# Security Framework for IoT End Nodes with Neural Networks

**Article** · August 2019

DOI: 10.18178/ijmlc.2019.9.4.814

Jesus Pacheco

Universidad de Sonora (Unison)

Victor Hugo Benitez Baltazar

Universidad de Sonora (Unison)

# Security Framework for IoT End Nodes with Neural Networks

Jesus Pacheco, Victor H. Benitez, and Zhiwen Pan

***Abstract*—The premise of the Internet of Things (IoT) is to connect not only computers and mobile devices, but also interconnect smart buildings, homes, and cities, as well as electrical and water grids, automobiles, and airplanes just to mention some examples. IoT leads to the development of a wide range of advanced information services that are pervasive, cost-effective, and can be accessed from anywhere and at any time. In this paper we present a multilayer architecture to integrate devices to the IoT, making it available from everywhere at any time. However, with the introduction of IoT we will be experiencing grand challenges to secure and protect its advanced information services due to the significant increase of the attack surface, complexity, heterogeneity, and number of interconnected resources. In order to deal with such challenges, we introduce an IoT Framework to build trustworthy and secure IoT applications and services. The framework enables developers to consider security issues at all IoT layers and integrate security algorithms with the functions and services offered in each layer instead of considering security in an ad-hoc and after thought manner. We show the applicability of our methodology to secure and protect IoT end nodes providing them with the capabilities for self-monitoring and self-recovering after an external event has occurred.**

***Index Terms*—Internet of things, access control, threat detection, neural networks.**

## I. Introduction

Advances in mobile and pervasive computing, and the exponential growth in sensors, actuators and controllers have led to the development of the Internet of Things (IoT). IoT will be a key enabling technology to develop smart services that will revolutionize the way we do business, maintain our health, manage critical infrastructures, conduct education, and how we secure, protect, and entertain ourselves [1], [2]. IoT-based services and systems are usually comprised of complex systems (e.g., Cyber Physical Systems, CPS) and characterized by interdependence, independence, cooperation, competition, distribution, and adaptation [3], [4]. In addition, IoT enables monitoring and controlling large number of heterogeneous devices and systems that are geographically dispersed by collecting, processing, and acting on the data generated by smart objects, systems or humans [5]. With the utilization of IoT devices and communication protocols, we are experiencing grand challenges to secure and protect such advanced services due to the significant increase in the attack surface [6]. Even devices which are intended to operate only in local area networks are sometimes connected to the Internet due to careless configuration or to satisfy special needs (e.g., they need to be remotely monitored). This makes IoT vulnerable to attacks that lead to incorrect information delivery to users, causing them to take wrong actions or to be unaware of an attack underway, as was the case with the Stuxnet attack [2], [3].

In this work we first introduce an IoT hierarchical architecture that can be used to deploy IoT applications. Then, we extend the architecture to our IoT security Framework. The main objective of introducing our framework is to enable developers to address security issues in a systematic way while designing and developing each IoT layer. In our approach, IoT hierarchy consists of four layers: Application, Service, Communications, and End-Devices layers. By insuring for each layer that all existing threats can be identified, and mitigation solutions will be applied, our framework will provide the architectural support to deliver trustworthy IoT services that can: 1) Protect IoT services against epidemic attacks; 2) Ensure that critical IoT systems can survive faults and destructive attacks; and 3) Ensure IoT security and privacy. To highlight the usability of our approach, we show how to use our framework to develop an anomaly behavior analysis based on neural networks, to detect attacks targeting effectors (e.g. DC motors) that are integrated to the IoT.

The rest of the paper is organized as follows. Section II shows the required background to understand IoT cyber security, Abnormal Behavior Analysis IDS, and the use of a threat model. Section III elaborates our decentralized security framework for IoT CPS. Section IV is devoted to explaining our anomaly behavior analysis (ABA) methodology for threat detection in IoT end nodes layer. Section V presents our experimental environment and discusses our evaluation results. Finally, we conclude the paper and discuss future research direction in Section VI.

