Towards Learning-automation IoT Attack Detection through Reinforcement Learning

Tianbo Gu*, Allaukik Abhishek†, Hao Fu*, Huanle Zhang*, Debraj Basu *, Prasant Mohapatra*
* Department of Computer Science, University of California, Davis, CA, USA
† ARM Research, Austin, TX, USA
Email: {tbgu, haofu, dtczhang, dbasu, pmohapatra}@ucdavis.edu, Allaukik.Abhishek@arm.com

***Abstract*—As a massive number of the Internet of Things (IoT) devices are deployed, the security and privacy issues in IoT arouse more and more attention. The IoT attacks are causing tremendous loss to the IoT networks and even threatening human's safety. Compared to traditional networks, IoT networks have unique characteristics, which make the attack detection more challenging. First, the heterogeneity of platforms, protocols, software, and hardware exposes various vulnerabilities. Second, in addition to the traditional high-rate attacks, the low-rate attacks are also extensively used by IoT attackers to obfuscate the legitimate and malicious traffic. These low-rate attacks are difficult to detect and can persist in the networks. Last, the attackers are evolving to be more intelligent and can dynamically change their attack strategies based on the environment feedback to avoid being detected, which makes it more challenging for the defender to discover a consistent pattern to identify the attack.**

In order to adapt to the new characteristics in IoT attacks, we propose a reinforcement learning based attack detection model that can automatically learn and recognize the transformation of the attack pattern. Therefore, we can continuously detect IoT attacks with less human intervention. In this paper, we explore the crucial features of IoT traffics and utilize the entropy-based metrics to detect both the high-rate and low-rate IoT attacks. Afterward, we leverage the reinforcement learning technique to continuously adjust the attack detection threshold based on the detection feedback, which optimizes the detection and the false alarm rate. We conduct extensive experiments over a real IoT attack dataset and demonstrate the effectiveness of our IoT attack detection framework.

***Index Terms*—Internet of Things, Intrusion detection, Reinforcement learning, Anomaly detection, IoT security.**

I. Introduction

Based on the report from [1], there will be 75 billion Internet of Things (IoT) connected devices in the world by 2025. The growth of IoT devices has exploded over the past ten years. IoT techniques have been applied to a variety of fields, such as smart home, smart city, and smart industry. The IoT networks deployed dramatically enhance the productivity, quality, and efficiency of work and also creates substantial economic profits. For instance, an IoT application can automatically open the home window or the air conditioner once the indoor temperature exceeds a threshold, which makes our lives more automatic without human intervention.

The immense network comprising of IoT devices is interrelated and independent with the current Internet. But, the corresponding security and privacy solutions do not keep up with the increasing scale of IoT devices. The vast commercial market also earns the attention of attackers. The attackers attempt to explore and launch a variety of attacks in IoT networks, which may cause enormous economic loss and pose a severe threat to security and privacy. However, the unique characteristics in IoT networks make it more challenging to provide comprehensive security and privacy solutions.

First, in order to maintain a long life cycle and combat with the limitation of computations and battery capacity, most IoT devices have to be compromised with security and privacy issues. Some IoT devices adopt weak encryption or even no encryption process. Meanwhile, due to no consistent standard, there are many IoT platforms and communication protocol proposed in the IoT network. The common IoT platforms include Samsung SmartThings [2], Google Home [3], Apple HomeKit [4], etc. These platforms also use different wireless communication protocols, such as *WiFi, ZigBee, Z-Wave, BLE, etc.* However, the heterogeneity of platforms and communication protocols exposes various attack interfaces, thereby making it considerably challenging for a defender to safeguard against the attacks [5].

Second, most IoT devices transmit the data in a relatively low rate. In order to avoid being detected, the attackers launch the intrusion with a low-rate trait and produce malicious traffic that blends in the normal traffic, which is challenging to be detected with superior disguise. Furthermore, with the rapid development of *Artificial Intelligence* and *Machine Learning*, the attacker is gradually possessing the capabilities of learning-automation and becoming more smart. In order to delay the detection time, the attacker can study the feedback from the environment and dynamically alter their attack vectors or launch a new evolved attack to avoid intrusion detection. The detected attack pattern by the defender will be invalid immediately, and the defender has to make an instantaneous response and recognize the new attack pattern fast.

