Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Securing Internet of Things (IoT) with Machine Learning

Sherali Zeadally*[1] | Michael Tsikerdekis[2]

[1]College of Communication and Information, University of Kentucky, KY, USA
[2]Computer Science Department, Western Washington University, WA, USA

Correspondence
*Corresponding author: Email: szeadally@uky.edu

Present Address
Sherali Zeadally
315 Little Library Building
College of Communication and Information
University of Kentucky
Lexington, KY 40506-0224

Summary

Advances in hardware, software, communication, embedding computing technologies along with their decreasing costs and increasing performance have led to the emergence of the Internet of Things (IoT) paradigm. Today, several billions of Internet-connected devices are part of the IoT ecosystem. IoT devices have become an integral part of the Information Communication Technology (ICT) infrastructure that supports many of our daily activities. The security of these IoT devices has been receiving a lot of attention in recent years. Another major recent trend is the amount data that is being produced every day which has reignited interest in technologies such as machine learning and artificial intelligence. We investigate the potential of machine learning techniques in enhancing the security of IoT devices. We focus on the deployment of supervised, unsupervised learning techniques, as well as reinforcement learning for both host-based and network-based security solutions in the IoT environment. Finally, we discuss some of the challenges of machine learning techniques that need to be addressed in order to effectively implement and deploy them so that they can better protect IoT devices.

KEYWORDS:
Attack, Internet of Things, Machine learning, Security

1 | INTRODUCTION

The last two decades have witnessed the rapid convergence of the physical world and the digital world. Initially, the Internet played a crucial role in connecting the two worlds. As embedded computers become smaller, more powerful, and affordable, they are being deployed in many devices and systems cost-effectively making these devices Internet-enabled and reachable from any device, anytime, from anywhere. These technological developments have paved the way for the emergence of the Internet of Things (IoT). Today, numerous IoT application and devices have been deployed in various sectors including healthcare, transportation, smart home, manufacturing, and so on to provide value added services to consumers[1]. Several recent reports have projected a rise in the number of connected devices in the home. For instance, one report[2] has predicted the number of connected devices per Australian household will be over 30 by 2021. In the US, broadband households have more than 10 connected devices[3] which include PC/Mac, smartphone, tablet, and in-home entertainment platforms such as gaming console, smart TV, streaming media player, Blu-ray player, DVR, and so on.

Recently, there has been a growing interest in exploring the potential of machine learning technologies to improve IoT security. This interest stems from the rise of intelligent adversaries powered by the proliferation of machine learning tools and the rise of big data as well as the inability for signature-based defenses to adapt quickly enough to zero day threats[4,5]. Machine learning approaches differ in their efficiency and computational requirements. For example, some supervised models have high

CPU and memory requirements during their training phase but require less resources once implemented compared to unsupervised learning models. When machine learning techniques are used to improve the security of networks they are executed on traditional networked devices (e.g., laptops, routers, and so on)[4] and as such they are uniquely positioned to help protect the IoT infrastructure.

1.1 | Research contributions of this work

Most security approaches for IoT devices focus on passive measures (e.g., encryption) on these devices (or rely on traditional network security monitoring practices (e.g., signature-based intrusion detection). Limited progress has been made toward identifying viable network-based machine learning defenses for IoT as well as host-based machine learning defense techniques that can be used on these devices. Furthermore, when deploying machine learning-based security techniques for IoT devices, we need to consider both the limitations associated with various categories machine learning algorithms (supervised, unsupervised, and reinforced) as well as the specific characteristics (such as limited computing and storage resources) of these IoT devices.

We summarize the main research contributions of this work as follows:

- We briefly review the characteristics of IoT devices and the common attacks that have been launched against them.
- We present a taxonomy of network-based and host-based machine learning approaches (along with their strengths and weaknesses) that can be used to improve IoT security in practice.
- We discuss some key challenges that need to be addressed in the future to enable stronger and more efficient IoT security with machine learning techniques.

2 | BACKGROUND AND MACHINE LEARNING

2.1 | Internet of Things

Internet of Things provides users ease of use of Internet-connected devices and seamless connectivity both of which support our daily activities in cyberspace. As more devices connect to the IoT ecosystem, the potential attack surface becomes larger. In fact, given the large number of IoT devices that are interconnected in the IoT environment, the possibilities for amplification attacks are much higher. Securing IoT devices in cyberspace remains a significant challenge. For instance, using tools such as Shodan, a search engine for IoT devices, millions of devices (e.g., home security cameras) with critical ports (e.g., port numbers 143 for the Internet Message Access Protocol or 445 for the Microsoft Directory Services) have been discovered. Many of these devices continue to use default login credentials. In fact, with the rapid proliferation of IoT, insecure protocols such as Telnet are frequently being used again. Moreover, many of these Internet-enabled devices operate using wireless communication technologies which make these devices accessible outside the traditional wired network perimeter if they are not secured properly.

An example of such an attack is the recent data exfiltration that was conducted through a casino's Internet-connected aquarium thermostat device (although it was residing on the casino's Intranet)[6]. The attackers used the device to access the internal network and transferred several gigabytes of data from the casino's high-roller database that contained private information about its richest high-roller guests out of the network. The impact of infiltrated IoT devices is not isolated to the individuals or organizations that host these devices on their networks. Compromised IoT devices can be used as part of a botnet to launch large-scale Denial-of-Service (DoS) attacks on other systems and networks[7,8,9,10]. A recent report has quantified the cost that such attacks have on organizations and society at large[11]. The cost of IoT hacks for small U.S. firms amounts to 13% of their annual revenue[12].

Strong, cost-effective IoT security is harder to achieve compared with traditional information security because of several reasons which include[13]: a) the environment these IoT devices reside. Today, many IoT devices use wireless connections which may be accessible outside the organization's internal network perimeter; b) Firmware update and vulnerability patching are frequently done on standard IT systems. This is not quite the case with consumer IoT security wherein automatic updates of firmware/software and patching mechanisms are not well implemented. In many cases of IoT devices, the user is left to do the updates or patching which in most cases is not effective; c) the limited storage, size, and computing power of many IoT devices make it difficult to apply the same security controls that are deployed on traditional networked computing systems.

