitchell and Chen 2012b, 2012a] use a binary failure threshold to classify a node as malicious or normal—that is, based on if the node's current compliance degree is lower or higher than a threshold. Other failure threshold criteria based on fuzzy failure criteria [Bastani et al. 1994; Chen and Bastani 1991; Chen et al. 1995] may prove to be more effective against environmental noise and/or smart attackers. Identifying environment variables, defining environment changes in terms of environment variable changes, and incorporating the laws of physics to define acceptable behavior upon environment changes are critical milestones in this line of investigation.

Eighth, researchers should pursue responses tailored to attacker behavior. The best intrusion response is situational: if the adversary is persistent, eviction is the priority. If the adversary is transient, repairing the system is the priority. If the adversary is ineffective, establishing attribution for the attack is the priority. Investigators should study proactive responses: a CPS that completely depends on reactive measures can fall victim to attacks by an adversary that continually provokes a mission-affecting intrusion response. In the ICT world, often active approaches (aiming at actively blocking the malicious traffic) are preferred to passive ones (logging alerts into a big file for analysis). Finding effective response approaches for CPSs is an important challenge to address.

Ninth, there is little CPS IDS research in the literature that considers the closed loop control blocks of a CPS. These are key artifacts of CPSs; their real-time requirements challenge IDSs to avoid disruption while they afford IDSs opportunities in the form of highly predictable behavior profiles.

Tenth, researchers should study federated CPS IDSs. Actors from different enclaves will have different missions and therefore different behavior profiles. IDSs from different enclaves will struggle to establish trust so they can share audit data and effect trans-enclave sanctions. Multitrust can be a key design factor in building future federated CPS IDSs.

Finally, there is little CPS IDS research in the literature that considers automotive applications. However, automotive applications are highly relevant as our vehicles become more intelligent (e.g., collision avoidance systems); our mobility patterns evolve to include three-dimensional motion, longer commutes, and urban canyon traversals; and the human capacity to compute while traveling (due to vehicular autonomy) increases.

REFERENCES

Hamid Al-Hamadi and Ing-Ray Chen. 2013. Redundancy management of multipath routing for intrusion tolerance in heterogeneous wireless sensor networks. *IEEE Transactions on Network and Service Management* 19, 2, 189–203.

K. M. Ali, W. Venus, and M. S. Al Rababaa. 2009. The affect of fuzzification on neural networks intrusion detection system. In *Proceedings of the 4th Conference on Industrial Electronics and Applications*. Xi'an, China, 1236–1241.

B. Asfaw, D. Bekele, B. Eshete, A. Villafiorita, and K. Weldemariam. 2010. Host-based anomaly detection for pervasive medical systems. In *Proceedings of the 5th International Conference on Risks and Security of Internet and Systems*. Montreal, QC, Canada, 1–8.

Fenye Bao, Ing-Ray Chen, MoonJeong Chang, and Jin-Hee Cho. 2011. Trust-based intrusion detection in wireless sensor networks. In *Proceedings of the International Conference on Communications*. Kyoto, Japan, 1–6.

Fenye Bao, Ing-Ray Chen, MoonJeong Chang, and Jin-Hee Cho. 2012. Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection. *IEEE Transactions on Network and Service Management* 9, 2 (June 2012), 169–183.

Rafael Barbosa and Aiko Pras. 2010. Intrusion detection in SCADA networks. In *Mechanisms for Autonomous Management of Networks and Services*, Burkhard Stiller and Filip De Turck (Eds.). Lecture Notes in Computer Science, Vol. 6155. 163–166.

Farokh B. Bastani, Ing-Ray Chen, and Tai-Wei Tsao. 1994. Reliability of systems with fuzzy-failure criterion. In *Proceedings of the Annual Reliability and Maintainability Symposium*. Anaheim, California, USA, 442–448.