Currently, most existing detection approaches leverage the traditional anomaly detection model [6] without considering the unique characteristics in the IoT domains. Most detection methods do not consider the diversity of attack interfaces and can only detect very few attack types. Some researchers use deep learning [7]–[9] to implement the network intrusion detection. But the computation power and time cost of these approaches are too expensive to be executed on a simple IoT gateway. Also, the existing detection approaches are not sensitive to low-rate attacks. Moreover, these solutions cannot adapt to the learning-automation attacks in IoT [10],

[11]. The expensive detection process of attack patterns may be immediately invalid if the attackers continuously change their intrusion strategies. Therefore, we propose a generalized solution that can detect a variety of attacks by analyzing the features of IoT traffic. A lightweight detection model is designed using an entropy-based detection approach, which is more sensitive to low-rate attacks and has a low computation cost. More importantly, we propose a reinforcement learning based IoT attack detection framework that can detect the evolving IoT attacks with more effectiveness.

Contribution: In summary, our contributions break down into the following aspects:

- We thoroughly explore all the unique attack characteristics in IoT networks, such as the low-rate attack, and propose a lightweight entropy-based attack detection approach that can efficiently detect a variety of attacks.
- To resist the more intelligence of IoT attacks, we propose a reinforcement learning based attack detection framework in IoT networks, which can automatically adjust the attack detection threshold and fast discover the attacks even with the continuous transformation and evolutionary of IoT attacks.
- To the best of our knowledge, we are the first to incorporate reinforcement learning to develop a novel attack detection framework in IoT networks. Our approach can provide smarter defenses with less human supervision, which can greatly facilitate the current research in IoT security.
- We design and conduct extensive experiments on a ~ 60 GB IoT dataset involving a dozen IoT devices in the market and various types of IoT attacks. The evaluation demonstrates that our proposed IoT attack detection framework has th capability of adapting to the new attack characteristics and exhibit significant improvement of system utilities.

Road map: The rest of the paper is organized as follows. In Section II, we discuss the adversary model for IoT attack. Section III describes the IoT attack detection framework including the model and algorithm. We present our evaluation in Section IV. Related work and conclusion are presented in Section V and VI respectively.

II. Adversary model

In this work, a typical IoT network (such as smart home, smart industry, smart city) is considered, where the IoT devices are first connected to the IoT gateway and then connected to the Internet thereby forming a vast IoT network. The adversary scans the IoT devices and networks, and explores the existing vulnerability that IoT networks have exposed. The exposed attack interfaces could include the bugs in IoT platform, the weak encryption vulnerabilities in wired and wireless communication protocols, the defects of IoT devices' hardware and software, etc.

We divide the attack types into two categories: (1) direct attacks and (2) reflection attacks. The direct attacks to IoT include TCP/UDP flooding, ARP spoofing, Ping of Death, etc. The reflection attacks include TCP SYN, Smurf, SNMP, etc. Some attacks attempt to consume all the available resources in IoT devices by attacking the network protocols in different layers, rendering the services of the IoT system unavailable. Some other attacks can exploit the security vulnerabilities of the IoT networks and try to compromise other devices in the network or inject false information in the system.

Moreover, most attacks need to persist in the IoT network for an extended period in order to operate effectively, such as DDoS attack. To persist in the network, the attacks launched by the adversary usually have two properties: *stealth* and *resilience*. The adversary produces malicious traffic with no difference with legitimate traffic, thereby remaining elusive, such as low-rate attacks. On the other side, some attackers become more intelligent and can transform or evolve their attack procedures via observing the feedback from the environment, which could make it more challenging to identify the IoT attacks and sustain the resilience.

Our goal is to design an attack detection framework that considers the unique attack characteristics in IoT. The detection framework can identify IoT attacks with high superiority. For instance, the low-rate attacks sneak their malicious traffic into normal traffic, thereby avoiding detection. Furthermore, the defense strategies also need to be upgraded and evolved as the development of learning-automation for attackers. The adversary could transform their attack strategies or develop a new attack approach based on the feedback of the environment. The more intelligent and flexible attacks significantly increase the difficulties of defense. An efficient defensive scheme to mitigate attacks and secure IoT systems should be designed and implemented with considering the new attack characteristics in IoT.

III. IoT Attack Detection Framework

In this section, we mainly describe the components of our attack detect framework in IoT networks, which is shown in Figure 1. First, we revisit the information theory for anomaly detection and propose to use the entropy-based metrics to distinguish the benign and malicious traffic in IoT networks. Then, we investigate the performance of distinct features that are utilized for IoT attack detection. Due to the emergence of learning-automation attacks, as shown in Figure 2, the classification boundary between normal and malicious traffic may move when the attacker changes its attack parameters, such as attack rate and type. Therefore, we introduce a reinforcement learning model to our attack detection framework to continuously update the classification boundary, thereby well adapting to the new IoT attacks.