2.2 | Machine Learning

The big data era has been fueled by the explosive growth of data that is being generated by diverse sources including systems (e.g. devices, sensors), people, services, and others. Today, many data-driven techniques are being applied to the huge amounts of data produced in order to reap important benefits [13] such as value-added services, resource optimization, new functionality, and so on. Recently, there has been a strong renewed interest in the capabilities of machine learning in several application domains. One area where we have witnessed increasing interests in the use of machine learning techniques is cybersecurity. Machine learning algorithms can be used to solve classification problems (e.g., predicting stocks or identifying whether a fruit is an apple or pear) as well as clustering problems (e.g., identifying distinct groups of people based on their socioeconomic background) [14,15,16]. Classification problems (i.e., data are labeled prior to the analysis) are often solved using *Supervised Learning* algorithms such as Random Forests [17] or Neural Networks [16]. Clustering problems, which involve unlabeled data, are often solved using *Unsupervised Learning* algorithms such as k-means clustering [18] and DBSCAN [19]. Finally, *Reinforcement Learning* is a distinct category in machine learning algorithms that involves identifying solutions that aim to either maximize or minimize a function (e.g., agent actions based on a set of states that prolong interaction with an adversary). These models use a Markov decision process to model state and actions and iterate in order to "learn". An algorithm in this category is Q-Learning [20].

3 | IOT CHARACTERISTICS AND COMMON ATTACKS

IoT devices are required to operate in various conditions that are not traditionally the same as in the operational environments of common computing devices. Often, they have specialized functions and objectives that are supported by their unique characteristics. The security vulnerabilities of IoT devices are generally exploited by using traditional attack methods which often target their unique properties. Next, we present some typical characteristics of IoT devices and common attacks on them.

3.1 | Characteristics of IoT devices

We identify a few characteristics that are unique to IoT devices.

Sensing

Various types of sensors are available for implementation into IoT devices and several may be implemented in a device depending on the application domain. These include body sensors, environmental sensors, vehicle sensors among many others [21]. However, these sensors also contribute to measurement error rates or missing data altogether (e.g., sensor misreads or fails to read for a particular interval). This is a well known limitation of sensors and studies often propose the use of various statistical methods to correct these data anomalies [22].

Dynamic states

IoT devices operate as finite-state automata at the device-level more so than any other computing device. They are expected to transition between states (e.g., sleeping, active) depending on environmental changes and their programming protocol. These changes also generate unique conditions related to the state of data availability. For example, a sensor may not be reading while a device is in the sleep state leading to missing data for that time period. The missed data cannot be easily replaced by using traditional approaches (e.g., use a daily average to fill in the missing values). Instead, an analyst who is monitoring a particular device would verify if the absence of data coincides with the state of the device and may classify the event as a false positive. When machine learning algorithms are used to automate this process, the learning models need to incorporate any network traffic behavior that is specific to a device's state (e.g., when a device is sleeping it transmits 80% less network packets [23]).

Connectivity

Traditional networked devices often use a wired network connection. This is the case for devices used in industrial control environments. However, wireless technologies such as 802.11-based networks, Zigbee, and others are increasingly being deployed in these industrial control systems. Many IoT devices are currently using Zigbee, which has low power consumption (therefore extending battery life) [24,25]. However, these protocols tend to be much slower in terms of their throughput (e.g., 250 Kbit/s) and

have a shorter range. Zigbee communications are also affected by interference caused by 802.11 communications, which leads to an increase in packet loss[26].

Limited hardware resources

In an effort to reduce the cost of IoT devices and their power consumption, these devices tend to have limited computing resources. Memory is by far the least available resource in some IoT devices (some with as low as a few Kilobytes). CPU limitations also limit the availability of services that can operate on such devices. All these hardware limitations have direct implications on the type of machine learning approaches that can be deployed to secure these IoT devices.

Heterogeneity

The wide range of IoT configurations is another limiting characteristic that may often require customized solutions for each IoT implementation. For example, many devices use lightweight protocols such as the Message Queueing Telemetry Transport (MQTT) which is a publish-subscribe messaging transport protocol[27]. MQTT delivers higher throughput compared to HTTP and requires substantially less resources. However, in order to use MQTT, we need a designated broker device that acts as a collector from the rest of the IoT devices. Users access the broker device (instead of the IoT devices) in order to retrieve sensor information.

3.2 | Common Attacks on IoT devices

In order to better understand the benefits resulting from the different machine learning defense implementations, we briefly describe some common attacks on IoT devices below. A more extensive discussion of attacks on IoT devices is beyond the scope of this paper but can be found in the recent literature[28,29,30,31]. Several well-known attacks have been described in the literature.

DOS attacks either disable the IoT device itself or use a compromised IoT device in order to launch other DoS attacks[32]. The first approach aims to overwhelm the lightweight communication channel that is utilized. The second approach takes advantage of weak security and maintenance (e.g., default authorization credentials set by the vendor of the IoT device are not changed or the owner of the IoT device fails to regularly update the device's firmware and software) that often exists in many of these devices. DOS attacks can also involve crafting bad packets with the goal of disabling the device. For example, a designated field in a packet that defines the payload length may be replaced with an incorrect value in order to cause a buffer overflow in a device's system or exhaust resources on the system to handle that particular exception. In addition to artificially manufacturing faulty packets, an attacker can also replay past packets in order to disrupt network communications or the operation of the device itself. A spoofing attack involves impersonating an existing device in the IoT network in order to access the communication channel or otherwise set the stage for a future man-in-the-middle attack. Man-in-the-middle attack enables an attacker to eavesdrop but also alter information transmitted over a communication channel. Typically, a successful attack in this category will allow the attacker to impersonate the IoT gateway which connects multiple IoT devices[28]. An eavesdropping attack aims to acquire information from IoT channels. Since surveillance is often the first step for most attacks, easily accessible communication channels are often the prime target of attackers. For IoT communications, often the data exchanged locally or remotely is not encrypted either because of a lack of options or negligence on the part of system administrators. Other types of attacks involve the use of trojans, worms as well as viruses in order to compromise, control and collect information from IoT devices have also been reported[28,33]. For example, an attacker may use a remote execution service's vulnerability on an IoT device in order to install a trojan that will provide him or her access to the device.

4 | MACHINE LEARNING: AN OPPORTUNITY FOR IMPROVING IOT SECURITY

As we mentioned earlier, the number of connected devices in various IoT environments (e.g., smart homes, healthcare, critical infrastructures) keeps increasing. Consequently, we need efficient security controls that can be automated and quickly analyze and differentiate between benign and malicious IoT data. Recently, several research efforts[28,34,35,36,37] have started to apply machine learning techniques to detect malicious traffic data and behaviors of devices running in IoT environments. Several drivers are responsible for this surge in interest in machine learning techniques aimed at improving IoT security. These include[13]: a) the resource-constrained nature of IoT devices in terms of limited storage, processing power and network connectivity makes

these devices less complex when compared with general computing and networked devices. As a result, automating the analysis of IoT data such that it is possible to differentiate between malicious and benign behaviors becomes feasible with machine learning; b) Deploying machine learning techniques in traditional computing systems and networks is more challenging than when deploying them in IoT environments. This is because of the frequent variations (such as bandwidth, duration, delay, and so on) in both the traffic data and the behaviors of traditional network systems and devices. Thus, it becomes more difficult to establish baseline for "normal" traffic in traditional network settings. In contrast, IoT devices are generally designed to execute specific and other repetitive functions which remain the same during the lifetime of the IoT device. This more predictable behavior makes it easier to develop a machine learning model that uses IoT traffic data to characterize benign (i.e., normal) behavior and distinguish it from malicious behavior.