Carlo Bellettini and Julian Rrushi. 2008. A product machine model for anomaly detection of interposition attacks on cyber-physical systems. In *Proceedings of the 23rd International Federation for Information Processing International Information Security Conference*. Milan, Italy, 285–300.

John Bigham, David Gamez, and Ning Lu. 2003. Safeguarding SCADA systems with anomaly detection. In *Computer Network Security*, Vladimir Gorodetsky, Leonard Popyack, and Victor Skormin (Eds.). Lecture Notes in Computer Science, Vol. 2776. 171–182.

Sonja Buchegger and Jean-Yves Le Boudec. 2002. Performance analysis of the CONFIDANT protocol. In *Proceedings of the 3rd International Symposium on Mobile Ad Hoc Networking & Computing*. Lausanne, Switzerland, 226–236.

Andrea Carcano, Alessio Coletta, Michele Guglielmi, Marcelo Masera, Igor Nai Fovino, and Alberto Trombetta. 2011. A multidimensional critical state analysis for detecting intrusions in SCADA systems. *IEEE Transactions on Industrial Informatics* 7, 2 (May 2011), 179–186.

Andrea Carcano, Igor Nai Fovino, Marcelo Masera, and Alberto Trombetta. 2010. State-based network intrusion detection systems for SCADA protocols: A proof of concept. In *Critical Information Infrastructures Security*, Erich Rome and Robin Bloomfield (Eds.). Lecture Notes in Computer Science, Vol. 6027. 138–150.

Oliver Chapelle, Bernhard Schölkopf, and Alexander Zien. 2006. *Semi-Supervised Learning*. Vol. 2. MIT Press, Cambridge, MA.

Ing-Ray Chen, Fenye Bao, MoonJeong Chang, and Jin-Hee Cho. 2010. Trust management for encounter-based routing in delay tolerant networks. In *Proceedings of the Global Communications Conference*. Miami, FL, USA, 1–6.

Ing-Ray Chen, Fenye Bao, MoonJeong Chang, and Jin-Hee Cho. 2013. Dynamic trust management for delay tolerant networks and its application to secure routing. *IEEE Transactions on Parallel and Distributed Systems*.

Ing-Ray Chen and Farokh B. Bastani. 1991. Effect of artificial-intelligence planning-procedures on system reliability. *IEEE Transactions on Reliability* 40, 3, 364–369.

Ing-Ray Chen, Farokh B. Bastani, and Tai-Wei Tsao. 1995. On the reliability of AI planning software in real-time applications. *IEEE Transactions on Knowledge and Data Engineering* 7, 1, 4–13.

Ing-Ray Chen, Tsong-Min Chen, and Chiang Lee. 1998. Performance evaluation of forwarding strategies for location management in mobile networks. *Computer Journal* 41, 4, 243–253.

Ing-Ray Chen, Anh Speer, and Mohamed Eltoweissy. 2011. Adaptive fault tolerant QoS control algorithms for maximizing system lifetime of query-based wireless sensor networks. *IEEE Transactions on Dependable and Secure Computing* 8, 2, 161–176.

Ing-Ray Chen and Ding-Chau Wang. 1996a. Analysis of replicated data with repair dependency. *Computer Journal* 39, 9, 767–779.

Ing-Ray Chen and Ding-Chau Wang. 1996b. Analyzing dynamic voting using petri nets. In *Proceedings of the 15th IEEE Symposium on Reliable Distributed Systems*. Niagara Falls, Canada, 44–53.

Steven Cheung, Bruno Dutertre, Martin Fong, Ulf Lindqvist, Kieth Skinner, and Alfonso Valdes. 2007. Using model-based intrusion detection for SCADA networks. In *Proceedings of the SCADA Security Scientific Symposium*. Miami, FL, USA, 127–134.

Jin-Hee Cho, Ing-Ray Chen, and Phu-Gui Feng. 2010. Effect of intrusion detection on reliability of mission-oriented mobile group systems in mobile ad hoc networks. *IEEE Transactions on Reliability* 59, 1, 231–241.

