Towards Scalable, Secure, and Smart Mission-Critical IoT Systems: Review and Vision (Special Session Paper)

Xiaolong Guo
Kansas State University
guoxiaolong@ksu.edu

Song Han
University of Connecticut
song.han@uconn.edu

X. Sharon Hu
University of Notre Dame
shu@nd.edu

Xun Jiao
Villanova University
xun.jiao@villanova.edu

Yier Jin
University of Florida
yier.jin@ece.ufl.edu

Fanxin Kong
Syracuse University
fkong03@syr.edu

Michael Lemmon
University of Notre Dame
lemmon@nd.edu

ABSTRACT

Recent emerging technologies such as artificial intelligence and machine learning have been promising enormous economic and societal benefits. While it is desirable to deploy these technologies to Internet-of-Things (IoT) infrastructures in many applications such as medical, energy, transportation, and industrial automation systems, such deployments present daunting challenges in performance, efficiency, and dependability of scaling-up IoT infrastructure, due to the ever-increasing number of edge devices, ever-increasing levels of device and system heterogeneity, and more stringent requirements of reliability, robustness, and security in mission-critical settings. This position paper elaborates the needs for a cross-layer and full hardware/software stack solution for the *design and deployment of scalable, secure, and smart mission-critical IoT systems* from four different perspectives and research fields. We present a review of recent studies on such issues and identify the potential challenges and gaps, based on which we highlight some important research directions and future works that can be conducted to tackle such challenges.

CCS CONCEPTS

• Computing methodologies → Embedded System; **• Hardware → Electronic design automation**; **Robustness**; **• Security and privacy**; **• Computer systems organization → Embedded and cyber-physical systems**; **Real-time systems**;

ACM Reference Format:
Xiaolong Guo, Song Han, X. Sharon Hu, Xun Jiao, Yier Jin, Fanxin Kong, and Michael Lemmon. 2021. Towards Scalable, Secure, and Smart Mission-Critical IoT Systems: Review and Vision (Special Session Paper) . In *2021 International Conference on Embedded Software Companion (EMSOFT'21 Companion), October 8–15, 2021, Virtual Event, USA*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3477244.3477624

1 INTRODUCTION

The growing capabilities of sensing, computing and communication devices are leading to an explosion of Internet of Things (IoT). Somewhat orthogonally, disruptive technologies such as artificial intelligence have been promising enormous economic and societal benefits. While it is naturally desirable to deploy these technologies in IoT infrastructures, such deployments present daunting challenges for increasingly scaling-up IoT infrastructures in mission-critical applications such as medical, energy, transportation, and industrial automation systems. These challenges pose immediate threat to the performance, efficiency, and dependability of scaling-up IoT infrastructure.

The challenges stem from several major aspects in terms of scalability. First, the number of edge devices can be enormous, e.g., in the order of billions [17], which makes a centralized management infeasible. Second, there are multiple layers of heterogeneity [73]. An IoT system can consist of heterogeneous computing subsystems; each subsystem can have heterogeneous computing devices; and each single device can be composed of different kinds of computing components. Third, mission-critical applications have stringent requirements in correctness, resilience, timeliness, security, and safety [80]. It is difficult for a large-scale IoT system to satisfy these requirements due to the increasing adversarial surfaces.

An IoT infrastructure is typically a layered structure composed of data centers, gateways/aggregators, and edge devices (Fig. 1), which exhibit the following features. First, the target system's performance deeply impacts business/organization operations on, e.g., transportation and industrial automation industries. Second, the considered IoT platform consists of data centers, gateways/aggregators, and edge devices, which are naturally heterogeneous in many aspects including computational capacity, network latency, and hardware and software. Third, each single device can contain heterogeneous computing components which are good at processing different types of workloads.

EMSOFT'21 Companion, October 8–15, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8712-5/21/10…$15.00
https://doi.org/10.1145/3477244.3477624

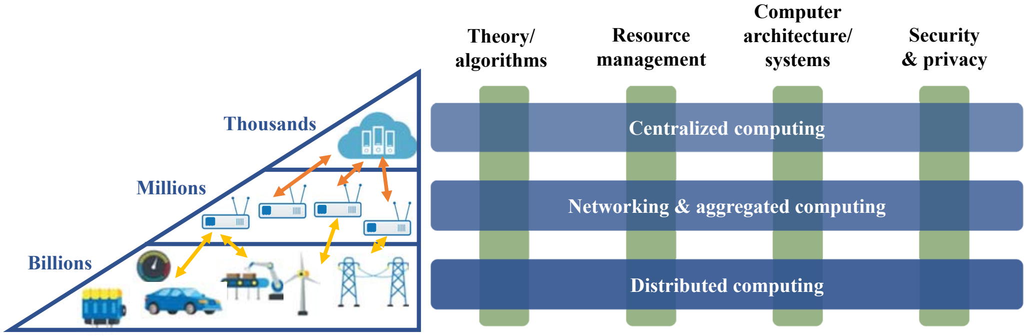

Figure 1: Overview of a cross-layer and full hardware/software stack solution for mission-critical IoT system design.

Hence, to guarantee the performance of such IoT systems, we argue that a cross-layer and full hardware/software stack solution is needed. In this paper, we approach such challenges and problems from four perspectives: (i) theory/algorithm (data-driven modeling); (ii) resource management (network resource management); (iii) architectures/systems and (iv) security/privacy (design-time vulnerability detection, and runtime detection and recovery of physical attacks). For each perspective, we highlight key challenges, review representative recent work, and then provide our vision and insights. We conclude the paper with a discussion on the lessons learned.

2 DATA-DRIVEN MODELING AND CONTROL OF QOS IN SAFETY-CRITICAL IOT SYSTEMS

Safety-critical IoT fabrics are complex dynamical systems for which it is impossible to identify accurate *a priori* models. This section describes novel data-driven theory/algorithms that learn dynamical models of the IoT fabric and outlines how these models can be used in a hierarchical manner to manage IoT resources that control the fabric's quality-of-service (QoS).

2.1 Challenges

Feedback control theory has long been considered for managing complex computer systems forming the IoT fabric because this theory provides tools allowing one to manage a dynamical system's behavior in an application independent manner. The effective use of this theory, however, has been hampered by the lack of accurate process models. Internet congestion control [49] was one of the earliest uses of control theory in network congestion control. But moving from theory to real-life [36] required creative ways of estimating network parameters. Control theory has been suggested for managing software systems [62] and the quality of service (QoS) of IoT systems [22, 69]. This prior work relies on simplistic models with few results showing the robustness of these methods with respect to the modeling uncertainty seen in real-life. So while feedback control provides an attractive theory for managing IoT system QoS, the identification of system models remains an obstacle to the successful use of the theory in real life.

2.2 Existing Work

System complexity, scale, and openness all conspire to make model identification challenging in IoT applications [79]. IoT systems are *complex* and *large-scale* networks with many interacting nodes using ad hoc protocols to provide best-effort delivery. These IoT systems are *open* to human users generating time-varying workloads and an exogenous environment that perturbs network parameters and topology. The large-scale and open nature of these systems injects a great deal of modeling uncertainty which couples with dynamic complexity to make it extremely difficult to obtain accurate models required for control system design. IoT modeling must therefore address these challenges of scale, complexity, and openness before feedback control methods become practical tools for managing real-life IoT applications.

Machine learning paradigms such as deep reinforcement learning (DRL) [55] cannot fully address these issues. DRL uses a deep neural network to realize the actor in reinforcement learning's actor-critic schema. DRL-based control has been used to manage IoT computing/radio resources [91] in support of smart city services [56]. While DRL based control can avoid scaling issues, it seems poorly suited for open systems. Safety critical IoT systems (traffic control or industrial automation) experience transient disruptions that must be addressed in real time. DRL has difficulty handling such transient disruptions because its training is so time consuming. It has been suggested [89] that incrementally augmenting the actor neural network can improve transient response. While the strategy has shown promise in smart city traffic control [89], the black-box nature of the neural network training algorithms makes it difficult to see how well these methods would generalize to other IoT applications.