## II. Background

### A. *IoT Cyber Security*

IoT can be viewed as a ubiquitous network that enables monitoring and controlling large number of heterogeneous devices that are geographically dispersed by collecting, processing, and acting on the data generated by intelligent end-to-end systems that enable smart solutions and covers a diverse range of technologies including sensing, communications, networking, etc. [7], [8]. Traditional IT security solutions are not directly applicable to IoT due to the

Manuscript received December 11 2018; revised June 12, 2019. This work was partially supported by the Engineering Division of Universidad de Sonora, and PRODEP under Grant 215589.

Jesus Pacheco and Victor H. Benitez are with the Universidad de Sonora, Blvd. Luis Encinas y Rosales, Col. Centro, Hermosillo, Sonora 83103, Mexico (e-mail: {jpacheco,vbenitez}@ industrial.uson.mx).

Zhiwen Pan is with Chinese Academy of Science, Beigin, China (e-mail: pzw@ict.ac.cn).

doi: 10.18178/ijmlc.2019.9.4.814

following issues [7]-[9]: 1) The IoT extends the “internet” through the traditional internet, mobile network, non-IP networks, sensor network, cloud computing, etc.; 2) Computing platforms are constrained in memory and processing capability and consequently may not support security algorithms; 3) All “things” will communicate with each other; and 4) Some IoT devices and services may be shared and could have different policies. These challenges need to be addressed in order to build a secure and resilient IoT infrastructure, where Confidentiality, Integrity, and Availability (CIA) must be assured [10].

### B. Anomaly Behavior Analysis

Current cyber-security solutions are far from being satisfactory to stop the exponential growth in number and complexity of cyber-attacks [11], [12]. There are two basic intrusion detection techniques to detect cyberattacks: signature based and anomaly based Intrusion Detection Systems (IDS) [13], [14]. Signature based IDS builds a database of known attack signatures or identities. However, these systems cannot detect new types of attacks or even a known attack with a slight change on its signature. The main feature of the anomaly detection approaches is their capability in detecting novel and new attacks. The Anomaly Based IDS defines a baseline model for normal behavior of the system through off-line training and consider any activity which lies outside of this normal model as anomaly [14].

Any attack, misconfiguration or misuse will lead to a deviation from the normal behavior; we name it as abnormal behavior. The main limitation of this approach is the large number of false alarms that can be produced. To overcome this limitation, our approach performs fine-grain anomaly behavior analysis as will be discussed in further detailed when we introduce our IDS approach.

### C. Threat Model

Improving security and reducing risks in smart systems heavily depends on analyzing threats, risks and vulnerabilities to develop the appropriate countermeasures and mitigate their exploitations [15]. To better understand the IoT security landscape, a general IoT threat model needs to be developed [16]. A threat model defines threat scenarios with associated risk distributions, likelihood of occurrence, and impact. When created in the design phase, a threat model helps to identify changes that need to be made to the design to mitigate potential threats. In general, the steps to create a threat model are [16]: 1) Identify attackers, assets, and threats, 2) Rank the threats, 3) Choose mitigation strategies, and 4) Build mitigation solutions based on these strategies. We will follow the mentioned steps to create the threat model for our IoT framework and then we will show how to use it to secure and protect our IoT end nodes.

## III. Decentralized Security Framework for IoT CPS

### A. IoT Hierarchical Architecture

There are several IoT architectures that can be applied to build services for smart infrastructures [17]-[19]. Fig. 1 shows our IoT architecture used to build trustworthy and resilient IoT applications, as well as to guide the security development of IoT. It consists of four layers: IoT end Nodes, Network, Services, and Applications. Roughly speaking cyberattacks can be launched against the functions and services provided by each layer. In what follows, we introduce each layer and illustrate potential attack surface, impact and mitigation mechanisms.