Jin-Hee Cho, Ananthram Swami, and Ing-Ray Chen. 2009. Modeling and analysis of trust management for cognitive mission-driven group communication systems in mobile ad hoc networks. In *Proceediings of the International Conference on Computational Science and Engineering*. 641–650.

Jin-Hee Cho, Ananthram Swami, and Ing-Ray Chen. 2011. A survey on trust management for mobile ad hoc networks. *IEEE Communications Surveys and Tutorials* 13, 4, 562–583.

Jin-Hee Cho, Ananthram Swami, and Ing-Ray Chen. 2012. Modeling and analysis of trust management with trust chain optimization in mobile ad hoc networks. *Journal of Network and Computer Applications* 35, 3, 1001–1012.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2001. *Introduction to Algorithms*. MIT Press.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning* 20, 3, 273–297.

Hervé Debar, Marc Dacier, and Andreas Wespi. 2000. A revised taxonomy for intrusion-detection systems. *Annales Des Tele communications* 55, 7–8, 361–378.

Michele Di Santo, Alfredo Vaccaro, Domenico Villacci, and Eugenio Zimeo. 2004. A distributed architecture for online power systems security analysis. *IEEE Transactions on Industrial Electronics* 51, 6 (December 2004), 1238–1248.

DNP3. 2010. IEEE Standard for Electric Power Systems Communications Distributed Network Protocol (DNP3). *IEEE Std 1815-2010* (January 2010), 1–775.

Patrick Düssel, Christian Gehl, Pavel Laskov, Jens-Uwe Bußer, Christof Störmann, and Jan Kästner. 2010. Cyber-critical infrastructure protection using real-time payload-based anomaly detection. In *Critical Information Infrastructures Security*, Erich Rome and Robin Bloomfield (Eds.). Lecture Notes in Computer Science, Vol. 6027. 85–97.

Dewan M. Farid and Mohammad Z. Rahman. 2008. Learning intrusion detection based on adaptive Bayesian algorithm. In *Proceedings of the 11th International Conference on Computer and Information Technology*. Khulna, Bangladesh, 652–656.

Bingrui Foo, Yu-Sung Wu, Yu-Chun Mao, Saurabh Bagchi, and Eugene Spafford. 2005. ADEPTS: Adaptive intrusion response using attack graphs in an e-commerce environment. In *Proceedings of the International Conference on Dependable Systems and Networks*. Yokohama, Japan, 508–517.

Igor Nai Fovino, Andrea Carcano, T. De Lacheze Murel, Alberto Trombetta, and Marcelo Masera. 2010. Modbus/DNP3 state-based intrusion detection system. In *Proceedings of the 24th International Conference on Advanced Information Networking and Applications*. Perth, Australia, 729–736.

Wei Gao, Thomas Morris, Bradley Reaves, and Drew Richey. 2010. On SCADA control system command and response injection and intrusion detection. In *Proceedings of the 5th Annual Anti-Phishing Working Group eCrime Researchers Summit (eCrime)*. Dallas, TX, USA, 1–9.

Yunlu Gong, S. Mabu, Ci Chen, Yifei Wang, and K. Hirasawa. 2009. Intrusion detection system combining misuse detection and anomaly detection using Genetic Network Programming. In *Proceedings of the International Conference on Control, Automation and Systems–The Society of Instrument and Control Engineers*. Fukuoka, Japan, 3463–3467.

Baoshan Gu and Ing-Ray Chen. 2005. Performance analysis of location-aware mobile service proxies for reducing network cost in personal communication systems. *ACM Mobile Networks and Applications* 10, 4, 453–463.

Fariba Haddadi and Mehdi A. Sarram. 2010. Wireless intrusion detection system using a lightweight agent. In *Proceedings of the 2nd International Conference on Computer and Network Technology*. Bangkok, Thailand, 84–87.