A. Information Metrics for IoT Attack Detection

Information metrics are adopted widely to discover the anomalies in intrusion detection fields [12], [13]. In information theory, entropy measures the randomness of an information variable. The value of entropy tends to be larger when the information variable shows to be more random. On the contrary, when the amount of uncertainty for the information

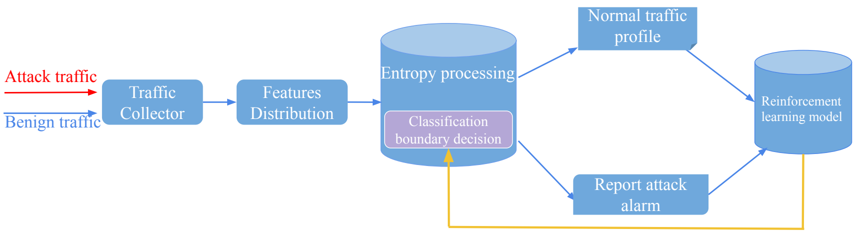


Fig. 1: Framework of IoT attack detection via reinforcement learning.

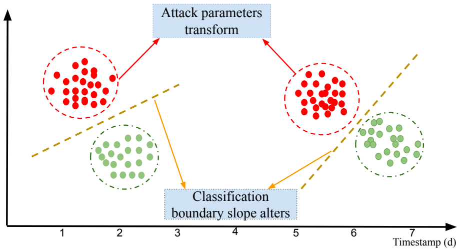

Attack parameters transform
Classification boundary slope alters
1
2
3
4
5
6
7
Timestamp (d)

Fig. 2: A scenario of mutative attack.

variable is small, the entropy will have a relatively low value. The low entropy value demonstrates the concentration of distribution for information variables, which could reveal that there may exist an anomaly in the current system.

In this work, we attempt to apply the entropy-based approach to IoT anomaly detection. Shannon and Reyi entropy are the two typical metrics adopted in the information theory to measure the dispersion and concentration for the feature variable. We consider a discrete probability distribution:

$$P(X) = \{p(x_1), p(x_2), ..., p(x_n)\} \sum_{i=1}^{n} p(x_i) = 1 \tag{1}$$

The variable X could be a feature that take the discrete values of $\{x_1, x_2, ..., x_n\}$. $p(x_i)$ represents the probability that X takes value x_i. Then the Renyi entropy of order α is defined as:

$$H_\alpha(X) = \frac{1}{1-\alpha} log_a \big(\sum_{i=1}^{n} p(x_i)^\alpha \big) \tag{2}$$

When $\alpha \rightarrow 1$, $H_\alpha(X)$ converges to Shannon entropy:

$$\begin{aligned} H_s(X) &= \lim_{\alpha \to 1} \big(\frac{1}{1-\alpha} log_a (\sum_{i=1}^{n} p(x_i)^\alpha) \big) \\ &= \sum_{i=1}^{n} p(x_i) log_a (\frac{1}{p(x_i)}) \end{aligned} \tag{3}$$

The Shannon entropy of X can also be regarded as the expected value of $log_a(\frac{1}{p(x_i)})$ with the probability mass function $p(x_i)$. The value of the entropy $H_s(X)$ reflects the randomness of the probability distribution and reveals that the feature has a more dispersed or concentrated property of probability distribution.

In addition to measure the randomness of feature variable, the extent of changes between assumed and observed distribution could also be measured for discovering some anomalies. Let us use two discrete probability distribution U and V for the feature variable X. The information divergence between distribution of U and V of order α could be denoted as:

$$D_\alpha(U||V) = \frac{1}{1-\alpha} log_a \big(\sum_{i=1}^{n} \frac{u(x_i)^\alpha}{v(x_i)^{1-\alpha}} \big) \tag{4}$$

When $\alpha \rightarrow 1$, $D_\alpha(U||V)$ converges to Kullback-Leibler divergence:

$$\begin{aligned} D_{KL}(U||V) &= \lim_{\alpha \to 1} \frac{1}{1-\alpha} log_a \big(\sum_{i=1}^{n} \frac{u(x_i)^\alpha}{v(x_i)^{1-\alpha}} \big) \\ &= \sum_{i=1}^{n} u(x_i) log_a \frac{u(x_i)}{v(x_i)} \end{aligned} \tag{5}$$

In order to measure the randomness of the probability distribution only, which is independent of the number of distinct values of the feature variable, we need to normalize the entropy by dividing it with $log_a(N_0)$, where N_0 is the number of distinct values for information variable. Then the entropy will be re-defined as follows:

$$H_{ns}(X) = \frac{\sum_{i=1}^{n} p(x_i) log_a (\frac{1}{p(x_i)})}{log_a(N_0)} \tag{6}$$

Then the values of the entropy will fall in the range of $(0, 1)$.