2.3 Vision

Our vision sees an approach for managing safety-critical IoT that builds on recent advances in hierarchical control and data-driven learning. In particular, we propose a method for identification and control based on a novel synthesis of moment matching reduced order models (MM-RoM) [3] and Koopman decompositions [6] used in an hierarchical control framework [24] where the physical system "approximately simulates" [25] the system's model. Our preliminary work suggests this data-driven approach provides a scalable way of identifying and controlling complex dynamical systems that are open to the outside environment.

We treat the IoT fabric as an input/output system, denoted as Σ_c. This system has two types of input signals; the *workload*, w_k, and the *control*, u_k. The workload, w_k, is an aggregate measure of the total work submitted to the system at time k and is treated as an exogenous signal. The control input, u_k, is the *desired* QoS that the fabric has been commanded to enforce. The output, y_k, of Σ_c is the actual QoS delivered by the fabric at time k. Because of the time-varying nature of the workload, the actual QoS, y_k, will deviate from the desired QoS, u_k. This variation is dynamic in the sense that y_k at time k depends on all prior inputs. The fabric, in other words, has a "memory" which is usually represented concretely as a *state vector*, $\mathbf{x}_k$. For control purposes we need to identify a state-based model for Σ_c that captures the dynamic relationship between inputs, (w_k, u_k), states, $\mathbf{x}_k$, and outputs, y_k.

Our proposed approach uses the fact that the system's output, y_k, can be decomposed as

$$\text{total response, } y_k = \text{natural response} + \text{forced response} \qquad (1)$$

The natural response is that part of y_k generated solely by the system's internal state, $\mathbf{x}_k$. The forced response is the *steady-state* part of y_k generated in response to the input w_k alone. Rather than identifying a single monolithic model or control algorithm (as is done in DRL), we identify separate models for the natural and forced response that can then be smoothly integrated into a hierarchical control framework.

The forced response model is a moment-matching reduced order model (MM-RoM) [3] denoted as Σ_a whose output, ψ_k, is the MM-RoM's prediction of the physical system's quality of service. This model, Σ_a, assumes that the workload is also generated by a dynamical system, Σ_g, called the workload generator. This assumption allows one to identify a family of MM-RoM whose dynamics are determined by Σ_g and where the nonlinear map from these internal model states to output ψ_k is all that needs to be learned. Identification of the MM-RoM therefore reduces to a standard linear regression problem [70] that can be solved using recursive algorithms that balance prior and posterior information in an optimal manner. The system's, Σ_c, historic off-line outputs and workload are used to generate the *a priori* RoM, Σ_a, which is updated on-line so the model can adapt to changes in the physical system.

A model for the natural response is learned from the *model-following system*, $\Sigma_a - \Sigma_c$, mapping control input, u_k, onto the model following error, $e_k = \psi_k - y_k$. This is another stage of the model learning process which occurs after a posterior MM-RoM has been learned. For this second learning problem, we use a *delay-embedding* [39] of these errors to form a state vector, $\mathbf{z}_k$, that evolves linearly through the system's *Koopman operator* [6]. This *linear* model of the natural dynamics can also be efficiently identified (learned) in a data driven manner using the dynamic mode decomposition (DMD) algorithm [44]. So not only do we have an efficient way of identifying the natural dynamics, we also have a model that is "linear" in terms of the delay embedded, state, $\mathbf{z}_k$.

Both of these models fit naturally within Girard's *hierarchical control architecture* [24] shown in Fig. 2. This architecture has the MM-RoM, Σ_a, generate an output, ψ_k, that feeds into a *control interface* that drives the physical system Σ_c, so it *approximately simulates* the MM-RoM [25]. *Approximate simulation* uses the term "simulation" in the sense defined by Milner [51]. The approximate nature of the simulation relation means that the model following error remains bounded for bounded perturbations of the workload generator's, Σ_g, output. Because the natural dynamics are "linear" through the Koopman operator, passivity based control [85] can be readily used to design an approximately simulating control interface that is robust to passive model uncertainties. Maintaining approximate simulation also means that one may use the MM-RoM for planning and management with an assurance that the actual system will follow this plan.

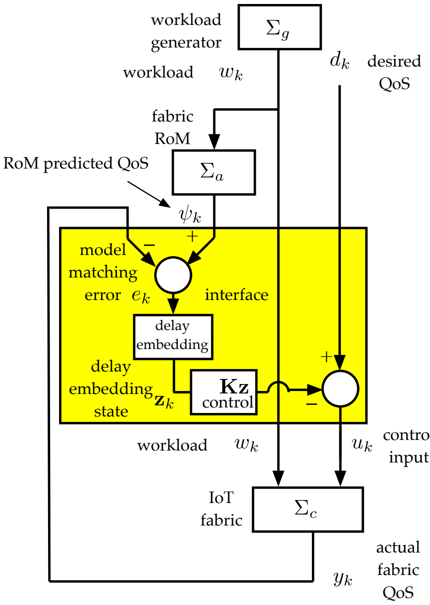

Figure 2: Hierarchical Control Architecture for Data-Driven Control of the IoT Fabric QoS

The control system shown in Fig. 2 is the building block used to construct a hierarchy of controllers. The top layer of this hierarchy consists of controllers managing attached routers. The second layer of the hierarchy has routers control the edge nodes. The approximate simulating nature of the control architecture ensures that high-level commands are faithfully executed by lower level nodes even in the face of bounded disturbances to expected workload, system topology and parameters.

Our prior work [43] and more recent unpublished results support the assertions made above regarding the benefits of this approach. In particular, our recent work has demonstrated scalable and robust model following of a complex hopping robot [64] based solely on data from detailed robot simulations. While these mechanical systems are not large-scale IoT applications, the proposed framework and design methods are independent of the application and so these methods should also provide a way to manage an IoT network's average QoS. Current work is assessing the extent to which this data-driven approach can be used to dynamically model errors in voltage-scaled ASICs [35] and end-to-end delay in 6TiSCH networks [86].

3 DYNAMIC RESOURCE MANAGEMENT FOR REAL-TIME WIRELESS IOT NETWORKS

Real-time wireless networks (RTWNs) are a critical resource in mission-critical IoT fabrics as they form the backbone that connect edge devices with gateways and the cloud. In this section, we discuss challenges, related work and our vision on network resource management for mission-critical IoT fabrics.

3.1 Challenges

In recent years, we have witnessed the rapid development and deployment of real-time wireless technologies in various industrial sectors, including but not limited to smart transportation and advanced industry automation [75, 82]. Compared with their wired counterparts, RTWNs are featured with easier deployment, reduced maintenance cost and enhanced device mobility. This paradigm shift makes RTWNs the foundation of many existing and emerging mission-critical IoT systems.

However, RTWNs face several unique challenges in their resource management. First of all, it is critical but challenging to meet the stringent timing requirements of sensing and control tasks running on RTWNs in mission-critical IoT applications. Traditional medium access mechanisms (e.g., CSMA/CA) may cause unexpected packet loss and undetermined transmission latency. In contrast, RTWNs typically adopt time division multiple access (TDMA) based mechanisms to achieve deterministic end-to-end message delivery. Packet scheduling in RTWNs thus plays a central role in achieving the desired performance but becomes challenging when the network scales up.