The impact on performance of a machine learning model depends on both the machine learning algorithm as well as the context and characteristics of the data. In order to develop a machine learning model, we need good training data. We can collect both data that characterizes normal behavior and data that characterizes malicious behavior. Data from the latter behavior is more challenging to collect. This explains why most security solutions make use of unsupervised or one-class machine learning techniques which focus on normal device behavior characteristics. One-class learning involves distinguishing one class of target data from other types of data by using a classifier with a training set that only contains the target data. In the case of IoT devices, with one-class modeling we develop and train a model by using data that represents the normal behavior of an IoT device. After training the model, it is deployed in an actual IoT environment wherein it will try to detect any data that is different (which will be categorized as malicious) from the normal behavior of the IoT device.

Machine learning-based security solutions are viable alternatives to traditional IoT security that focuses largely on access control and encryption. However, from a threat-centric defense perspective, such defenses are bound to fail and as such an additional layer of machine learning-based security can prove beneficial in enhancing security. Machine learning applications are largely divided in three main categories: supervised, unsupervised and reinforcement learning [38]. However, given the availability of network data, these solutions can be further divided in network-based and host-based. This is similar to traditional network security monitoring solutions that involve network as well as host intrusion detection systems.

Figure 1 illustrates a simple example that demonstrates the deployment of machine learning techniques at the host and network levels in an IoT environment. Table 1 presents a summary of the machine learning algorithms and their potential for being implemented at the network-level or host-level as well as some challenges related to IoT characteristics. In the next few sections, we discuss appropriate machine learning techniques that can be deployed at the host-level and at the network-level. We focus our discussions based on the impact (which includes computational limitations, communication loss/latency tolerance, missing data tolerance and dynamic state utilization) of the relative characteristics of typical IoT devices. We also identify the strength and weaknesses of these machine learning techniques based on their deployment location.

TABLE 1 Summary of machine learning techniques based on the impact of characteristics of IoT devices.

		Representative algorithms	Computational limitations	Communication loss/latency tolerance	Missing data tolerance	Dynamic state utilization
Network-based	Supervised	CNN, ADA	Low	Medium	High	-
	Unsupervised	DBSCAN	Low	Medium	High	-
	Reinforcement	SARSA, DQN	Medium	High	High	-
Host-based	Supervised	k-NN, SVM	High	High	Low	Available
	Unsupervised	k-Means, GGMs	High	High	Low	Available
	Reinforcement	Q-Learning	High	High	High	Available

Note: CNN:Convolutional Neural Networks; ADA:ADAptive boosting; DBSCAN: Density-Based Spatial Clustering of Applications with Noise; SARSA:State–Action–Reward–State–Action; DQN:Deep Q-network; k-NN: k-Nearest Neighbors; SVM:Support Vector Machines; GGM:Gaussian Graphical Model.

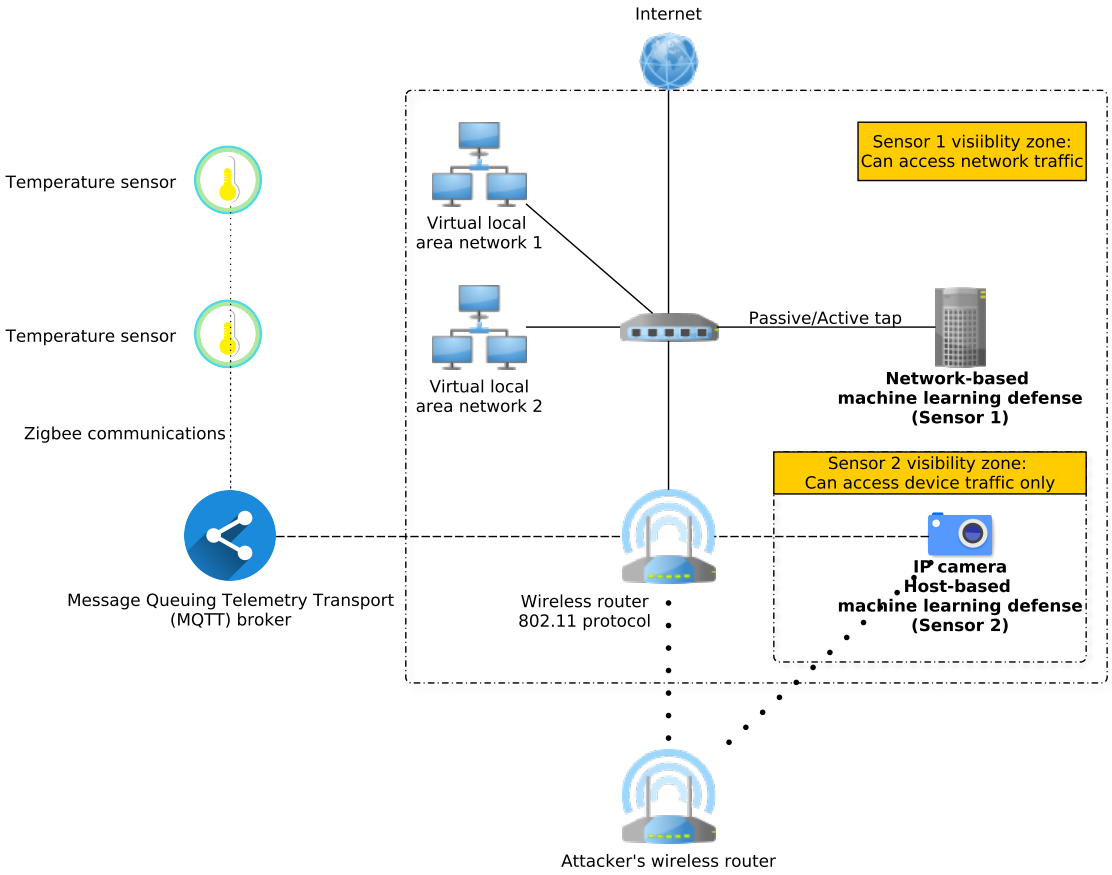

FIGURE 1 An example of a network that includes several IoT devices that utilize different communication protocols. MQTT is a protocol used for messaging between IoT devices. Host and network-based machine learning defenses are shown. The limitations for monitoring certain communications (e.g., temperature sensors to broker) unless further host-based solutions are implemented are also presented.