The Shannon entropy and Kullback-Liebler relative entropy could be both utilized for the attack detection in IoT networks. The information metric can overcome the limitations of other intrusion detection methods and can well adapt to the IoT attack detection scenario. We analyze the advantages of employing information metrics to IoT attack detection as follows:

- The typical IoT attacks could be detected through analyzing the networking and application layer traffic. IoT gateway is the interface that bridges the IoT devices and IoT networks, which can aggregate a considerable amount of traffic. The entropy-based approach is more suitable for flow-based attack detection.
- Nowadays, there is a trend that most computations are moving to the edge in order to significantly reduce the latency of services. Compared to the massive computation power in the cloud, the edge usually has relatively low computation power. The entropy-based attack detection approach requires a lower computation cost and can be deployed in the edge and even on the IoT gateway.
- As more attacks have characteristics of covertness, the intrusion detection approach is required to have the capabilities of identifying the low-rate attack types. The entropy-based approach is more sensitive and can well distinguish the difference between legitimate traffic and malicious traffic using a minimum number of attributes.
- Due to the lower computation cost, the entropy-based approach can detect the attack in a real-time fashion.

Furthermore, multiple instances can be running simultaneously due to its excellent scalability, which can be leveraged to monitor different types of IoT attacks. Such an approach also shows the capability of integration into our advanced reinforcement learning based attack detection framework.

The above advantages will be further assessed and demonstrated in the evaluation section. Next, we start to analyze the features in IoT networks that could be used for the attack detection via entropy-based anomaly discovery.

B. Feature Extraction for IoT Attack Detection

We have introduced the entropy as the method to measure the feature variable and discover the potential IoT attacks. Let us take a *ARP spoofing* as an example to explain how the entropy works to detect the attack. Then, we try to explore all the features that could unveil a variety of attacks.

ARP spoofing is a typical attack occurring in the IoT networks. The attacker sends falsified ARP (Address Resolution Protocol) messages over a local IoT network resulting in the link of the attacker's MAC address with the IP address of a legitimate IoT device on the network. Once the attacker's MAC address is bound to an IP address of an authenticated IoT device, the attacker will begin receiving any data that is intended for that IoT device, which can enable the attacker to collect the users' private data and also intercept, modify or even stop data-in-transit. Suppose we use the feature variable X to represent the number of packets transmitted with different protocols during a time window t. For each protocol x_i, we can calculate the probability:

$$p(x_i) = \frac{\text{\# pkts transmitted with protocol } x_i}{\text{Total number of pkts transmitted}} \tag{7}$$

The normalized factor is $log_a(N_0)$, where N_0 is the number of active networking protocols observed during the measured time window. In the window, due to the increased number of ARP packets transmitted caused by ARP spoofing, the probability distribution may show the characteristic of concentration resulting in the decrease of the entropy value.

If the value of entropy is below the preset threshold θ^*:

$$H_{ns}(X) = \frac{\sum_{i=1}^{n} p(x_i) log_a(\frac{1}{p(x_i)})}{log_a(N_0)} < \theta^* \tag{8}$$

That may indicate the presence of an anomaly and alert the defender. Furthermore, ARP spoofing attacks are often used to facilitate other IoT attacks such as Denial-of-service (DoS) attacks. DoS attacks can utilize ARP spoofing to link the set of IP addresses from multiple IoT devices with a single target's MAC address. As a result, all the traffic that is intended for many IoT devices will be redirected to the target's MAC address, overloading the target with traffic. Such attacks are also similar to the TCP/UDP flooding, which immediately flood the requests to the target IoT devices and disable their services. To detect such kinds of attacks, the IoT gateway is the ideal place to aggregate the traffic and conduct the analysis.

In this work, we consider to detect a variety of attacks in IoT networks, including direct attack and reflection attack. The direct attack could be further categorized into network layer attack and application layer attack. Thus, we introduce more features to improve the capability of detection for various attacks.