The second challenge lies in the fact that almost all RTWNs need to deal with unexpected *disturbances* since they are typically deployed in complex and mission-critical environments. Unexpected disturbances in general can be classified into external disturbances of the target physical systems (e.g., sudden pressure change in an oil pipeline) and internal disturbances within the network fabric (e.g., link failure due to multi-user interference). To assure stable and safe operations in the presence of external disturbances, mission-critical tasks in the target system may increase their demands to the network resources (e.g., requesting higher sampling rates). On the other hand, internal disturbances may lead to permanent or transient faults in the network which can also reduce the network's capacity. Therefore, disturbances will not only impact the RTWN's demand but also its supply of the network resources.

To handle unexpected disturbances without pessimistic resource reservation, dynamic and distributed resource management solutions need to be designed. However, finding the right level of dynamic decision making is not trivial as it is a tradeoff between efficient use of network resources (e.g., no wasted bandwidth) and achievable Quality of Service (QoS) (e.g., the number of messages missing end-to-end timing constraints).

3.2 Existing Work

Many RTWNs perform resource management via static data link layer scheduling [31, 45, 68, 78, 90] to periodically gather the network health status, and then recompute and distribute the updated network schedule information. This process is slow, not scalable and incurs considerable network overhead, and thus is not suitable for handling unexpected disturbances and can lead to less responsive systems, which is not acceptable for mission-critical IoT applications. In response to various disturbances, centralized link layer scheduling approaches [10, 11, 13, 71, 72] have been proposed. However, those protocols either are not able to respond to external disturbances [13, 72], or assume that only a predetermined number of link layer schedules are stored in the system [71].

There are a few work on adapting to unexpected external disturbances in control systems. For example, rate-adaptive and rhythmic task models are introduced in [7] and [41], respectively. These models allow the system to adapt to external disturbances by changing the periods and relative deadlines of the tasks in the run time. They, however, cannot be straightforwardly applied to RTWNs because their schedulability analyses do not consider the situation of end-to-end packet delivery in mission-critical IoT systems.

To overcome the weakness of the prior work, we have developed a suite of dynamic resource management techniques in RTWNs to provide guaranteed QoS in the presence of unexpected disturbances. The first work along this line is a hybrid dynamic packet scheduling framework, referred to as D^2-PaS, to handle external disturbances which cause abruptly increased network traffic [97, 99]. Different from traditional centralized resource management methods under which a centralized control node undertakes all the work to handle external disturbances, D^2-PaS offloads the computation from the centralized controller node to local nodes and only executes a lightweight algorithm in the controller node to determine the corresponding response to the external disturbance. In this way, better QoS can be achieved. To handle the internal disturbances, we further introduce a reliable dynamic packet scheduling framework, called RD-PaS [26]. RD-PaS can not only react to on-line network traffic changes caused by external disturbances in a dynamic fashion, but also construct reliable schedules to deal with packet loss caused by internal disturbances. Both D^2-PaS and RD-PaS rely on a centralized controller node to make on-line decisions. Such centralized approaches will cause scalability issue when the network scales up. To address this issue, in our most recent work, a fully distributed packet scheduling framework, referred to as FD-PaS, is introduced [98, 100]. FD-PaS incorporates several key advances in both algorithm design and data link layer protocol design to enable individual nodes to make on-line decisions locally and achieve guaranteed response time to unexpected disturbances.

3.3 Vision

We envision that to effectively manage large-scale and heterogeneous RTWNs for mission-critical IoT applications, spatial channel reuse must be exploited and hierarchical resource management approaches should be considered.

Exploiting spatial channel reuse. The scalability of FD-PaS is limited by a drawback that it only supports single-channel networks without spatial channel reuse. This leads to severe resource under utilization. The limitation of FD-PaS are due to the fact that each device determines its local dynamic schedule *independently* only based on the locally stored information (*e.g.*, static schedule and interference table). Thus a device's decision is not known by other devices on the path of the mission-critical tasks and may result in inconsistent usage of transmissions slots by different devices.

To address the inconsistency issue, we propose a *successive and distributed* packet scheduling framework, SD-PaS. SD-PaS lets each device along the transmission path of the mission-critical task take turns to determine the dynamic schedule for their transmissions and propagate the decision to the subsequent devices on the path. To ensure SD-PaS always lead to consistent schedules with high utilization, two technical challenges need to be tackled. For the first challenge—how to avoid transmission interference and schedule inconsistency when constructing dynamic schedules locally based only on local interference information, we shall determine (i) the essential information needed to generate local schedules, and (ii) the essential information to be propagated at particular time along the path of the mission-critical task. For the second challenge—how to construct the dynamic local schedules that will collaboratively result in the minimum number of dropped packets due to the system overload caused by unexpected disturbances, one can consider to formulate an optimization problem for dynamic local schedule construction, analyze its complexity and propose both exact and efficient approximation algorithms to meet the user requirements.

Hierarchical resource partitioning. Dynamic, distributed network resource management techniques such as D^2-PaS, RD-PaS, FD-PaS and even SD-PaS only focus on a single RTWN. However, in reality, for large-scale IoT fabrics, it is not uncommon that multiple independent and heterogeneous RTWNs are deployed in the same geographical area and share the same spectra. Without knowing the exact communication schedules or the existence of one another, these co-existing RTWNs may create severe interference to each other and unavoidably degrade the QoS of all the networks.

We propose a hierarchical resource partitioning framework, HARP, to manage the network resources among co-existing heterogeneous RTWNs. As shown in Fig. 3, HARP assumes the deployment of a high-level manager to collect abstract resource demands (with no detailed schedules) from individual RTWNs. HARP then allocates resource partitions (in the form of channel-time blocks) accordingly to each RTWN. Based on the allocated resource partition, the network manager of each RTWN constructs its own communication schedule without interfering the operation of any other networks. When disturbances occur, the affected networks can update their resource demands to the high-level resource manager which in turn adapts the current partition and disseminates the updated partition to all affected networks. To realize the concept of HARP, we envision the following three key building blocks: (i) a resource interface to capture the essential resource demands from RTWNs; (ii) novel algorithms to construct resource partitions for individual RTWNs; and (iii) partition adaptation semantics and adaptation protocols to ensure timely update of partition information in the presence of disturbances. HARP can be further integrated with the hierarchical control architecture presented in Section 2 to determine the required update on the resource demands from individual RTWNs based on the observed QoS at runtime.

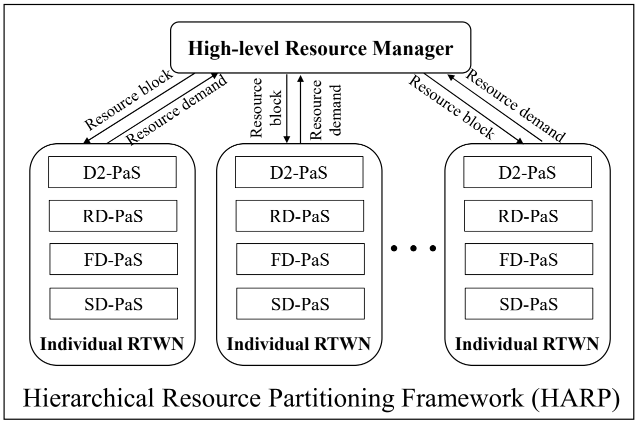

Figure 3: Overview of the HARP framework.

4 VULNERABILITY DETECTION FOR IOT SYSTEM THROUGH FORMAL METHOD

Two features of IoT system, heterogeneous and connectivity, lead to the security threats of mission-critical IoT system. We discuss the security issues into two categories – static and runtime. In this section we focus on the static protection of the IoT system, while the runtime protection is presented in the Section 5.

4.1 Challenges

The rapid growth of IoT industry put the threat of computer systems, considering the collaboration of hardware and software together, front and center among all security concerns. Heterogeneous applications in scalable IoT systems and platforms significantly expand the attack surface and increase the vulnerability. In the meantime, a large number of computing components are deployed and accessible in the physical layer, introducing the security uncertainty in the connectivity.