Hadeli Hadeli, Ragnar Schierholz, Markus Braendle, and Cristian Tuduce. 2009. Leveraging determinism in industrial control systems for advanced anomaly detection and reliable security configuration. In *Proceedings of the Conference on Emerging Technologies Factory Automation*. Palma de Mallorca, Spain, 1–8.

Dina Hadžiosmanović, Lorenzo Simionato, Damiano Bolzoni, Emmanuele Zambon, and Sandro Etalle. 2012. N-Gram against the machine: On the feasibility of the N-Gram network analysis for binary protocols. In *Research in Attacks, Intrusions, and Defenses*, Davide Balzarotti, Salvatore J. Stolfo, and Marco Cova (Eds.). Lecture Notes in Computer Science, Vol. 7462. 354–373.

Hong Han, Xin-Liang Lu, and Li-Yong Ren. 2002. Using data mining to discover signatures in network-based intrusion detection. In *Proceedings of the International Conference on Machine Learning and Cybernetics*, Vol. 1. Beijing, China, 13–17.

Qian He and Rick S. Blum. 2011. Smart grid monitoring for intrusion and fault detection with new locally optimum testing procedures. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*. Prague, Czech Republic, 3852–3855.

Geoffrey Hinton and Terrence J. Sejnowski. 1999. *Unsupervised Learning: Foundations of Neural Computation*. MIT Press.

ISO 11898. 2003. Road Vehicles—Interchange of Digital Information—Controller Area Network (CAN) for High Speed Communication.

Gregg Keizer. 2010. Is Stuxnet the Best Malware Ever? http://www.computerworld.com/s/article/9185919/Is_Stuxnet_the_best_malware_ever_

Kevin S. Killourhy and Roy A. Maxion. 2009. Comparing anomaly-detection algorithms for keystroke dynamics. In *Proceedings of the International Federation for Information Processing International Conference on Dependable Systems Networks*. Lisbon, Portugal, 125–134.

Kevin Killourhy and Roy Maxion. 2010. Why did my detector do that?! In *Recent Advances in Intrusion Detection*, Somesh Jha, Robin Sommer, and Christian Kreibich (Eds.). Lecture Notes in Computer Science, Vol. 6307. 256–276.

Adrian P. Lauf, Richard A. Peters, and William H. Robinson. 2010. A distributed intrusion detection system for resource-constrained devices in ad-hoc networks. *Ad Hoc Networks* 8, 3, 253–266.

Yinan Li and Ing-Ray Chen. 2011. Design and performance analysis of mobility management schemes based on pointer forwarding for wireless mesh networks. *IEEE Transactions on Mobile Computing* 10, 3, 349–361.

Ondrej Linda, Todd Vollmer, and Milos Manic. 2009. Neural network based intrusion detection system for critical infrastructures. In *Proceedings of the International Joint Conference on Neural Networks*. Atlanta, GA, USA, 1827–1834.

Jinshan Liu and Valerie Issarny. 2004. Enhanced reputation mechanism for mobile ad hoc networks. *Trust Management.* Lecture Notes in Computer Science, Vol. 2995. 48–62.

Yang-Xia Luo. 2010. The research of Bayesian classifier algorithms in intrusion detection system. In *Proceedings of the International Conference on E-Business and E-Government*. Guangzhou, China, 2174–2178.

Matthew V. Mahoney and Philip K. Chan. 2003. An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection. In *Recent Advances in Intrusion Detection*, Giovanni Vigna, Christopher Kruegel, and Erland Jonsson (Eds.). Lecture Notes in Computer Science, Vol. 2820. 220–237.

John McHugh. 2000. Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. *ACM Transactions on Information and System Security* 3, 4 (November 2000), 262–294.

Sudip Misra, P. Venkata Krishna, and Kiran Isaac Abraham. 2010. Energy efficient learning solution for intrusion detection in wireless sensor networks. In *Proceedings of the 2nd International Conference on Communication Systems and Networks*. Bangalore, India, 1–6.