5 | NETWORK-BASED MACHINE LEARNING DEFENSE

Machine learning solutions in this category enable a variety of approaches to be implemented without any restrictions on computational resources. However, the extent to which network communications are encrypted may also limit the ability of network-based approaches to detect security incidents. For example, encrypted MQTT packets will not be readable by these network-based models. Instead, the machine learning model will have to rely on the quality of visible metadata (e.g., IPs, ports) of the network traffic in order to appropriately classify it.

5.1 | Supervised Learning

Supervised models focus on identifying known patterns that characterize benign and suspicious behavior. From a network perspective one can identify various types of attacks that will output network traffic (e.g., DOS, jamming). Supervised models require configuration ahead of time in order to be able to identify appropriate data points (i.e., features). As we mentioned earlier, IoT devices exhibit fairly routine, simplistic behavior due to the services they support as well as their single-purpose function. As such, a limited amount of useful data may exist for supervised models, which can potentially exclude the need to use algorithms (e.g., deep learning) that are more geared towards highly-dimensional datasets.

Resource requirements

For network-based machine learning, options such as Support Vector Machines (SVM) or k-Nearest Neighbors (kNN) can yield positive results as when they are used in traditional network security monitoring[39,40]. Furthermore, methods such SVM can also be computationally efficient because the same model (once trained) can be used for identifying malicious traffic from an adversary. In contrast, prediction in kNN requires the algorithm to compare each new data point (e.g., packet) to a list of other existing data points (the training set). In other words, there is no training phase that produces a vector of weights and the algorithm needs to be recomputed against a training set every time. Put simply, model-based methods such as SVM, regression or decision trees are more portable across different networks as long as the IoT devices are of the same vendor/type (i.e., their network behaviors are similar across networks).

Missing data tolerance

IoT devices can experience network connectivity issues as well as data measurement errors which will affect the way network traffic is being monitored. As result, it becomes difficult to differentiate between normal network behavior and malicious network behavior which can also confuse supervised learning models. For example, where data noise (measurement error) is consistent, ensemble models (e.g., Random Forest) can be used to average the error through the classification process. However, large missing data sequences due to connectivity issues cannot be efficiently handled by supervised models. For example, missing temperature readings by an IoT device for an hour could be the result of a connectivity failure or a compromise. Such missing data needs to filled in when a machine learning model is trained, which leads to many challenges (e.g., should the mean be used for the missing values?)[41]. The other challenge with supervised models is that, when some data points are missing, a prediction cannot be made (unless a model is especially trained to understand what missing data means). In other words, a model cannot easily infer if the missing data traffic is normal or abnormal.

Limited state information

IoT devices contain state information (e.g., an IoT will not advertise when it is going into a sleep state) that network-based supervised models do not normally have access to. As such, although hierarchical supervised learning models exist (e.g., decision trees), they cannot be efficiently used in order to learn different IoT behaviors under different states. This is not a serious limitation of these supervised models but it means that they cannot be used to their full potential.

5.2 | Unsupervised Learning

Unsupervised learning focuses on categorizing (clustering) of data to distinguish malicious activity from normal traffic[42,43]. As a result, no ground truth is required for the input (i.e., we do not need to tell the model what malicious or benign traffic looks like). However, this means that a human needs to interpret the output and determine what malicious traffic looks like. As a result, methods such as hierarchical or k-means clustering[44] require more human involvement even though they are both unsupervised learning algorithms. However, these methods have the potential to detect zero-day attacks because unknown patterns may be clustered separately (depending on how machine learning features that are fed into the model are organized) in the network traffic being monitored.

Resource requirements

Unsupervised learning algorithms tend to be more computationally intensive, which can hinder their implementation when the speed of data (data velocity) that is being parsed by these algorithms increases. This can occur when there are many IoT devices in a network. Given that there is no training phase, the classification of traffic data is completed using a set of dynamic data (e.g., new daily traffic pattern data). In addition, although NP-hard, many of these algorithms (e.g., k-means) have a computation time complexity that is practically linear. As a result, many of these unsupervised learning solutions are fairly lightweight to implement and scale well too. Therefore, the computational requirements of many unsupervised learning algorithms are rather low, making them an attractive solution for IoT machine learning defense.

Missing data tolerance

When the data generated by an IoT device is missing because of connectivity issues or the IoT device has been compromised, it becomes difficult to use many unsupervised algorithms. For example, k-means clustering requires a complete matrix which contains no missing values (e.g., null entries). However, techniques such as k-POD (a method for k-means clustering of missing data[45]) can overcome this restriction. Moreover, traditional unsupervised learning models make explicit assumptions about the normality of the data as well as a uniform variance that is expected to exist between groups. Noisy data that may be produced by IoT devices as a result of measurement errors from sensors will affect the effectiveness of unsupervised learning algorithms. In cases where such noisy data exists, it is necessary to use more robust methods such as Gaussian Mixture Models (GMMs)[46] or hierarchical Density-Based Spatial Clustering of Applications with Noise (DBSCAN)[47]. Some machine learning models do not perform well with noisy data compared to others such as ensemble methods[48]. Ensemble methods aggregate prediction results through multiple models (e.g, decision trees) and then average the differences among these multiple models. For example, if the ensemble model consists of five decision trees and three of them classify the traffic as malicious then the ensemble's outcome would also classify the traffic as malicious. Thus, in order to maintain a higher accuracy in detecting malicious traffic, we need to make sure that we select the appropriate machine learning algorithm if the data is noisy.

Limited state information

Similar to supervised learning, state information needs to be communicated via the network otherwise it cannot be incorporated in such models. If state information is available, hierarchical clustering algorithms (such as Ward's method[49]) can also be used.

5.3 | Reinforcement Learning

Reinforcement learning focuses on establishing environmental conditions (states) and action space for machines. The algorithms in this category enable the execution of multiple scenarios based on action and state permutations that lead to variable outcomes. As such, these algorithms adopt "winning" strategies which depend on the environmental state. The implementations of such algorithms at the network-level are expensive because multiple environmental states and actions need to be defined for the whole network. For example, a man-in-the-middle attack may appear at various places on a network and communicate with an arbitrary set of IoT devices. In this case, although reinforcement learning models do not scale well, they have been successfully implemented in network security monitoring[50,28,51]. Thus, adapting such methods for improving IoT security at the network level is possible.