In detecting vulnerabilities of hardware system, formal methods have been proved effective among all existing techniques [15, 16, 30, 33, 37, 38, 48, 65, 101]. However, very few of the current formal verification approaches are scalable and practical in industry due to the lack of automatic and efficient tools. Therefore, a framework composed by the formal verification is needed to efficiently detect vulnerabilities that may reside deeply in binaries deployed in heterogeneous IoT devices.

4.2 Existing Work

To overcome these security challenges from hardware perspective, numerous of research studies have been investigated. We discuss the works of information flow tracking (IFT) and Satisfiability Modulo Theories (SMT) solver as follows.

Information Flow Tracking (IFT) We represent heterogeneous applications as data-flow model and perform security checking using IFT. IFT is a powerful approach to efficiently protect confidentiality in a hardware system by detecting the sneaky path of sensitive information leakage. IFT associates data or operations with labels/taint indicating the security levels. Existing static IFT solutions at RT-level require manual work in either annotating codes or proving properties. These solutions enforce the noninterference policy, which eliminates the dependency between lower sensitive outputs and higher sensitive inputs. Given that the inner structure of the system is treated as a black box, confidentiality of a system is presented as that an adversary learns no more information from system executions than from direct observations. That is, from the observable outputs, attackers cannot deduce anything about the secret inputs [67]. Various secure RTL programming languages, such as Caisson [47], Sapper [46], SecVerilog [95], etc. are developed to check the noninterference property.

However, when applying these IFT solutions, users must learn the complex tag system and manually denote labels standing for the trust level to specify the information flow policy. QIF-Verilog is then proposed to protect the confidentiality as an alternative simplified and automatic solution [29]. QIF-Verilog only extends one security label from the standard Verilog to reduce the cost of learning from the developers' side. It relaxes noninterference by quantifying how much information is being leaked [76]. In QIF-Verilog, an information leakage metric, called accumulated RU, is generated through calculating the entropy to quantify the leakage of labelled secrets in the hardware design. However, QIF-Verilog can only protect confidentiality and validate IFT properties in IP level.

Satisfiability Modulo Theories (SMT) Solver Satisfiability (SAT) solvers have been used in many electronic design automation fields like logic synthesis, verification, and testing. The SAT solvers are originally designed to solve the well-known Boolean Satisfiability problem, which decides whether a propositional logic formula can be satisfied given value assignments of the variables in the formula. Based on SAT solver, satisfiability modulo theories (SMT) solver is derived by including several first-order theories, such as arithmetic, bit-vectors, quantifiers, etc [14]. However, due to the high computational complexity, there is no hardware implementation for SMT solvers, and the software based SMT solver are not scalable to large designs.

Symbolic execution is a program analysis technique that can explore multiple paths that a program could take under different inputs [5]. In this method, execution paths that the program should take are explored systematically to avoid the space explosion problem. Specifically, inputs are represented as symbols and the solvers are used to check whether there are counter examples of the property. For each path, a Boolean formula is derived to describe the conditions of the branches, while a symbolic memory is used to map variables to symbolic expressions. The Boolean formula is updated after executing the branch and the symbolic memory is updated after each assignment. Integrating these two techniques overcome the NP-Hard computation complexity issue in SAT solver and it provides a comprehensive protection by automatically checking the customized properties.

4.3 Vision

We propose a framework for detecting vulnerabilities from the binaries to the whole IoT system statically. Checking in the SMT solver, formal verification is utilized to protect the interactions among different layers in the mission-critical IoT framework. The communication and control protocols established between distributed computing devices and network gateways as well as between gateways and centralized servers are analyzed before the deployment. At the system level, the information confidentiality is protected against sneaky paths using IFT. Two levels of the privacy protection are delivered – 1) privacy will not be leaked to a user who should not be a receiver inside the system, and 2) privacy will not be leaked outside the system.

In this framework, formal verification is utilized to protect the interactions among different layers. The runtime overhead is eliminated By validating the communication channels in the mission-critical IoT framework statically. Specifically, communication and control protocols established 1) between distributed computing devices and network gateways and 2) between gateways and centralized servers will be analyzed before the deployment.

The SMT solver is adopted as the platform, which determines the satisfiability of propositional complex formulas in theories such as arithmetic, bit-vectors, quantifiers, etc. Given a proposition of above formulas, the SMT solver decides whether the proposition can result in a true conclusion through assigning appropriate value to its variables. On the other hand, the SMT solver is able to efficiently derive the satisfiability without compromising completeness or full automation. In the past decade, SMT solver has become the essential checking engine in a broad range of technologies and has been widely used in formal verification, program analysis, testing, and program synthesis.

The security properties are pre-defined depending on the security requirements which are various for different systems. A library is also generated including general used security properties. By selecting appropriate combinations from the library, users are capable to customize their own security policies. Meanwhile, the developed verification method includes an extraction of the behavior model from the communication protocols. A intermediate representations (IR) format is developed to represent equivalent expressions derived from different protocols. The security properties will also be presented using the same IR. The automated checking will be performed to detect the mismatch between the model and the properties. In order to protect the information confidentiality from sneaky paths in the mission-critical IoT system, we first convert the whole system or protocols' IR to a data-flow graph (DFG), then perform IFT on the generated DFG. This static checking process is performed before the system deployment, hence, no runtime overhead is introduced.

To enforce the confidentiality, two levels H (high sensitive, aka Private) and L (low sensitive, aka Public) are designed and then H labeled signals are restricted flowing to L labeled signals. Note that the reverse operation is allowed. The information tracking policy is designed based on the sensitive levels which label the signals transmitting in the system. All the distributed devices are treated as nodes in the DFG. When considering the whole IoT, more dedicated DFG model needs to be built. For instance, all DFGs are connected and cover the entire IoT system. The sensitive labels inside this DFG are initialized and propagated to check privacy confidentiality within the IoT environment. Besides the IFT-based techniques, we also envision an integration of formal methods and fuzz testing techniques based on concrete execution [9, 50, 87]. The integrated solution may help identify more vulnerabilities statically.

5 REAL-TIME ADAPTIVE SENSOR ATTACK DETECTION FOR IOT SYSTEMS

Many mission-critical IoT systems tightly interact with the physical system via sensors and actuators. One crucial security risk in such systems is sensor attacks. Acting on malicious sensing information may drive the system to perform dangerous actions and cause serious consequences. In this section, we discuss challenges, related work and our vision on defending against sensor attacks for mission-critical IoT security. While Section 4 focuses on the cyber part and offline design phase, this section complements it and targets the physical aspect and runtime phase.

5.1 Challenges

We consider sensor attacks in IoT, where an attacker modifies sensing information to negatively affect and even causes safety issues to the physical system. Consider an example attack that alters measurements of a speed sensor of an autonomous vehicle to smaller values. Then, the attack can misguide the controller to speed up a vehicle, and the actual speed may be greater than the desired speed [32, 42, 96]. This may eventually result in an accident. Also, changing temperature readings to smaller values can cause a power plant overheated and even an explosion [18].

To defend against sensor attacks is challenging because of the wide attack surface and runtime aspects including detection accuracy and timing constraints.

First, the sensor attack surface is wide. Sensor attacks can be launched by compromising software or the communication between sensors and controllers. Besides these convectional cyber attack surfaces, sensors can be also corrupted by transduction attacks, which manipulate the physical property to affect sensor measurements [1, 2, 12, 34, 60, 63, 92, 102]. For example, an attacker can inject fake GPS signals to misguide a yacht [66], compromise wheel speed sensors to corrupt antilock braking systems [74], or affect gyroscope readings by sound noises [77].

Second, the detection accuracy is important. We need to not only consider false negatives that may cause safety validation but also false alarms that decide the usability of an detection method. That is, an undetected attack may drift off the system to the unsafe region; and more false alarms will reduce the usability of a detector.