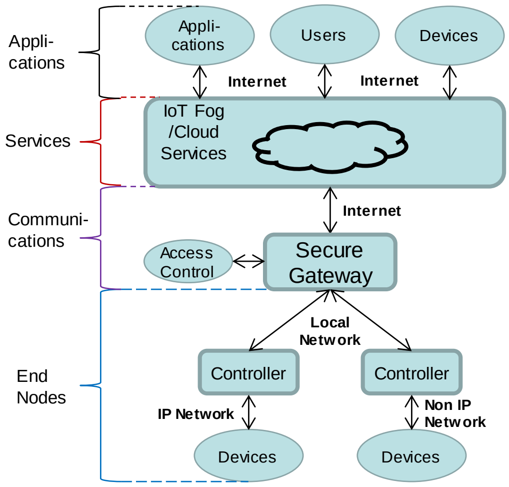


Fig. 1. IoT hierarchical architecture for cyber physical systems.

The end nodes layer includes components such as sensors for capturing and representing the physical world in the digital domain, and actuators to modify the environment to a desired state [5], [20]. It focuses on exchanged information about device properties and environmental conditions passes through physical devices to identify or sense the physical world. At this level, cyberattacks target local controllers, sensors, and actuators.

The network layer is responsible for reliable information transmissions from/to end nodes. The network layer includes mobile communication networks, wireless sensor networks, communication protocols, etc. [20]. Cyber-attacks at this layer target firewalls, routers, protocols, and personal information.

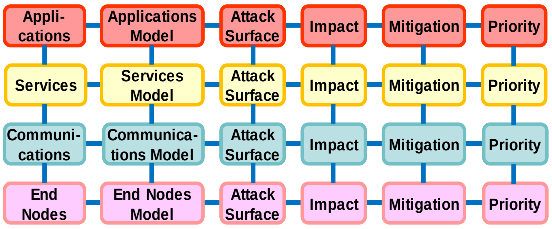
The services layer acts as an interface between the application layer and the network layer [21]. At this layer, required computational power is mostly provided as fog and cloud services. In this layer, cyberattacks target personal and confidential information, IoT end devices, monitor and control functions.

The application layer provides personalized services according to user needs [20], [21]. The access to IoT services is possible through mobile technology (i.e., cellphone, mobile applications) and smart appliances and devices. In this layer, data sharing is an important characteristic and consequently application security must address data privacy, access controls and information leaks.

Essentially, we need to ensure the security of all layers in a given IoT application. To achieve this task, we propose our security development framework to cope with the threats in each layer.

### B. IoT Security Development Framework (SDF)

The main goal of the SDF is to provide the architectural support to develop highly secure and trustworthy IoT services that can proactively detect and tolerate malicious behaviors that can be due to attacks, faults (malicious or natural) or accidents. We define a trustworthy service to be the one that can secure and protect the system against cyberattack (self-protection), that can continue to operate normally by meeting its performance requirements in spite of faults and destructive attacks (self-healing, and self-optimizing), and can update its configuration and security policies to maintain security, privacy, resilience to faults and accidents, and quality of service requirements (self-configuration). The SDF integrates security mechanism at the design and development stage as shown in Fig. 2.



| | | | | | |
|---|---|---|---|---|---|
| Appli-cations | Applications Model | Attack Surface | Impact | Mitigation | Priority |
| Services | Services Model | Attack Surface | Impact | Mitigation | Priority |
| Communi-cations | Communica-tions Model | Attack Surface | Impact | Mitigation | Priority |
| End Nodes | End Nodes Model | Attack Surface | Impact | Mitigation | Priority |

Fig. 2. IoT security development framework.

The SDF is organized as 2-Dimensional architecture with four layers mapped to our IoT hierarchical architecture. Each layer is implemented into five planes: Function Specification (model), Attack Surface, Impact, Mitigation, and Priority planes. For each layer, we first identify the Attack Surface that characterizes the entry points that can be exploited by attackers to inject malicious events or behaviors in the IoT environment, followed by identifying potential impact of exploding the vulnerabilities, then we identify the mitigation mechanisms that can be implemented to diminish these attacks, and finally we prioritize the service according to the potential impact to the system. By following this methodology, we can ensure the development of highly secure and trustworthy IoT applications.