Resource requirements

The training of reinforcement learning models is time consuming but once developed these models are portable to other networks (assuming network traffic behaviors are similar). The final models have low memory requirements.

Missing data tolerance

Reinforcement learning models tend to be more robust in terms of missing data and measurement errors in the data. Since an arbitrary number of scenarios can be executed when training the model, the outcomes obtained have a set of decision probabilities for any particular action. As such, scenarios can include previously unseen IoT behaviors and the model is guaranteed to make a decision (even if it involves an inaccurate classification) for a previously unknown specific device behavior.

Limited and dynamic state information

Since the state of a particular IoT device needs to be incorporated in the reinforcement learning model, any lack of data during training or when implementing reinforcement learning will affect the accuracy of detecting malicious network activity. This is especially true because when these models are trained, the IoT states and behaviors are all simulated. In a real scenario, tracking IoT states at the network-level can be challenging even when these states are known. For example, when an IoT device communicates that it is entering a sleep state over the network and the latter is experiencing congestion, it may cause the reinforcement learning model to interpret this lack of traffic as suspicious because the sleep state is not communicated on time. Given the low throughput for some of the communication protocols that IoT devices use (e.g., Zigbee), it may be difficult to implement reinforcement learning at the network-level. Furthermore, reinforcement learning models are also limited by the size

of the state space as well as how frequently the space changes. For example, for most real-world scenarios, it becomes difficult to guarantee that an reinforcement learning model will make the optimum decisions as learning becomes expensive. As such, reinforcement learning is currently limited to simple attack "scenarios" for large multi-state IoT networks.

6 | HOST-BASED MACHINE LEARNING DEFENSE

The implementation of machine learning defense algorithms on IoT devices enables the detection of attacks that sometimes are hard to identify from network traffic alone. Next, we review the three categories of machine learning algorithms when deployed at the host for IoT device security.

6.1 | Supervised learning

Algorithms such as SVM and k-NN have been used to identify malicious incidents with IoT devices[52,28] because they can help classify traffic and effectively detect intrusions, spoofing attacks as well as malware. A host-based approach involves incorporating these machine learning algorithms to identify specific interactions of the host with other networked devices (which may launch possible attacks). For example, k-NN has been used for outlier detection in wireless sensor networks[53]. The approach is successful in detecting specific communications and it incurs low power consumption. The classification of suspicious traffic is shared with other IoT devices. Other supervised algorithms such as Random Forests (ensemble methods that average noise across multiple predictive models)[17] have been recommended for detecting malware or for preventing spoofing[28].

Resource requirements

The ability to effectively implement supervised learning techniques is hindered by the limited computational resources of IoT devices. For example, while methods such as k-NN are effective, their computation complexity is proportional to the amount of training data (e.g., number of packets in the training set). However, as there is no training stage for some machine learning algorithms (such as k-NN), they are executed with the same complexity every time which can quickly exhaust the limited computing and storage resources of the IoT device. Other supervised algorithms (such as SVM) that allow for pre-training of models can be computationally efficient but depending on the algorithm and the amount of training data, they may exhaust the available memory on a device. For example, a Random Forest model (i.e., an ensemble algorithm) may consist of hundreds of decision trees all of which have to be loaded into memory in order to quickly classify and identify malicious activities.

Measurement error tolerance

Host-based machine learning defense solutions can ignore any network connectivity issues that may exist because their goal is to correctly classify traffic that reaches the IoT device. As such, the biggest challenge for host-based security solutions is data measurement errors, that may exist which are undetected and are used in machine learning models. For example, an IoT system may be programmed to enter into a temporary sleep state if ambient temperatures reach a certain threshold. The device that detects such a condition may signal to other adjacent devices to do the same (regardless of their local temperature readings) in a scenario where these devices are linked together. Similarly, an adversary can exploit this behavior by signaling other IoT devices to temporarily enter their sleep states (i.e., launching a DOS attack on these devices). A machine learning model can cross-reference past data with the current temperature readings to identify any drastic variations (inconsistencies). However, measurement errors may affect the accuracy of the supervised model. Therefore, supervised learning models that can efficiently handle more noisy data (e.g., Random Forest) are more likely to yield better results in these scenarios.

Limited network view

Although host-based defense using supervised learning is expected to be effective for targeted IoT attacks, the limited view of the network by the IoT device means that not all attacks can be detected by the host-based security solution in place. For example, a distributed denial of service attack from various IoT devices to a switch or router is not detectable by other IoT devices (although they may experience an increase in latency in their network communications).

6.2 | Unsupervised Learning

Techniques that employ unsupervised learning can be highly effective when they are implemented as host-based solutions. For example, the Infinite Gaussian Mixture Model has been used to evaluate the authentication of IoT devices that are close to one another[54]. In this work, authentication between two devices is achieved based on a series of features such as signal strength, sequence numbers as well as Media Access Control (MAC) addresses of wireless packets observed. This approach, based on the Infinite Gaussian Mixture Model, is highly effective against eavesdropping attacks because a client does not have to disclose its location information (i.e., protecting the position of an IoT device) to attackers.

Resource requirements

Unsupervised learning methods can be computationally expensive because they have to classify incoming data on the fly (there is practically no training phase where a model can be prepared ahead of time). As a result, the amount of data that these techniques will be applied to directly depend on the computational capability of an IoT device to execute them. Furthermore, the frequency (e.g., every minute) that an algorithm needs to be executed will also affect the feasibility of implementation of the unsupervised learning methods. These computation and implementation constraints can be addressed by reducing the amount of data used as well as the frequency of the algorithm's execution. But unsupervised models can lead to false negatives because of under-sampling or oversampling[28].

Measurement error tolerance

The selection of features used by unsupervised learning models to detect malicious traffic can have a significant impact on the model's accuracy CITATION. As such, similar to supervised learning, if measurement errors exist because of the low quality of sensor data or signal interference, the errors will affect the accuracy of detecting malicious events for unsupervised algorithms.

Limited network view

The inability to observe various states in the network may influence the efficiency with which various traffic patterns are categorized by the IoT device. However, for a process targeting outlier detection this may not substantially degrade the process of identifying malicious events but it will increase the "false positive" rate in the sense that a security analyst will have a more difficult time discerning whether such an incident is happening.

6.3 | Reinforcement Learning

Host-based solutions that utilize reinforcement learning mainly involve the use of model-free reinforcement learning algorithms which do not incorporate the environmental state that the algorithm executes in and instead focus on optimizing the reward for a series of actions that need to be taken based on the state of an agent. The advantage of such a reinforcement learning model is that it can be easily adapted to multiple network or host settings and in the IoT environment it has been successfully used for authentication, anti-jamming offloading and malware identification[28].