Third, timing of attack detection is also important. Untimely detection, i.e., finding an attack after consequences happen, is just as damaging. Consider the same example of the speed sensor attack as above. The attack needs to be detected before the vehicle crashes; otherwise, the detection result is useless even if it is accurate.

5.2 Existing Work

The threats and challenges mentioned above have motivated extensive studies on sensor attack detection. The following will discuss these studies from two orthogonal perspectives: system behavior prediction and statistics tracking.

System Behavior Prediction. Existing studies can be divided into several threads according to how they predict system behaviors. First, some works rely on sensor redundancy and conduct cross-checking information of the redundant sensors for the detection [19, 61, 93, 94]. Second, there are some signature-based works that compare run-time patterns with a pre-defined dictionary. The dictionary includes attack types or attack patterns already known [21, 40]. Third, some works use behavioral rule-based detection, where the detector raises an alarm if a system does not follow some specifications on state transitions or execution constraints defined beforehand [4, 53, 54].

Fourth, a major thread of works studies how physical invariants can be used to detect sensor attacks, where a physical invariant follows certain physical laws. The basic idea here is first to extract a physical invariant of a system beforehand and then compare the observed sensor readings with the values predicted by the invariant at runtime. If the difference between the two is greater than a pre-defined threshold, an alert will be raised. There are two kinds of physical invariants that are widely-used in the literature. The first kind captures the dynamics of a physical system using a system model described by differential or difference equations or learnt models [12, 23, 27, 28, 52, 63]. The second kind of invariant refers to sensor correlation. The correlation captures the fact that multiple sensors react to the same physical phenomenon in a correlated way [2, 20, 32, 59, 81]. For example, when braking a vehicle, the vehicle speed and engine speed will both decrease and GPS readings will be also affected accordingly.

Statistics Tracking. There are mainly two different methods to track the statistics of difference between the predicted and the observed behavior: stateless and stateful. For stateless detection works, they raise an alarm for a considerable difference (called residual) between a predicted value and an observed value by the sensor at every single time point [32, 57, 58, 88]. For stateful detection works, they use a statistic that keeps tracking of historical residuals, e.g., average or cumulative sum. Then, they generate an alarm if a persistent deviation across multiple time points is confirmed [8, 12, 63, 83, 84]. In general, stateful detection works come with fewer false alarms (or higher usability) but longer detection delay, while stateless detection works in the opposite way since less data points are used for detection.

5.3 Vision

In spite of the large volume of existing works, the timing and usability of sensor attack detection have not been well addressed. The timing here refers to the detection delay, which is defined as the time interval between the start of an attack and the detection of it. The usability is tied to the false alarm rate, and a higher (lower) rate means a worse (better) usability. Some works focus on improving either the detection delay or false alarm rate. Other works attempt to minimize the two metrics at the same time. However, this attempt is deemed to hardly succeed because of the inherent trade-off between the two metrics. That is, a lower detection delay usually comes with more false alarms; and vice versa [2, 23, 83, 84]. Further, when a system is closer to the unsafe region, there is less time remaining for the detection and thus reducing the detection delay will have higher priority than reducing false alarms; and vice versa. Hence, we envision that attack detection should prefer different metrics when a system runs in different states.

We propose real-time adaptive sensor attack detection for mission-critical IoT systems. First, we need a real-time detector to address the timing challenge. As noted, untimely defense may also cause serious consequences. Second, we need an adaptive detector that can adjust its detection delay and false alarms. The real-time adaptive detector will discover sensor attacks before the detection deadline while improving its usability. To achieve this goal, we argue that a detector needs to have the capabilities of computing the detection deadline and adapting its detection delay. The following discusses how to enable such capabilities for attack detection.

Estimating Detection Deadline. Existing detection works usually do not consider the detection deadline as a design factor. Instead, they just report the resulted delay in their evaluations and then claim fast detection [1, 2]. Further, they do not address how fast is fast enough to avoid unsafe situations. We propose to use the detection deadline to bound the detection delay. However, it is challenging to calculate the detection deadline.

A multiple-step approach is proposed. First, we define the detection deadline as as the time interval between the current time and a future time point when a system may reach the unsafe region. Because sensor measurements can be arbitrarily modified by an attacker, the generated control inputs are unpredictable. Therefore, the detection deadline needs to be estimated in a conservative way in order to cover the arbitrary sensor modification.

Second, the detection deadline changes as the system state varies. If a system is already close to the unsafe region, then it will take less time to touch the region (if it keeps moving towards the direction), that is, a short or stringent detection deadline. Therefore, to obtain the detection deadline offline is infeasible, and instead, it needs to carry out estimation at run time. Third, the online estimation needs to have low computational overhead; otherwise, after the computation completes, the resulted deadline may be already outdated in a time-constraint environment.

Adjusting Detection Delay. We then need to adapt the detection delay of the detector according to the detection deadline. We believe that adjusting the delay appropriately can not only guarantee timely detection but also improve the usability of the detector. However, little attention has been paid to how to adapt the detection delay [1, 2]. To enable such adaptability is not easy, and an attack detector needs to have at least two features as follows.

First, the detection delay of the detector needs to be predictable in the first place in order to be adjustable. This means that the delay can bounded by a maximum value when identifying an attack, i.e., the detected attack starts no earlier than a time point. Note that this is meaningful only if the attack is detected; otherwise, when the attack occurs is unknown as the detector fails to find it. Further, cumulative sum (CUSUM) based detection approaches tracks a trimmed sum of all historical residuals and thus do not have predictable detection delay by nature [2, 63]. Second, to align with the varying detection deadline, the detector needs to dynamically change its detection delay at run time. Thus, stateless detection or detection with fixed delay is inapplicable here.

The vision discussed above focuses on bridging the gap on the timing aspects for attack detection. Thus, it is orthogonal to existing works that are confined to the detection accuracy. We believe that combining our vision with those existing works will yield both timely and accurate detection.

6 LESSON LEARNED

To deploy a scalable, secure, and smart mission IoT systems, a multi-faceted research approach is imperatively needed from different domains. This position paper envisions such a problem from four research areas: theory/algorithms, resource management, security, and computer systems.

From the theory/algorithm perspective, we envision data-driven distributed hierarchical control algorithms [24] based on moment-matching [3] and approximate simulation [25] concepts that can *regulate* application workload's utilization of IoT real-time services when the IoT infrastructure is overloaded. This is essential to achieve scalable control of IoT systems through the distributed nature of the hierarchical controls and smart control via a novel mixture of supervised and unsupervised learning schemes that identify moment-matching reduced order abstractions of the IoT infrastructure's behavior.

From the resource management perspective, we envisioned a scalable *runtime* resource management framework based on *decentralized* techniques for handling workload uncertainties caused by changes in either the application environment or the IoT infrastructure itself. The considered resources encompass computing and communication resources with considerations in real-time, reliability, and energy consumption requirements. The framework can be employed in tandem with the distributed hierarchical control algorithms.

From the computer system and security perspectives, We envisioned a hybrid mechanism composed of both design-time and runtime protection approaches for secure deployment of IoT systems. Design-time techniques (e.g., formal verification) focuses on discovering vulnerabilities in cyber systems. However, this does not protect the IoT system from attacks originated from physical attackers; for example, sensor readings can be manipulated. Runtime detection and recovery, especially under real-time constraints, must be developed. The unique cyber-physical characteristic of IoT systems calls for such hybrid protection mechanism.