## IV. Anomaly Behavior Analysis for IoT End Nodes Layer

In our previous work in [2], we developed strategies based on wavelets to detect if sensors in the IoT end nodes are being compromised and then take the required actions. In this work we will show how to use the SDF with a strategy based on neural networks to verify if the actuators in the IoT end nodes are behaving correctly or not (detect abnormal behavior).

Our ABA methodology uses as principle that systems normal behavior can be characterized using global variables, for instance in a DC motor can be speed, position, torque, energy consumption, etc. In general, we follow the ABA deployment methodology depicted in Fig. 3.

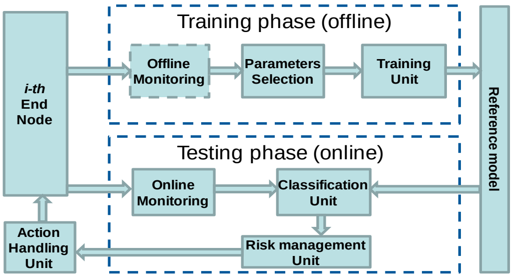


Fig. 3. IoT security deployment for end nodes layer.

### A. *Training Phase*

This phase aims to model the end-nodes. This step is conducted offline. The outcome is a reference model that characterizes the normal behavior of the end nodes. In what follows, we describe each step in the training phase.

#### 1) *Offline monitoring*

This module is in charge of continuously monitoring the performance of the device at the end node. All the parameters are collected here depending on the type of system being inspected. Usually the offline monitoring is performed by a local controller or computer.

#### 2) *Parameters selection*

As we are talking about Cyber Physical Systems, we need to accurately choose the parameters that better describe the system. For instance, if the device at the end node is a DC motor, we are interested in collecting information about motor speed and current consumption.

#### 3) *Training unit*

The training unit is responsible for providing the model of the device being inspected, and to verify that the control unit can understand the given model. For this step we can lean on predefined model (e.g., given by the provider), or we can obtain a parametric model using intelligent control techniques such as Artificial Neural Networks.

### B. *Reference Model*

Our proposed scheme is intended to work with CPS that can perform self-optimization (automatic monitoring and control of resources), self-protection (proactive identification and protection from threats), self-regulation (maintain steady state without external control), and self-management (manage itself without external intervention). To achieve these goals, we proposed the decentralized control scheme shown in Fig. 4, which represents an arbitrary connected device.

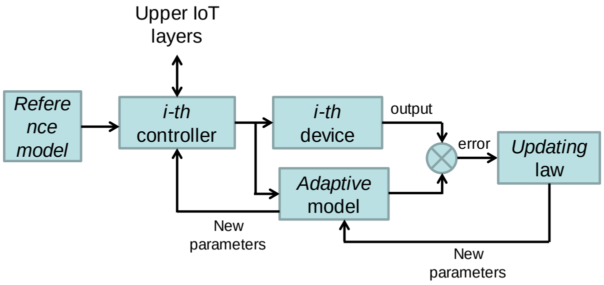


Fig. 4. Decentralized control scheme for the i-th node.

Decentralized control scheme is a robust fruitful technique that has been used to address large-scale complex systems such as transportation systems, electric power grids, and communication networks, to mention few [22]. In what follows we describe the functional building blocks of our proposed approach applied to the i-th node.

#### 1) *i-th Device*

The building block device includes any of the IoT end nodes whose task involves managing or processing for the upper layers. For control purposes this block can be any object (plant or device), such that a model description is not available, or very hard to obtain due to high complexity or physical restrictions (e.g., distribution on large geographical locations). We consider that the model of the plant is described as a perturbed affine nonlinear system that can be represented by (1).

$$\begin{aligned} \dot{x}_i &= f_i(x_i) + g_i(x_i)u_i + \xi_i(t) \\ y_i &= h_i(t) + \upsilon_i(t) \end{aligned} \tag{1}$$

*2) Adaptive model*

The decentralized adaptive system considers the dynamics of the device considering two approaches for the fault detection task: 1) Internal Fault: Due to the dynamical response of the device which is affected by its internal operation and by perturbation signals that are inherent at the rated conditions. 2) External Fault: Due to cyber-attack. The adaptive model must be able to distinguish from an internal/external fault (to detect a threat), and to restore the i-th node to its healthy operative condition. For the model of the i-th device described by (1) we propose a decentralized recurrent high-order neural network model (D-RHONN) whose properties have been proved in [23] as an excellent adaptive identifier for nonlinear plants.