Resource requirements

For some reinforcement learning algorithms, the computation time complexity for optimization can be as high as $O(en)$, where n is the state space and e is the total number of actions[55]. Furthermore, the number of actions always tends to be higher than the state space. However, if the problem space does not include duplicate actions or there is a linear upper bound on the number of actions, the computation complexity of the reinforcement learning algorithm (e.g., Q-learning) can be as low as $O(n^2)$[55], which substantially reduces the computational overhead. However, reinforcement learning models that involve deep learning may be impossible to implement directly on some IoT devices especially when the feature space increases (e.g., image, audio or embedded radar data analysis).

Missing data tolerance

Similar to network-based solutions, reinforcement learning models are more tolerant to measurement errors that may arise from properties that influence the state and sensors of the device. When local state information is used, reinforcement learning models

implemented at the host-level may even better contextualize the conditions that lead to measurement errors and artificially correct them. For example, a device may learn that the standard deviation of data being polled from a sensor may be larger in some specific state during which the polling rate is slower.

Limited and dynamic state information

The implementation of reinforcement learning models as host-based solutions means that the device may not be aware about the states of other devices (or the device may receive a delayed signal about the states of other devices) but it is completely aware about its own state at any given moment which is an important condition for producing useful and accurate reinforcement learning models. In reinforcement learning, a state would represent the state of multiple aspects (such as active or sleeping, number of bytes sent, and so on) associated with a device. There is still a concern with distributed states across associated IoT devices that makes state space search expensive. However, for single devices the impact of this factor is limited. Successful host-based implementations of reinforcement learning techniques have demonstrated their effectiveness on securing IoT devices[56,51].

Limited network view

The lack of a complete picture of the entire network makes it inherently impossible to detect scenarios that involve multi-stage attacks (similar to other machine learning methods). However, reinforcement learning algorithms can be further influenced to exhibit poor performance under certain conditions such as context insensitivity and reward sparsity[57]. Context insensitivity refers to the ineffective implementation of defining a reward that does not effectively capture context. For example, instead of rewarding the algorithm for identifying bad traffic, we can provide a reward when the algorithm identifies clusters of malicious traffic, which is often the case when we have a sequence of attack events. Similarly, when reinforcement learning algorithms are unaware of their progress for a long time (reward sparsity), this can lead to a random search of the state space and less accurate detection of malicious traffic. There is a lack of full context awareness with host-based reinforcement learning solutions. This means that even if they can be provided with more context-aware information (accuracy of detecting malicious traffic), the availability of contextual data (i.e., what is happening in the overall network) does not exist.

7 | CHALLENGES AND OPPORTUNITIES

There are strong opportunities for machine learning approaches to improve ioT security but we still need to address several key challenges associated with machine learning techniques. These challenges include adversarial machine learning, algorithm portability, and further research into computationally cheaper machine learning algorithms as well as robustness against eavesdropping attacks.

7.1 | Adversarial machine learning

All machine learning algorithms are susceptible to adversarial attacks regardless of the type of machine learning algorithm that is used. In general, the more asymmetry there is between a defender and an attacker, the more protected a machine learning model is from adversarial attacks[58?] . However, all models are prone to causative as well as exploratory attacks[59]. The former refers to the ability of an attacker to influence the model by either influencing the training or testing data. The latter refers to an attacker who performs reverse engineering on a model's detection baseline and then uses that knowledge in order to avoid detection. It is possible to also launch both attacks simultaneously. Overall, machine learning models that often require re-training or the use of recent data are more prone to these types of attacks. For example, it is well-known that small changes (e.g., by adding small a small amount of noise while keeping the data indistinguishable from the original data) in the input data used by the learning model can lead to quite different results. More efforts are needed on adversarial machine learning research in order to detect adversarial and evasion attacks. Machine learning systems are vulnerable to poisoning attacks wherein an attack can insert malicious data into the dataset that is used to train the learning algorithm. In this case, the learning process is severely affected. Consequently, the learning algorithm is taught a bad model. It is worth mentioning that poisoning attacks can be used as a first step before launching evasion attacks later[13].

Machine learning also have vulnerabilities which can be exploited by an attacker. In particular, an attacker can exploit weaknesses of the underlying learning algorithms. For instance, during an evasion attack, an attacker creates malicious data that

intentionally produce errors in the machine learning system. Such attacks are possible because of vulnerabilities that arise: a) when the initial data filtering or the detection of outliers are ineffective. In this case, the attacker can inject data that significantly differs from the data expected resulting in unpredictable behavior of the learning algorithm, b) when the data used to train the learning algorithm is not always sufficient to achieve a perfect (ideal) underlying learning model for the task it will be subjected to[13].

While the development of more robust machine learning algorithms is necessary, we recommend that priority be given to addressing the development of "better" models by IoT security experts in the future. In general, models that require less re-training in practice are those that use more complex features and as such are inherently less prone to both causative as well exploratory attacks.

Further, the development of robust, secure machine learning systems that can withstand sophisticated attacks of smart adversaries remains a significant challenge. By investing more research efforts to address this challenge, we can pave the way for the deployment of machine learning technologies which can strengthen security in areas such as the Internet of Things and others.

7.2 | Algorithm portability

The current state of machine learning algorithms limits the ability for them to be portable across various IoT implementations. Although algorithm portability is not an absolute requirement (in fact adversaries may benefit from a standardized defense model), it would be useful to establish standard features that are known to be successful in various IoT machine learning security implementations. The lack of standardized tools and libraries for machine learning means that the adoption of such methods by IoT device manufacturers as well as security engineers will be limited.

7.3 | Eavesdropping detection

Detecting an attacker who is eavesdropping on communications is still a difficult task for IoT devices. The more passive the actions of an attacker are, the more difficult it becomes to detect them. Most of the current security measures focus on passive measures that include layers of encryption on the communication channel. However, an attacker can still access such an encrypted channel by spoofing one of the communication devices. Some progress has been made in using sensors to identify a potential eavesdropper in the physical space[28]. Give the seamless and ubiquitous connectivity of the IoT devices with physical space solutions against eavesdropping are likely to be effective if machine learning approaches are implemented.