7 CONCLUSION

The ever-increasing scale of the IoT infrastructure in mission-critical applications poses emerging and serious threats to the functionality and dependability of such applications. This perspective paper elaborates the challenges in deploying a scalable, secure, and smart mission-critical IoT system by reviewing existing literature and identifying potential gaps. We stress that a cross-layer multi-faceted approach is the key to address such challenges, and discuss potential solutions including data-driven modeling, real-time resource management, formal methods-based vulnerability detection, and adaptive sensor attack detection and recovery. We hope that the paper provides possible directions for future research endeavors in mission-critical IoT system deployment.

8 ACKNOWLEDGEMENTS

The work is supported in part by the National Science Foundation through the PPoSS Planning Project (NSF-2028897, NSF-2028910, NSF-2028875, NSF-2028879, NSF-2028889, and NSF-2028740).

REFERENCES

[1] Francis Akowuah and Fanxin Kong. Physical invariant based attack detection for autonomous vehicles: Survey, vision, and challenges. In *4th International Conference on Connected and Autonomous Driving (MetroCAD)*. IEEE, 2021.

[2] Francis Akowuah and Fanxin Kong. Real-time adaptive sensor attack detection in autonomous cyber-physical systems. In *27th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 2021.

[3] Alessandro Astolfi. Model reduction by moment matching for linear and non-linear systems. *IEEE Transactions on Automatic Control*, 55(10):2321–2336, 2010.

[4] Stanley Bak, Karthik Manamcheri, Sayan Mitra, and Marco Caccamo. Sandboxing controllers for cyber-physical systems. In *2011 IEEE/ACM Second International Conference on Cyber-Physical Systems*, pages 3–12. IEEE, 2011.

[5] Roberto Baldoni, Emilio Coppa, Daniele Cono D'Elia, Camil Demetrescu, and Irene Finocchi. A survey of symbolic execution techniques. *arXiv preprint arXiv:1610.00502*, 2016.

[6] S.L. Brunton, B.W. Brunton, J.L. Proctor, and J.N. Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PLoS ONE*, 11(2-e0150171), 2016.

[7] Giorgio C Buttazzo, Enrico Bini, and Darren Buttle. Rate-adaptive tasks: Model, analysis, and design issues. In *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–6. IEEE, 2014.

[8] Alvaro A Cárdenas, Saurabh Amin, Zong-Syun Lin, Yu-Lun Huang, Chi-Yen Huang, and Shankar Sastry. Attacks against process control systems: risk assessment, detection, and response. In *Proceedings of the 6th ACM symposium on information, computer and communications security*, pages 355–366, 2011.

[9] Yuanliang Chen, Yu Jiang, Fuchen Ma, Jie Liang, Mingzhe Wang, Chijin Zhou, Xun Jiao, and Zhuo Su. Enfuzz: Ensemble fuzzing with seed synchronization among diverse fuzzers. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 1967–1983, 2019.

[10] Octav Chipara, Chenyang Lu, and Gruia-Catalin Roman. Real-time query scheduling for wireless sensor networks. *IEEE transactions on computers*, 62(9):1850–1865, 2013.

[11] Octav Chipara, Chengjie Wu, Chenyang Lu, and William Griswold. Interference-aware real-time flow scheduling for wireless sensor networks. In *Real-Time Systems (ECRTS), 2011 23rd Euromicro Conference on*, pages 67–77. IEEE, 2011.

[12] Hongjun Choi, Wen-Chuan Lee, Yousra Aafer, Fan Fei, Zhan Tu, Xiangyu Zhang, Dongyan Xu, and Xinyan Deng. Detecting attacks against robotic vehicles: A control invariant approach. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 801–816, 2018.

[13] Tanya L. Crenshaw, Spencer Hoke, Ajay Tirumala, and Marco Caccamo. Robust implicit edf: A wireless mac protocol for collaborative real-time systems. *ACM Trans. Embed. Comput. Syst.*, 2007.

[14] Leonardo De Moura and Nikolaj Bjørner. Z3: An efficient smt solver. *Tools and Algorithms for the Construction and Analysis of Systems*, pages 337–340, 2008.

[15] Flavio M De Paula, Marcel Gort, Alan J Hu, Steven JE Wilton, and Jin Yang. Backspace: formal analysis for post-silicon debug. In *Proceedings of the 2008 International Conference on Formal Methods in Computer-Aided Design*, page 5. IEEE Press, 2008.

[16] S. Drzevitzky. Proof-carrying hardware: Runtime formal verification for secure dynamic reconfiguration. In *International Conference on Field Programmable Logic and Applications*, pages 255–258, 2010.

[17] Dave Evans. The internet of things: How the next evolution of the internet is changing everything. *CISCO white paper*, 1(2011):1–11, 2011.

[18] James P Farwell and Rafal Rohozinski. Stuxnet and the future of cyber war. *Survival*, 53(1):23–40, 2011.

[19] Fan Fei, Zhan Tu, Ruikun Yu, Taegyu Kim, Xiangyu Zhang, Dongyan Xu, and Xinyan Deng. Cross-layer retrofitting of uavs against cyber-physical attacks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 550–557. IEEE, 2018.

[20] Arun Ganesan, Jayanthi Rao, and Kang Shin. Exploiting consistency among heterogeneous sensors for vehicle anomaly detection. Technical report, SAE Technical Paper, 2017.

[21] Wei Gao and Thomas H Morris. On cyber attacks and signature based intrusion detection for modbus based industrial control systems. *Journal of Digital Forensics, Security and Law*, 9(1):3, 2014.

[22] Arthur Gatouillat, Youakim Badr, and Bertrand Massot. Qos-driven self-adaptation for critical iot-based systems. In *International Conference on Service-Oriented Computing*, pages 93–105. Springer, 2017.

[23] Jairo Giraldo, David Urbina, Alvaro Cardenas, Junia Valente, Mustafa Faisal, Justin Ruths, Nils Ole Tippenhauer, Henrik Sandberg, and Richard Candell. A survey of physics-based attack detection in cyber-physical systems. *ACM Computing Surveys (CSUR)*, 51(4):1–36, 2018.

[24] Antoine Girard and George J Pappas. Hierarchical control system design using approximate simulation. *Automatica*, 45(2):566–571, 2009.

[25] Antoine Girard and George J Pappas. Approximate bisimulation: A bridge between computer science and control theory. *European Journal of Control*, 17(5-6):568–578, 2011.

[26] Tao Gong, Tianyu Zhang, Xiaobo Sharon Hu, Qingxu Deng, Michael Lemmon, and Song Han. Reliable dynamic packet scheduling over lossy real-time wireless networks. In *31st Euromicro Conference on Real-Time Systems (ECRTS 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[27] Pinyao Guo, Hunmin Kim, Nurali Virani, Jun Xu, Minghui Zhu, and Peng Liu. Exploiting physical dynamics to detect actuator and sensor attacks in mobile robots. *arXiv preprint arXiv:1708.01834*, 2017.

[28] Pinyao Guo, Hunmin Kim, Nurali Virani, Jun Xu, Minghui Zhu, and Peng Liu. Roboads: Anomaly detection against sensor and actuator misbehaviors in mobile robots. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 574–585. IEEE, 2018.

[29] Xiaolong Guo, Raj Gautam Dutta, Jiaji He, Mark M Tehranipoor, and Yier Jin. Qif-verilog: Quantitative information-flow based hardware description languages for pre-silicon security assessment. In *2019 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pages 91–100. IEEE, 2019.

[30] Xiaolong Guo, Raj Gautam Dutta, Yier Jin, Farimah Farahmandi, and Prabhat Mishra. Pre-silicon security verification and validation: A formal perspective. In *Proceedings of the 52Nd Annual Design Automation Conference*, DAC '15, pages 145:1–145:6, 2015.

[31] Song Han, Xiuming Zhu, Aloysius K Mok, Deji Chen, and Mark Nixon. Reliable and real-time communication in industrial wireless mesh networks. In *2011 17th IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 3–12. IEEE, 2011.