*3) Updating law*

The outputs of the device and their corresponding model are compared to generate an error signal. Based on the nature of the faults, two types of errors are considered: dynamic errors ed, which occurs due to dynamical behavior of the plant; non-dynamical error ~ed, due to a cyber-attack performed by external entities. The composition of error et, is described by (2).

$$et = ed + \sim ed \tag{2}$$

The total error drives a learning law block, which generates new parameters to the adaptive model in a closed loop architecture that converges to the device output once that *et* = 0 (or very close to it).

### C. Testing Phase

Once we have obtained our model, the next step is to verify if the node that is sending the information to the upper layers is being compromised.

*1) Online monitoring*

This module is in charge of continuously monitoring the performance of the device at the end node layer. It works in a similar way as in the offline; however, it filters the required parameters to send only required information to the classification unit.

*2) Classification unit*

This unit is the core of our model. It performs the identification based on the model obtained offline. This unit can be used to authenticate the node that is sending information from a given location (or IP). The parameters are obtained online from the monitoring unit and then compared against the reference model. It is clear that, for a dynamic system, subject to external cyber physical disturbances, the reference model may be not enough to accurately identify the inspected node, and hence we need to verify the error. Given the error et, in (2), we can compute the total error as shown in (3).

$$\mathcal{L}_2|e_t| = \sqrt{\frac{1}{T}\int e_t{}^2\,dt} = k \tag{3}$$

where we can statistically establish a threshold for $k$ for a time of interest $T$ (Montgomery 2009), such that if the threshold is bypassed, the end node is either under attack or a malicious device is trying to send information on behalf of the legitimate one (e.g. Impersonation attack).

## V. Experimental Results

In order to validate our IoT framework which integrate autonomic threat detection, we test the capability of our threat detection to identify if the connected node that is sending information to the upper layers is being compromised. The goal is to map the parameters of the connected node in the model stored in our controller and use this model as authentication mechanism to verify the health condition of the end node that is spreading information into the IoT ecosystem.

### A. End Nodes Testbed

We prototype an IoT end nodes testbed, which obtains all the characteristics and functionalities of the actual IoT end nodes such as sensors, actuators, automation systems, and communication channels. The elements to control (actuators) are lights, lamps, DC motors, door lock, and electric sockets (where televisions can be connected), etc. The information from the sensors is acquired by an Arduino board [24] every millisecond but updated in memory every 5 milliseconds. The main tasks of the Arduino board are: 1) collecting information from sensors, 2) analyzing devise usages looking for abnormalities such as high-frequency (speed changes), command issues, or excessive power consumption, 3) triggering alerts in case an abnormality is detected, and 4) sending the collected information to the secure gateway (including the alerts) through serial communication.

### B. Threat Detection Experimental Setup

We used two DC motors as devices in IoT end-nodes, according to the proposed decentralized scheme in Fig. 4.

*1) Adaptive model*

A neural network configured as a nonlinear autoregressive with external input (NARX) architecture is designed as the adaptive model block [25]. The neural model fits the input - output map of DC motors.

*2) Controller*

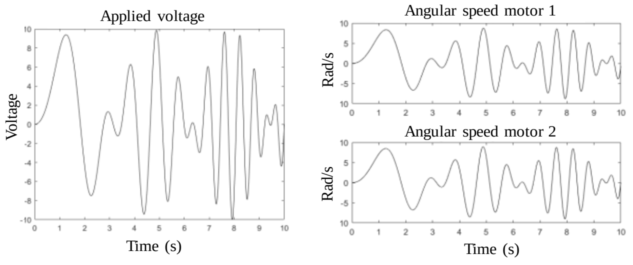
It is well known that the majority of industrial applications rely in Proportional- Integral-Derivative (PID) controllers (Xuemei Zhu 2009). Therefore, we chose a PID controller which will be tuned to control angular speed of DC motors.