8 | CONCLUSION

Internet of Things devices are being been extensively integrated an deployed within organizations following the promise that more data will lead to better decision-making. However, such devices are rarely implemented with security concerns in mind. Furthermore, even when such measures are implemented, there are distinct limitations on the rules and signatures that can be used to identify potential adversaries. In this paper, we have explored how we could improve IoT security at the network-level and at the host-level with machine learning techniques. We have also highlighted the strength and limitations of the machine learning algorithms because of the unique characteristics of the IoT devices and their environments. Given the explosive growth and increasing adoption of IoT devices, security researchers need to develop novel, cost-effective, scalable machine learning techniques as well as adapting existing ones to better meet the computational and environmental constraints of the IoT ecosystem.

References

1. Bello O, Zeadally S. Toward efficient smartification of the Internet of Things (IoT) services. *Future Generation Computer Systems* 2019; 92: 663–673. doi: 10.1016/j.future.2017.09.083

2. National Broadband Network . On track for over 30 connected devices per Aussie household by 2021. 2017. https://www.nbnco.com.au/blog/connected-homes/on-track-for-over-30-iot-devices-per-aussie-household-by-2021. Date Accessed: 2019-05-17.

3. Park Associates . Consumers own more than 10 connected devices. 2019. http://www.parksassociates.com/blog/article/pr-03272019. Date accessed: 2019-05-17.

4. Tsai CF, Hsu YF, Lin CY, Lin WY. Intrusion detection by machine learning: A review. *Expert Systems with Applications* 2009; 36(10): 11994–12000. doi: 10.1016/j.eswa.2009.05.029

5. Sommer R, Paxson V. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In: Proceedings of the IEEE Symposium on Security and Privacy (S&P '10). IEEE; 2010: 305–316

6. Mclay F. Privacy law: How to save face: Data and privacy safeguards. *Governance Directions* 2018; 70(4): 202–206.

7. Kolias C, Kambourakis G, Stavrou A, Voas J. DDoS in the IoT: Mirai and Other Botnets. *Computer* 2017; 50(7): 80–84. doi: 10.1109/MC.2017.201

8. Habibi J, Midi D, Mudgerikar A, Bertino E. Heimdall: Mitigating the Internet of Insecure Things. *IEEE Internet of Things Journal* 2017; 4(4): 968–978. doi: 10.1109/JIOT.2017.2704093

9. Bertino E, Islam N. Botnets and Internet of Things Security. *Computer* 2017; 50(2): 76–79. doi: 10.1109/MC.2017.62

10. Gardner MT, Beard C, Medhi D. Using SEIRS Epidemic Models for IoT Botnets Attacks. In: Proceedings of the International Conference of Design of Reliable Communication Networks (DRCN '17). ; 2017: 1–8.

11. Fong K, Hepler K, Raghavan R, Rowland P. rIoT: Quantifying Consumer Costs of Insecure Internet of Things Devices. tech. rep., University of California, Berkeley, School of Information; 2018.

12. Help Net Security . The cost of IoT hacks: Up to 13% of revenue for smaller firms. 2017.

13. Munoz-Gonzalez L, Lupu EC. The Secret of Machine Learning. *ITNOW* 2018; 60(1): 38–39. doi: 10.1093/itnow/bwy018

14. Witten IH, Frank E, Hall MA. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers Inc. . 2011.

15. Chio C, Freeman D. *Machine Learning and Security*. O'Reilly Media, Inc. . 2018.

16. Shrestha A, Mahmood A. Review of Deep Learning Algorithms and Architectures. *IEEE Access* 2019; 7: 53040–53065. doi: 10.1109/ACCESS.2019.2912200

17. Breiman L. Random Forests. *Machine Learning* 2001; 45(1): 5–32. doi: 10.1023/A:1010933404324

18. Lloyd S. Least squares quantization in PCM. *IEEE Transactions on Information Theory* 1982; 28(2): 129–137. doi: 10.1109/TIT.1982.1056489

19. Ester M, Kriegel HP, Sander J, Xu X. A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (SIGKDD '96). AAAI Press; 1996: 226–231.

20. Watkins CJ, Dayan P. Q-Learning. *Machine Learning* 1992; 8: 279–292.

21. Patel KK, Patel SM. Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges. *International Journal of Engineering Science and Computing* 2016; 6(5): 6122–6131. doi: 10.4010/2016.1482

22. Jadaliha M, Xu Y, Choi J, Johnson NS, Li W. Gaussian Process Regression for Sensor Networks Under Localization Uncertainty. *IEEE Transactions on Signal Processing* 2013; 61(2): 223–237. doi: 10.1109/TSP.2012.2223695

23. Google Cloud . Overview of Internet of Things. 2019. https://cloud.google.com/solutions/iot-overview. Date accessed: 2019-05-21.

24. Zhou Y, Yang X, Guo X, Zhou M, Wang L. A Design of Greenhouse Monitoring & Control System Based on ZigBee Wireless Sensor Network. In: Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '07). ; 2007: 2563–2567

25. Zheng L. ZigBee Wireless Sensor Network in Industrial Applications. In: Proceedings of the International Joint Conference on Society of Instrument and Control Engineers of Japan and International Conference on Applied Science and Engineering (SICE-ICASE '06). IEEE; 2006: 1067–1070

26. Zhao Z, Wu X, Zhang X, Zhao J, Li XY. ZigBee vs WiFi: Understanding issues and measuring performances of their coexistence. In: Proceedings of the IEEE International Performance Computing and Communications Conference (IPCCC '14). IEEE; 2014: 1–8

27. Hunkeler U, Truong HL, Stanford-Clark A. MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks. In: Proceedings of the International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08). ; 2008: 791–798

28. Xiao L, Wan X, Lu X, Zhang Y, Wu D. IoT Security Techniques Based on Machine Learning: How Do IoT Devices Use AI to Enhance Security?. *IEEE Signal Processing Magazine* 2018; 35(5): 41–49. doi: 10.1109/MSP.2018.2825478

29. Das AK, Zeadally S, He D. Taxonomy and analysis of security protocols for Internet of Things. *Future Generation Computer Systems* 2018; 89: 110–125. doi: 10.1016/j.future.2018.06.027

30. Nawir M, Amir A, Yaakob N, Lynn OB. Internet of Things (IoT): Taxonomy of security attacks. In: Proceedings of the International Conference on Electronic Design (ICED '16). ; 2016: 321–326

31. Deogirikar J, Vidhate A. Security attacks in IoT: A survey. In: Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC '17). ; 2017: 32–37

32. Andrea I, Chrysostomou C, Hadjichristofi G. Internet of Things: Security vulnerabilities and challenges. In: Proceedings of the IEEE Symposium on Computers and Communication (ISCC '15). IEEE; 2015: 180–187

33. Kambourakis G, Kolias C, Stavrou A. The Mirai botnet and the IoT Zombie Armies. In: Proceedings of the IEEE Military Communications Conference (MILCOM '17). ; 2017: 267–272