Robert Mitchell and Ing-Ray Chen. 2011. A hierarchical performance model for intrusion detection in cyber-physical systems. In *Proceedings of the IEEE Wireless Communication and Networking Conference*. 2095–2100.

Robert Mitchell and Ing-Ray Chen. 2012a. Behavior rule based intrusion detection for supporting secure medical cyber physical systems. In *Proceedings of the IEEE International Conference on Computer Communication Networks*. Munich, Germany.

Robert Mitchell and Ing-Ray Chen. 2012b. Specification based intrusion detection for unmanned aircraft systems. In *Proceedings of the ACM MobiHoc Workshop on Airborne Networks and Communications*. Hilton Head Island, SC, USA, 31–36.

Robert Mitchell and Ing-Ray Chen. 2013a. Adaptive intrusion detection for unmanned aircraft systems based on behavior rule specification. *IEEE Transactions on Systems, Man and Cybernetics.*

Robert Mitchell and Ing-Ray Chen. 2013b. Behavior rule based intrusion detection systems for safety critical smart grid applications. *IEEE Transactions on Smart Grid* 4, 3, 1254–1263.

Robert Mitchell and Ing-Ray Chen. 2013c. Effect of intrusion detection and response on reliability of cyber physical systems. *IEEE Transactions on Reliability* 62, 1, 199–210.

Robert Mitchell and Ing-Ray Chen. 2013d. On survivability of mobile cyber physical systems with intrusion detection. *Wireless Personal Communications* 68, 4, 1377–1391.

Modbus Application. 2012. MODBUS Application Protocol Specification. http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf.

Modbus Messaging. 2006. MODBUS Messaging on TCP/IP Implementation Guide. http://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b. pdf.

NASA. 2005. Swarming for success. *Astrobiology Magazine.*

National Science Foundation. 2011. Cyber-Physical Systems (CPS) Program Solicitation.

Lin Ni and Hong-Ying Zheng. 2007. An unsupervised intrusion detection method combined clustering with chaos simulated annealing. In *Proceedings of the International Conference on Machine Learning and Cybernetics*, Vol. 6. Hong Kong, China, 3217–3222.

Paul Oman and Matthew Phillips. 2007. Intrusion detection and event monitoring in SCADA networks. In *Critical Infrastructure Protection*, Eric Goetz and Sujeet Shenoi (Eds.). International Federation for Information Processing, Vol. 253. 161–173.

Kyungseo Park, Yong Lin, Vangelis Metsis, Zhengyi Le, and Fillia Makedon. 2010. Abnormal human behavioral pattern detection in assisted living environments. In *Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments*. 9:1–9:8.

Vern Paxson. 1999. Bro: A system for detecting network intruders in real-time. *Computer Networks* 31, 2324, 2435–2463. http://www.sciencedirect.com/science/article/pii/S1389128699001127.

Upeka K. Premaratne, Jagath Samarabandu, Tarlochan S. Sidhu, Robert Beresh, and Jian-Cheng Tan. 2010. An intrusion detection system for IEC61850 automated substations. *IEEE Transactions on Power Delivery* 25, 4 (October 2010), 2376–2383.

Rockwell Automation Technologies, Inc. 2009. Introduction to Historian System Management. http://samplecode.rockwellautomation.com/idc/groups/literature/documents/gr/hsepis-gr021_-en-e.pdf.

Sooyeon Shin, Taekyoung Kwon, Gil-Yong Jo, Youngman Park, and H. Rhy. 2010. An experimental study of hierarchical intrusion detection for wireless industrial sensor networks. *IEEE Transactions on Industrial Informatics* 6, 4 (November 2010), 744–757.

Snort. 2012. Snort. http://www.snort.org.

Robin Sommer and Vern Paxson. 2010. Outside the closed world: On using machine learning for network intrusion detection. In *Proceedings of the IEEE Symposium on Security and Privacy*. Oakland, CA, USA, 305–316.