[32] Tianjia He, Lin Zhang, Fanxin Kong, and Asif Salekin. Exploring inherent sensor redundancy for automotive anomaly detection. In *57th Design Automation Conference*, 2020.

[33] Thomas A Henzinger, Ranjit Jhala, Rupak Majumdar, and Grégoire Sutre. Software verification with blast. In *Model Checking Software*, pages 235–239. Springer, 2003.

[34] Abdulmalik Humayed, Jingqiang Lin, Fengjun Li, and Bo Luo. Cyber-physical systems security—a survey. *IEEE Internet of Things Journal*, 4(6):1802–1831, 2017.

[35] Xun Jiao, Dongning Ma, Wanli Chang, and Yu Jiang. Levax: An input-aware learning-based error model of voltage-scaled functional units. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(12):5032–5041, 2020.

[36] Cheng Jin, David X Wei, and Steven H Low. Fast tcp: motivation, architecture, algorithms, performance. In *IEEE INFOCOM 2004*, volume 4, pages 2490–2501. IEEE, 2004.

[37] Yier Jin. Design-for-security vs. design-for-testability: A case study on dft chain in cryptographic circuits. In *IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 19–24, 2014.

[38] Yier Jin, Bo Yang, and Yiorgos Makris. Cycle-accurate information assurance by proof-carrying based signal sensitivity tracing. In *IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, pages 99–106, 2013.

[39] Mason Kamb, Eurika Kaiser, Steven L Brunton, and J Nathan Kutz. Time-delay observables for koopman: Theory and applications. *SIAM Journal on Applied Dynamical Systems*, 19(2):886–917, 2020.

[40] Sanmeet Kaur and Maninder Singh. Automatic attack signature generation systems: A review. *IEEE Security & Privacy*, 11(6):54–61, 2013.

[41] Junsung Kim, K. Lakshmanan, and R. Rajkumar. Rhythmic tasks: A new task model with continually varying periods for cyber-physical systems. In *IEEE/ACM Third International Conference on Cyber-Physical Systems (ICCPS)*, pages 55–64, 2012.

[42] Fanxin Kong, Meng Xu, James Weimer, Oleg Sokolsky, and Insup Lee. Cyber-physical system checkpointing and recovery. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pages 22–31. IEEE, 2018.

[43] Vince Kurtz, Patrick M Wensing, Michael D Lemmon, and Hai Lin. Approximate simulation for template-based whole-body control. *arXiv preprint arXiv:2006.09921*, 2020.

[44] J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic mode decomposition: data-driven modeling of complex systems*. SIAM, 2016.

[45] Quan Leng, Yi-Hung Wei, Song Han, Aloysius K Mok, Wenlong Zhang, and Masayoshi Tomizuka. Improving control performance by minimizing jitter in rt-wifi networks. In *2014 IEEE Real-Time Systems Symposium*, pages 63–73. IEEE, 2014.

[46] Xun Li, Vineeth Kashyap, Jason K Oberg, Mohit Tiwari, Vasanth Ram Rajarathinam, Ryan Kastner, Timothy Sherwood, Ben Hardekopf, and Frederic T Chong. Sapper: A language for hardware-level security policy enforcement. In *ACM SIGARCH Computer Architecture News*, volume 42, pages 97–112. ACM, 2014.

[47] Xun Li, Mohit Tiwari, Jason K Oberg, Vineeth Kashyap, Frederic T Chong, Timothy Sherwood, and Ben Hardekopf. Caisson: a hardware description language for secure information flow. In *ACM SIGPLAN Notices*, volume 46, pages 109–120. ACM, 2011.

[48] E. Love, Y. Jin, and Y. Makris. Proof-carrying hardware intellectual property: A pathway to trusted module acquisition. *IEEE Transactions on Information Forensics and Security (TIFS)*, 7(1):25–40, 2012.

[49] Steven H Low and David E Lapsley. Optimization flow control. i. basic algorithm and convergence. *IEEE/ACM Transactions on networking*, 7(6):861–874, 1999.

[50] Dongning Ma, Jianmin Guo, Yu Jiang, and Xun Jiao. Hdtest: Differential fuzz testing of brain-inspired hyperdimensional computing. In *IEEE/ACM Design Automation Conference (DAC)*, 2021.

[51] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[52] Robert Mitchell and Ing-Ray Chen. A survey of intrusion detection techniques for cyber-physical systems. *ACM Computing Surveys (CSUR)*, 46(4):1–29, 2014.

[53] Robert Mitchell and Ray Chen. Adaptive intrusion detection of malicious unmanned air vehicles using behavior rule specifications. *IEEE Transactions on*

Systems, Man, and Cybernetics: Systems, 44(5):593–604, 2013.

[54] Robert Mitchell and Ray Chen. Behavior rule specification-based intrusion detection for safety critical medical cyber physical systems. *IEEE Transactions on Dependable and Secure Computing*, 12(1):16–30, 2014.

[55] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[56] Mehdi Mohammadi, Ala Al-Fuqaha, Mohsen Guizani, and Jun-Seok Oh. Semisupervised deep reinforcement learning in support of iot and smart city services. *IEEE Internet of Things Journal*, 5(2):624–635, 2017.

[57] Carlos Murguia and Justin Ruths. Cusum and chi-squared attack detection of compromised sensors. In *2016 IEEE Conference on Control Applications (CCA)*, pages 474–480. IEEE, 2016.

[58] Carlos Murguia and Justin Ruths. On reachable sets of hidden cps sensor attacks. In *2018 Annual American Control Conference (ACC)*, pages 178–184. IEEE, 2018.

[59] Michael Müter, André Groll, and Felix C Freiling. A structured approach to anomaly detection for in-vehicle networks. In *2010 Sixth International Conference on Information Assurance and Security*, pages 92–98. IEEE, 2010.

[60] Miroslav Pajic, James Weimer, Nicola Bezzo, Paulo Tabuada, Oleg Sokolsky, Insup Lee, and George J Pappas. Robustness of attack-resilient state estimators. In *ACM/IEEE 5th International Conference on Cyber-Physical Systems (ICCPS)*, pages 163–174. IEEE Computer Society, 2014.

[61] Junkil Park, Radoslav Ivanov, James Weimer, Miroslav Pajic, and Insup Lee. Sensor attack detection in the presence of transient faults. In *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*, 2015.

[62] Xin Peng, Bihuan Chen, Yijun Yu, and Wenyun Zhao. Self-tuning of software systems through dynamic quality tradeoff and value-based feedback control loop. *Journal of Systems and Software*, 85(12):2707–2719, 2012.

[63] Raul Quinonez, Jairo Giraldo, Luis Salazar, Erick Bauman, Alvaro Cardenas, and Zhiqiang Lin. SAVIOR: Securing autonomous vehicles with robust physical invariants. In *29th USENIX Security Symposium (USENIX Security 20)*, 2020.

[64] Marc H Raibert. *Legged robots that balance*. MIT press, 1986.

[65] Jeyavijayan Rajendran, Vivekananda Vedula, and Ramesh Karri. Detecting malicious modifications of data in third-party intellectual property cores. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, DAC '15, pages 112:1–112:6, New York, NY, USA, 2015.

[66] Aviva Hope Rutkin. "Spoofers" use fake gps signals to knock a yacht off course, August 14, 2013.

[67] Andrei Sabelfeld and Andrew C Myers. Language-based information-flow security. *IEEE Journal on selected areas in communications*, 21(1):5–19, 2003.

[68] Abusayeed Saifullah, You Xu, Chenyang Lu, and Yixin Chen. Real-time scheduling for wirelesshart networks. In *Real-Time Systems Symposium (RTSS), 2010 IEEE 31st*, pages 150–159. IEEE, 2010.