*3) Updating learning law*

The neural network is trained with Back Propagation learning rule based on Levenberg Marquardt algorithm [26], which is specifically designed to minimize sum-of-square error functions.

### C. Experimental Results

DC motors are excited offline using a chirp signal modulated by a sinus component with amplitude 10 units and a linear variation from 0.1 to 2 Hz. This stage is necessary to excite all possible frequencies (modes) of DC motors and to get all its range of operation. The modulated chirp signal and respective output response for both motors are shown in Fig. 5.



a) Input b) Output (both motors)

Fig. 5. Experimental motor data input (a) and outputs (b).

Input - output data is used to train the neural network, which is designed with a single hiding layer with 5 neurons; two delayed input - outputs are selected. Data are divide as follows: 70% is used to train the neural network; 15% is used to validate and remaining 15% is selected to test the identification performance of the neural model. The neural weights obtained by the learning law for the hidden layer are listed in Table I. Notice that Table I represents the normal or nominal behavior of motors, modeled by an adaptive neural model. Such model can be used by the upper layers in our framework to identify the nominal operation of nodes. Also, it is worth to mention that end nodes are modeled by an adaptive model.

TABLE I: SYNAPTIC WEIGHTS

| Hidden Layer Weights | | | |
|---|---|---|---|
| Weights{1,1} | | Weights{1,2} | |
| -0.1532 | -1.0154 | -0.4542 | 1.5057 |
| -0.8451 | 0.9395 | 0.0387 | 0.4292 |
| -0.4002 | -1.4983 | 1.8145 | -0.1946 |
| 0.0085 | 0.0161 | 0.6221 | -0.3135 |
| -1.0414 | 1.0004 | -0.3160 | -0.1904 |

### 1) Threat detection

An abnormal operation is simulated where motors are subject to a sudden change in speed (i.e. motor goes from a constant speed to a cero speed instantaneously). This fault condition can be considered as severe because industrial processes normally operates under speed profiles that never exhibit sudden changes. Under this condition, the local controller for each node is able to recover to a healthy condition. If fault operation persists the upper layers in the framework can apply the appropriated corrective actions to restore the nodes to their nominal operation status. Table II shows the hidden layer weights under abnormal operation.

TABLE II: ABNORMAL OPERATION EXAMPLE

| Hidden Layer Weights | |
|---|---|
| Weights{1,1} | Weights{1,2} |
| 1.1171 | -1.1133 |
| -1.2926 | 1.2074 |
| -1.1405 | 1.1242 |
| -0.2295 | -0.9482 |
| -0.0739 | 1.5081 |

Even though, in fault operation case, the controller can follow the signal input, an action is required by the upper levels in order to identify if this behavior is due to a dysfunctional operation of nodes or by an intrusion. In any case, the neural network data obtained in this operative mode is called Data_fault. Equation (3) is applied as a criterion to distinguish the source of abnormality.

$$K = \mathcal{L}_2\left|Data_{actual} - Data_{fault}\right| = \begin{bmatrix} 5.6178 & 3.4477 \\ 2.2614 & 8.0191 \\ 2.9344 & 3.5736 \\ 0.0000 & 1.2126 \\ 1.0698 & 0.0898 \end{bmatrix}$$

The gap between matrix K and the Data_actual reference matrix is evident. If the entries of K are bounded, it is possible that the risk management and the action handling units can take the appropriate commands to restore the CPS to a healthy condition. Moreover, because PID controllers remains in the transient error close to zero and the adaptive neural model converges to the dynamics of the plant fast enough it is possible to establish that this fault condition is induced by an external attack. Fig. 6 shows this scenario where a speed fault is introduced (from 10 to 0 rad/sec) after 1 second of normal operation.