For a IoT device in the network, it could work as a source address and send packets to multiple destination addresses. Also, it can be regarded as a destination and receive packets from other hosts with different source addresses. For a unique IoT device, we can calculate the probability that target device is the destination:

$$p(x_i) = \frac{\text{\# pkts transmitted from source address } x_i}{\text{Total number of pkts received}} \tag{9}$$

And also calculate the probability that target device is the source:

$$p(x_i) = \frac{\text{\# pkts transmitted to destination address } x_i}{\text{Total number of pkts sent}} \tag{10}$$

On the other hand, the packets transmitted from different ports can reveal some anomaly information used for attack detection. For the TCP/UDP flooding attack, some traffic flows are concentrated on a specific set of ports, resulting in that the following entropy value of port feature will be decreased to a level that causes for alarm:

$$p(x_i) = \frac{\text{\# pkts transmitted with port } x_i}{\text{Total number of pkts transmitted}} \tag{11}$$

However, for the PortScan attack, the attacker will send probe packets to random destination ports instead of a couple of ports, which will make the probability distribution more random, and therefore increase the entropy value. Once the entropy value exceeds a threshold $H_{ns}(X) > \theta^*$, it will also imply a potential attack. However, for simplification, we do not describe this case in our model.

Based on Formula 8, the threshold θ is used to determine whether there has appeared an IoT attack. In most work, the value is pre-defined based on expert experience or historical observation. However, the pre-defined threshold will be immediately invalid if the attacker changes their attack vector or switch its attack approach. For instance, the attacker may transform its high-rate attack to a low-rate attack, which may increase the miss detection rate if the defender still uses its preset threshold.

Therefore, dynamically optimizing the threshold based on the feedback of the detection is a better approach to increase the intrusion detection rate significantly. Furthermore, as the attackers evolve to be more intelligent, the fixed threshold cannot catch up with the dynamic changes of the attacks. Next, we will introduce reinforcement learning to our attack detection model to adapt to the new trend of attack detection.

C. Augmented Attack Detection via Reinforcement Learning

1) Why do we need Reinforcement Learning ?

Reinforcement Learning(RL) [14] is an algorithmic method for solving sequential decision-making problems wherein an

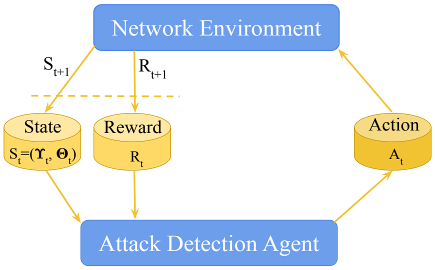


Fig. 3: Iterative reinforcement learning process

agent (or decision-maker) interacts with the environment via iterative processes to learn *how* to respond under different conditions, which is exhibited in Figure 3. Formally, the agent seeks to discover a policy that maps the system state to an optimal solution. Our goal of the work here is to train the agent to learn a policy that maximizes the total number of IoT attacks detected over time, thereby greatly increasing the system utility.

For IoT attacks, in order to persist in the IoT networks and obtain more attack profits, the attackers have to own the properties of *stealth* and *resilience*. In order to maintain the sneak of attacks, the attacker may leverage various methods to obfuscate the observable behaviors of attacks and evade detection. For instance, the attackers can periodically change the spatiotemporal properties of the traffic generated by attacks. With the development of AI and ML, the attackers have a trend to be more intelligent. Once the attacker finds that the specific attack is detected, the attackers could immediately and automatically adjust their attack strategy and continue to pursue attacks against IoT networks, leading to the possession of good resilience. The AI technique can even assist the attacker to automatically develop new attacks, which require the defender to have an advanced attack detection capability.

However, a defender typically has constraints on the number of available resources that can be used for attack detection, such as the limited computation power and human intervention. The defender's objective is to minimize the attack gain of the attacker using a limited number of resources, while the goal of the attackers is to obtain the greatest attack benefits by using fewer resources. Such a scenario can be modeled as a *Markov Decision Processes*, which could be solved using reinforcement learning technique.