[69] Keisuke Sato, Yuichi Kawamoto, Hiroki Nishiyama, Nei Kato, and Yoshitaka Shimizu. A modeling technique utilizing feedback control theory for performance evaluation of iot system in real-time. In *2015 International Conference on Wireless Communications & Signal Processing (WCSP)*, pages 1–5. IEEE, 2015.

[70] Giordano Scarciotti and Alessandro Astolfi. Data-driven model reduction by moment matching for linear and nonlinear systems. *Automatica*, 79:340–351, 2017.

[71] Mo Sha, Rahav Dor, Gregory Hackmann, Chenyang Lu, Tae-Suk Kim, and Taerim Park. Self-adapting mac layer for wireless sensor networks. In *Real-Time Systems Symposium (RTSS), 2013 IEEE 34th*, pages 192–201. IEEE, 2013.

[72] Wei Shen, Tingting Zhang, Mikael Gidlund, and Felix Dobslaw. SAS-TDMA: a source aware scheduling algorithm for real-time communication in industrial wireless sensor networks. *Wireless Networks*, 2013.

[73] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646, 2016.

[74] Yasser Shoukry, Paul Martin, Paulo Tabuada, and Mani Srivastava. Non-invasive spoofing attacks for anti-lock braking systems. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 55–72. Springer, 2013.

[75] Emiliano Sisinni, Abusayeed Saifullah, Song Han, Ulf Jennehag, and Mikael Gidlund. Industrial internet of things: Challenges, opportunities, and directions. *IEEE Transactions on Industrial Informatics*, 2018.

[76] Geoffrey Smith. On the foundations of quantitative information flow. In *International Conference on Foundations of Software Science and Computational Structures*, pages 288–302. Springer, 2009.

[77] Yunmok Son, Hocheol Shin, Dongkwan Kim, Youngseok Park, Juhwan Noh, Kibum Choi, Jungwoo Choi, and Yongdae Kim. Rocking drones with intentional sound noise on gyroscopic sensors. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 881–896, 2015.

[78] Jianping Song, Song Han, Al Mok, Deji Chen, Mike Lucas, Mark Nixon, and Wally Pratt. Wirelesshart: Applying wireless technology in real-time industrial process control. In *2008 IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 377–386. IEEE, 2008.

[79] John A Stankovic. Research directions for the internet of things. *IEEE internet of things journal*, 1(1):3–9, 2014.

[80] Petcharat Suriyachai, Utz Roedig, and Andrew Scott. A survey of mac protocols for mission-critical applications in wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 14(2):240–264, 2011.

[81] Adrian Taylor, Sylvain Leblanc, and Nathalie Japkowicz. Anomaly detection in automobile control network data with long short-term memory networks. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 130–139. IEEE, 2016.

[82] Federico Tramarin, Aloysius K Mok, and Song Han. Real-time and reliable industrial control over wireless lans: Algorithms, protocols, and future directions. *Proceedings of the IEEE*, 2019.

[83] Rohit Tunga, Carlos Murguia, and Justin Ruths. Tuning windowed chi-squared detectors for sensor attacks. In *2018 Annual American Control Conference (ACC)*, pages 1752–1757. IEEE, 2018.

[84] David I Urbina, Jairo A Giraldo, Alvaro A Cardenas, Nils Ole Tippenhauer, Junia Valente, Mustafa Faisal, Justin Ruths, Richard Candell, and Henrik Sandberg. Limiting the impact of stealthy attacks on industrial control systems. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1092–1105, 2016.

[85] A.J. Van der Schaft. *L2-gain and passivity techniques in nonlinear control*. Springer, 2000.

[86] Jiachen Wang, Tianyu Zhang, Dawei Shen, X.S. Hu, and Song Han. APaS: an adaptive parition-based scheduling framework for 6TiSCH networks. In *Proceedings 27th IEEE Real-time and Embedded Technology and Applications Symposium (RTAS 2021)*, 2021.

[87] Mingzhe Wang, Jie Liang, Yuanliang Chen, Yu Jiang, Xun Jiao, Han Liu, Xibin Zhao, and Jiaguang Sun. Safl: increasing and accelerating testing coverage with symbolic execution and guided fuzzing. In *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings*, pages 61–64. ACM, 2018.

[88] Ruixuan Wang, Fanxin Kong, Hasshi Sudler, and Xun Jiao. Hdad: Hyperdimensional computing-based anomaly detection for automotive sensor attacks. In *27th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), Brief Industry Paper Track*. IEEE, 2021.

[89] Xin Wei, Jialin Zhao, Liang Zhou, and Yi Qian. Broad reinforcement learning for supporting fast autonomous iot. *IEEE Internet of Things Journal*, 7(8):7010–7020, 2020.

[90] Yi-Hung Wei, Quan Leng, Song Han, Aloysius K Mok, Wenlong Zhang, and Masayoshi Tomizuka. Rt-wifi: Real-time high-speed communication protocol for wireless cyber-physical control applications. In *2013 IEEE 34th Real-Time Systems Symposium*, pages 140–149. IEEE, 2013.

[91] Yifei Wei, F Richard Yu, Mei Song, and Zhu Han. Joint optimization of caching, computing, and radio resources for fog-enabled iot using natural actor–critic deep reinforcement learning. *IEEE Internet of Things Journal*, 6, 2018.

[92] Chen Yan, Hocheol Shin, Connor Bolton, Wenyuan Xu, Yongdae Kim, and Kevin Fu. Sok: A minimalist approach to formalizing analog sensor security. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 480–495, 2020.

[93] Man-Ki Yoon, Bo Liu, Naira Hovakimyan, and Lui Sha. Virtualdrone: virtual sensing, actuation, and communication for attack-resilient unmanned aerial systems. In *Proceedings of the 8th International Conference on Cyber-Physical Systems*, pages 143–154, 2017.

[94] Man-Ki Yoon, Sibin Mohan, Jaesik Choi, Jung-Eun Kim, and Lui Sha. Securecore: A multicore-based intrusion detection architecture for real-time embedded systems. In *2013 IEEE 19th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 21–32. IEEE, 2013.

[95] Danfeng Zhang, Yao Wang, G Edward Suh, and Andrew C Myers. A hardware design language for timing-sensitive information-flow security. *ACM SIGPLAN Notices*, 50(4):503–516, 2015.

[96] Lin Zhang, Xin Chen, Fanxin Kong, and Alvaro A. Cardenas. Real-time recovery for cyber-physical systems using linear approximations. In *41st IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 2020.

[97] Tianyu Zhang, Tao Gong, Chuancai Gu, Huayi Ji, Song Han, Qingxu Deng, and Xiaobo Sharon Hu. Distributed dynamic packet scheduling for handling disturbances in real-time wireless networks. In *RTAS*, 2017.

[98] Tianyu Zhang, Tao Gong, Song Han, Qingxu Deng, and X Sharon Hu. Fully distributed packet scheduling framework for handling disturbances in lossy real-time wireless networks. *IEEE Transactions on Mobile Computing*, 2019.

[99] Tianyu Zhang, Tao Gong, Song Han, Qingxu Deng, and Xiaobo Sharon Hu. Distributed dynamic packet scheduling framework for handling disturbances in real-time wireless networks. *IEEE Transactions on Mobile Computing*, 2018.

[100] Tianyu Zhang, Tao Gong, Zelin Yun, Song Han, Qingxu Deng, and Xiaobo Sharon Hu. Fd-pas: A fully distributed packet scheduling framework for handling disturbances in real-time wireless networks. In *RTAS*, 2018.

[101] Xuehui Zhang and M Tehranipoor. Case study: Detecting hardware trojans in third-party digital ip cores. In *HOST*, pages 67–70, 2011.

[102] Youqian Zhang and KB Rasmussen. Detection of electromagnetic interference attacks on sensor systems. In *IEEE Symposium on Security and Privacy (S&P)*, 2020.