Compared with signature-based techniques [2], [3], [5], our proposed method is able to detect unknown abnormalities. This is possible because given the model for normal operation, which is not taken into consideration when using signature-based detectors.

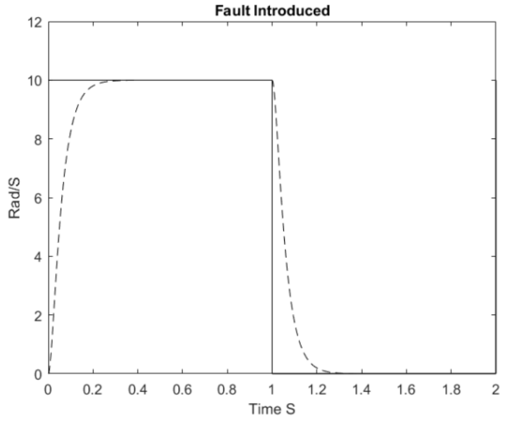


Fig. 6. A speed fault introduced after 1 second of normal operation.

## VI. DISCUSSION AND FUTURE WORK

In this paper, we presented a framework with a threat model that can be used to identify potential attacks against each layer of our framework, estimate their impacts and to mitigate and recover from these attacks. In addition, we integrate an autonomic threat detection to generally create a reference model for end nodes that describes its normal behavior and perform behavior analysis at runtime, to detect attack surface and, in particularly, abnormal behavior. We showed that our threat detection recognizes both known and unknown threats with high detection rate and low false positive alarms. It is important to emphasize that our proposed methodology is intended to protect the normal operation of IoT end nodes preventing the dissemination of attacks to other layers.

For the future work, we are working on extending our methodology with transitive relationships to express complex access control scenarios with multiple objects. In addition, we are also inspecting the possibility of detecting attacks by analyzing the behavior of other IoT layers. For example, if an attacker collects enough sensors' information (e.g., 1 day of information), it can launch a replay attack without the need of using the same data set.

## REFERENCES

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22-32, 2014.

[2] J. Pacheco and S. Hariri, "IoT security framework for smart cyber infrastructures," in *Proc. IEEE International Workshops on Foundations and Applications of Self* Systems*, 2016, pp. 242-247.

[3] V. Chiprianov, L. Gallon, M. Munier, P. Aniorte, and V. Lalanne. "Challenges in security engineering of systems-of-systems," *Troisième Conférence en IngénieriE du Logiciel*, p. 143, 2014.

[4] B. Boehm, A. Lane, P. M. Kern, A. C. Jost, R. Thayer, R. J. Leach, R. Valerdi, A. M. Ross, and D. H. Rhodes, "Systems engineering crosstalk," *Crosstalk*, vol. 801, pp. 775-5555.

[5] H. Suo, J. Wan, C. Zou, and J. Liu, "Security in the internet of things: A review," in *Proc. 2012 International Conference on Computer Science and Electronics Engineering (ICCSEE)*, 2012, vol. 3, pp. 648-651.

[6] P. Nassar, Y. Badr, F. Biennier, and K. Barbar, "Securing collaborative business processes: A methodology for security management in service-based infrastructure," in *Proc. IFIP International Conference on Advances in Production Management Systems*, pp. 480-487, Springer, Berlin, Heidelberg, 2011.

[7] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac. "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497-1516, 2012.

[8] M. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. Alfredo, G. Boggia, and M. Dohler. "Standardized protocol stack for the internet of (important) things," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1389-1406, 2013.

[9] J. Zhou, Z. Cao, X. Dong, and A.V. Vasilakos, "Security and privacy for cloud-based IoT: challenges," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 26-33, 2017.

[10] E. Carrillo, V. Benitez, C. Mendoza, and J. Pacheco, "IoT framework for smart buildings with cloud computing," in *Proc. 2015 IEEE First International Smart Cities Conference (ISC2)*, IEEE, 2015, pp. 1-6.