Maria Striki, Kyriakos Manousakis, Darrell Kindred, Dan Sterne, Geoff Lawler, Natalie Ivanic, and George Tran. 2009. Quantifying resiliency and detection latency of intrusion detection structures. In *Proceedings of the Military Communications Conference*. Boston, MA, USA, 1–8.

Stuxnet. 2013. Stuxnet. http://en.wikipedia.org/wiki/Stuxnet.

Emmanuel Munguia Tapia, Stephen S. Intille, and Kent Larson. 2004. Activity recognition in the home using simple and ubiquitous sensors. In *Pervasive Computing*, Alois Ferscha and Friedemann Mattern (Eds.). Lecture Notes in Computer Science, Vol. 3001. 158–175.

Chi-Ho Tsang and Sam Kwong. 2005. Multi-agent intrusion detection system in industrial network using ant colony clustering approach and unsupervised feature extraction. In *Proceedings of the International Conference on Industrial Technology*. Hong Kong, China, 51–56.

Prem Uppuluri and R. Sekar. 2001. Experiences with Specification-Based Intrusion Detection. In *Recent Advances in Intrusion Detection*, Wenke Lee, Ludovic M, and Andreas Wespi (Eds.). Lecture Notes in Computer Science, Vol. 2212. 172–189.

Jared Verba and M. Milvich. 2008. Idaho National Laboratory Supervisory Control and Data Acquisition Intrusion Detection System (SCADA IDS). In *Proceedings of the Conference on Technologies for Homeland Security*. Idaho Falls, ID, USA, 469–473.

Gregory B. White, Eric A. Fisch, and Udo W. Pooch. 1996. Cooperating security managers: A peer-based intrusion detection system. *IEEE Network* 10, 1 (January/February 1996), 20–23.

Michael E. Whitman and Herbert J. Mattord. 2011. *Principles of Information Security*. Course Technology Ptr.

Kun Xiao, Nianen Chen, Shangping Ren, Limin Shen, Xianhe Sun, K. Kwiat, and M. Macalik. 2007. A workflow-based non-intrusive approach for enhancing the survivability of critical infrastructures in cyber environment. In *Proceedings of the 3rd International Workshop on Software Engineering for Secure Systems*. Minneapolis, MN, USA.

Dayu Yang, Alexander Usynin, and J. Wesley Hines. 2005. Anomaly-based intrusion detection for SCADA systems. In *Proceedings of the 5th International Topical Meeting on Nuclear Plant Instrumentation, Control and Human Machine Interface Technologies*. Albuquerque, NM, USA, 12–16.

Lin Ying, Zhang Yan, and Ou Yang-jia. 2010. The design and implementation of host-based intrusion detection system. In *Proceedings of the 3rd International Symposium on Intelligent Information Technology and Security Informatics*. Jinggangshan, China, 595–598.

Yichi Zhang, Lingfeng Wang, Weiqing Sun, R. C. Green, and M. Alam. 2011a. Artificial immune system based intrusion detection in a distributed hierarchical network architecture of smart grid. In *Proceedings of the Power and Energy Society General Meeting*. Detroit, MI, USA, 1–8.

Yichi Zhang, Lingfeng Wang, Weiqing Sun, R. C. Green, and M. Alam. 2011b. Distributed intrusion detection system in a multi-layer network architecture of smart grids. *IEEE Transactions on Smart Grid* 2, 4 (December 2011), 796–808.

Shi Zhong, Taghi M. Khoshgoftaar, and Shyarn V. Nath. 2005. A clustering approach to wireless network intrusion detection. In *Proceedings of the 17th International Conference on Tools with Artificial Intelligence*. Hong Kong, China, 196–202.

Christopher Zimmer, Balasubramanya Bhat, Frank Mueller, and Sibin Mohan. 2010. Time-based intrusion detection in cyber-physical systems. In *Proceedings of the 1st International Conference on Cyber-Physical Systems*. Stockholm, Sweden, 109–118.

Received February 2013; revised September 2013; accepted November 2013