34. Shanthamallu US, Spanias A, Tepedelenlioglu C, Stanley M. A brief survey of machine learning methods and their sensor and IoT applications. In: Proceedings of the International Conference on Information, Intelligence, Systems & Applications (IISA '17). ; 2017: 1–8

35. Meidan Y, Bohadana M, Shabtai A, et al. ProfilIoT: A Machine Learning Approach for IoT Device Identification Based on Network Traffic Analysis. In: Proceedings of the Symposium on Applied Computing (SAC '17). ACM; 2017; New York, NY, USA: 506–509

36. Li H, Ota K, Dong M. Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing. *IEEE Network* 2018; 32(1): 96–101. doi: 10.1109/MNET.2018.1700202

37. Cañedo J, Skjellum A. Using machine learning to secure IoT systems. In: Proceedings of the Annual Conference on Privacy, Security and Trust (PST '16). ; 2016: 219–222

38. Maloof MA. , ed.*Machine Learning and Data Mining for Computer Security*. Advanced Information and Knowledge ProcessingLondon: Springer-Verlag . 2006

39. Mukkamala S, Janoski G, Sung A. Intrusion detection using neural networks and support vector machines. In: Proceedings of the International Joint Conference on Neural Networks. (IJCNN'02). ; 2002: 1702–1707

40. Liao Y, Vemuri V. Use of K-Nearest Neighbor classifier for intrusion detection. *Computers & Security* 2002; 21(5): 439–448. doi: https://doi.org/10.1016/S0167-4048(02)00514-X

41. Pelckmans K, De Brabanter J, Suykens J, De Moor B. Handling missing values in support vector machine classifiers. *Neural Networks* 2005; 18(5-6): 684–692. doi: 10.1016/j.neunet.2005.06.025

42. Guan Y, Ghorbani AA, Belacel N. Y-means: a clustering method for intrusion detection. In: Proceedings of the Canadian Conference on Electrical and Computer Engineering. Toward a Caring and Humane Technology (CCECE '03). ; 2003: 1083–1086

43. Chen Z, Li YF. Anomaly Detection Based on Enhanced DBScan Algorithm. *Procedia Engineering* 2011; 15: 178–182. doi: https://doi.org/10.1016/j.proeng.2011.08.036

44. Jianliang M, Haikun S, Ling B. The Application on Intrusion Detection Based on K-means Cluster Algorithm. In: Proceedings of the International Forum on Information Technology and Applications (ICITA '09). ; 2009: 150–152

45. Chi JT, Chi EC, Baraniuk RG. k-POD: A Method for k-Means Clustering of Missing Data. *The American Statistician* 2016; 70(1): 91–99. doi: 10.1080/00031305.2015.1086685

46. Reynolds DA, Quatieri TF, Dunn RB. Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing* 2000; 10(1-3): 19–41. doi: 10.1006/dspr.1999.0361

47. Schubert E, Sander J, Ester M, Kriegel HP, Xu X. DBSCAN Revisited, Revisited. *ACM Transactions on Database Systems* 2017; 42(3): 1–21. doi: 10.1145/3068335

48. Dietterich TG. An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. *Machine Learning* 2000; 40(2): 139–157. doi: 10.1023/A:1007607513941

49. Szekely GJ, Rizzo ML. Hierarchical Clustering via Joint Between-Within Distances: Extending Ward's Minimum Variance Method. *Journal of Classification* 2005; 22(2): 151–183. doi: 10.1007/s00357-005-0012-9

50. Chung K, Kamhoua CA, Kwiat KA, Kalbarczyk ZT, Iyer RK. Game Theory with Learning for Cyber Security Monitoring. In: Proceedings of the IEEE International Symposium on High Assurance Systems Engineering (HASE '16). IEEE; 2016: 1–8

51. Xiao L, Xie C, Chen T, Dai H, Poor HV. A Mobile Offloading Game Against Smart Attacks. *IEEE Access* 2016; 4: 2281–2291. doi: 10.1109/ACCESS.2016.2565198

52. Alsheikh MA, Lin S, Niyato D, Tan H. Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications. *IEEE Communications Surveys & Tutorials* 2014; 16(4): 1996–2018. doi: 10.1109/COMST.2014.2320099

53. Branch J, Szymanski B, Giannella C, Ran Wolff , Kargupta H. In-Network Outlier Detection in Wireless Sensor Networks. In: Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS '06). IEEE; 2006: 51–51

54. Xiao L, Yan Q, Lou W, Chen G, Hou YT. Proximity-Based Security Techniques for Mobile Users in Wireless Networks. *IEEE Transactions on Information Forensics and Security* 2013; 8(12): 2089–2100. doi: 10.1109/TIFS.2013.2286269

55. Koenig S, Simmons R. Complexity Analysis of Real-Time Reinforcement Learning. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI '93). ; 1993: 99–105.

56. Gwon Y, Dastangoo S, Fossa C, Kung HT. Competing Mobile Network Game: Embracing antijamming and jamming strategies with reinforcement learning. In: Proceedings of the IEEE Conference on Communications and Network Security (CNS '13). ; 2013: 28–36

57. Agarwal A, Hope R, Sycara KP. Challenges of Context and Time in Reinforcement Learning: Introducing Space Fortress as a Benchmark. *Computing Research Repository* 2018; abs/1809.0.

58. Huang L, Joseph AD, Nelson B, Rubinstein BI, Tygar JD. Adversarial machine learning. In: Proceedings of the ACM Workshop on Security and Artificial Intelligence (AISec '11). ACM Press; 2011; New York, New York, USA: 43

59. Huang L, Joseph AD, Nelson B, Rubinstein BIP, Tygar JD. Adversarial Machine Learning. In: Proceedings of the ACM Workshop on Security and Artificial Intelligence (AISec '11). ACM; 2011; New York, NY, USA: 43–58

AUTHOR BIOGRAPHY

Sherali Zeadally is an associate professor in the College of Communication and Information at the University of Kentucky. He received his bachelor degree in computer science from the University of Cambridge, England, and his doctoral degree in computer science from the University of Buckingham, England. His research interests include cybersecurity, privacy, Internet of Things, and energy-efficient networking. He is a Fellow of the British Computer Society and the Institution of Engineering Technology, England.

Michael Tsikerdekis is an assistant professor in the Computer Science Department at Western Washington University. His research interests include deception, data mining, cybersecurity, and social computing. Tsikerdekis has a PhD in Informatics from Masaryk University. Contact him at Michael.Tsikerdekis@wwu.edu.