[11] Q. Yaseen, F. AlBalas, Y. Jararweh, and M. Al-Ayyoub, "A fog computing based system for selective forwarding detection in mobile wireless sensor networks," in *Proc. IEEE International Workshops on Foundations and Applications of Self* Systems*, pp. 256-262, 2016.

[12] Y. Yang, L. Wu, and W. Hu. "Security architecture and key technologies for power cloud computing," in *Proc. 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE)*, 2011, pp. 1717-1720.

[13] O. Can and O. K. Sahingoz, "A survey of intrusion detection systems in wireless sensor networks," in *Proc. 2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*, pp. 1-6, 2015.

[14] S. Fayssal, S. Hariri, and Y. Al-Nashif, "Anomaly-based behavior analysis of wireless network security," in *Proc. Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services*, pp. 1-8, 2007.

[15] D. Xu, M. Tu, M. Sanford, L. Thomas, D. Woodraska, and W. Xu, "Automated security test generation with formal threat models," *IEEE Transactions on Dependable and Secure Computing*, vol. 9, no. 4, 526-540, 2012.

[16] R. Schlegel, S. Obermeier, and J. Schneider, "Structured system threat modeling and mitigation analysis for industrial automation systems," in *Proc. 2015 IEEE 13th International Conference on Industrial Informatics,* pp. 197-203, 2015.

[17] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, 2013, 1645-1660.

[18] H. Ferreira, E. Dias, and R.T. de Sousa. "IoT architecture to enable intercommunication through REST API and UPnP using IP, ZigBee and arduino," in *Proc. 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2013, pp. 53-60.

[19] M. D. Brito, S. Hoque, R. Steinke, and A. Willner, "Towards programmable fog nodes in smart factories," in *Proc. IEEE International Workshops on Foundations and Applications of Self* Systems*, 2016, pp. 236-241.

[20] M. Hossain, M. Fotouhi, and R. Hasan, "Towards an analysis of security issues, challenges, and open problems in the internet of things," in *Proc. 2015 IEEE World Congress on Services*, 2015, pp. 21-28.

[21] Manadhata, K. Pratyusa, and J. M. Wing, "An attack surface metric," *IEEE Transactions on Software Engineering*, vol. 3, 2010, pp. 371-386.

[22] P. Jokar, H. Nicanfar, and V. Leung, "Specification-based intrusion detection for home area networks in smart grids," in *Proc. 2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2011, pp. 208-213.

[23] V. H. Benitez, E. N. Sanchez, and A. G. Loukianov, "Decentralized adaptive recurrent neural control structure," *Engineering Applications of Artificial Intelligence*, vol. 20, no. 8, pp. 1125-1132, 2007.

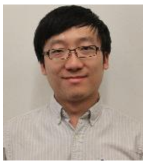
[24] Y. Badamasi. "The working principle of an Arduino," in *Proc. 2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*, 2014, pp. 1-4.

[25] S. A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*, John Wiley & Sons, 2013.

[26] S. S. Haykin. *Neural Networks and Learning Machines*, vol. 3, Upper Saddle River, NJ, USA: Pearson, 2009.

**Jesus Pacheco** is in the Electrical and Computer Engineering Department, University of Arizona. He is a full professor in the Industrial Engineering Department of the Universidad de Sonora. His research interest includes cyber security for critical infrastructures and cyber-physical systems.

**Victor H. Benitez** got the PhD in electrical engineering from Advanced Studies and Research Center, CINVESTAV-IPN, in 2009 respectively. He has worked in the manufacturing industry related to the manufacture of printed circuits boards, and in the area of testing engineering. He is a full professor from the Industrial Engineering Department, Mechatronics Area of the Universidad de Sonora. His research interests include myoelectric control systems; neural networks applied to control electromechanical systems and mechatronics design applied to solar concentration systems



**Zhiwen Pan** got the PhD from the Electrical and Computer Engineering Department, University of Arizona. He is an assistant researcher from Chinese Academy of Science, Beijing, China. His research interest includes cyber security for critical infrastructures and context awareness IDS.