2) Problem Statement

Suppose the attackers can launch a variety of attacks denoted by attack vector $A = (a_1, a_2, a_3, ..., a_n)$. The thresholds used to decide the corresponding attacks in time slot t can be represented by $T^t = (\theta^t_{a_1}, \theta^t_{a_2}, \theta^t_{a_3}, ..., \theta^t_{a_n})$. The elements in the attack vector are being observed simultaneously and continuously. Specifically, in time t, if the observed entropy $H^t_{a_i}$ for attack a_i about a feature exceeds the current threshold $\theta^t_{a_i}$. The IoT attack detector sends an alert to the administrator. The detection results can be be positive (identify the malicious traffic as a attack) or negative (identify the malicious traffic as not an attack). The detection results for each subject may or may not match the subject's actual status, which could be summarized as the following four cases:

TABLE I: Environment Feedback

True Positive:	$P_0 - C_0$	False Positive:	$-C_0 - C_1$
False Negative:	$-C_2$	True Negative:	P_1

- True Positive(TP): Malicious traffic identified as an attack
- False Positive(FP): Benign traffic falsely identified as an attack
- True Negative(TN): Benign traffic identified as normal
- False Negative(FN): Malicious traffic falsely identified as normal

For the above different cases, there will be distinct utilities triggered by the environment. If the IoT attack agent reports a potential attack and hands it to the upper layer for further examination, the cost will be denoted as $-C_0$. Finally, if the reported alarm is identified as an attack, the detection system will receive the profits denoted as P_0. Otherwise, if the reported attack is identified as a false alert, the system will get a penalty C_1. If the real attack is not identified and let it pass through, the system will also get a penalty C_2. The last case is that the legitimate traffic is also detected as normal. The corresponding profit that can be obtained is denoted as P_1. The ultimate **system utilities** for different cases are shown in Table I.

Now we calculate the accumulated occurrences for each case during a time period T comprising of n time slots $(t_1, t_2, t_3, ..., t_n)$. We use N^T_{11} to represent the number of authenticated true attacks. The number of false alarm is denoted as N^T_{12}. The number of real attacks but missed is N^T_{21}. N^T_{22} is used to denote the number of authenticated benign traffic flows. Then, in time period T, the complete reward obtained from the environment is:

$$R^T = (P_0 - C_0) * N^T_{11} - (C_0 + C_1) * N^T_{12} - C_2 * N^T_{21} + P_1 * N^T_{22} \tag{12}$$

where

$$N^T_{11} + N^T_{12} + N^T_{21} + N^T_{22} = n \tag{13}$$

Then, the hit rate Υ_T and false alarm rate Θ_T could be computed as follows:

$$\Upsilon_T = \frac{TP}{TP + FN} = \frac{N^T_{11}}{N^T_{11} + N^T_{21}} \tag{14}$$

$$\Theta_T = \frac{FP}{FP + TN} = \frac{N^T_{12}}{N^T_{12} + N^T_{22}} \tag{15}$$

The system state in time period T could be represented as:

$$S_T = (\Upsilon_T, \Theta_T) \tag{16}$$

In this work, we model the attack detection in IoT networks as *Markov Decision Processes*. The goal is to maximize the utility of IoT attack detection system R_T via optimizing the threshold θ_{a_i} for detecting the specific attack type a_i.

$$\theta^*_{a_i} = \underset{\theta_{a_i} \geq 0}{\arg\max}\, R_T \tag{17}$$

In order to maximize the system utility, the feature threshold needs to be appropriately chosen. We desire the detection system not to miss the detection of some attacks. On the other hand, we also do not want the system to report many false alarms, which may require massive human interventions. We need to stimulate the system to conduct attack detection carefully and accurately with corresponding rewards, thereby significantly reducing human interventions. Therefore, we propose a reinforcement learning-enabled attack detection approach to guide the defender's sequential decision-making process over time.

3) Reinforcement Learning based Attack Detection Model

To tackle this problem, we propose to formulate the problem as a *Markov game* G using reinforcement learning, which is defined by a tuple $G = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}$ are the sets of states, joint action space, transition probability functions, and reward functions respectively [15].

The detailed definitions are given as follows:

- **State:** $\mathcal{S} \triangleq \{s^1, s^2, ..., s^{\mathbb{A}}\}$ is the state space of the IoT attack detection agent. The system state s_T in time period T included in $\mathcal{S}$ is denoted as $s_T^i = (\Upsilon_T, \Theta_T)$, which represents the hit rate and false alarm rate with the threshold θ_{a_i} in time period T. In order to compress the state space, we chunk the continuous values of (Υ, Θ) and convert it to the discrete value pairs: $(\Upsilon_T, \Theta_T) \in \{(\Upsilon^i, \Theta^i), i = (1, ..., \mathbb{A})\}$. $\mathbb{A}$ is the size of the state space.
- **Actions:** $\mathcal{A} \triangleq \{a^1, a^2, ..., a^{\mathbb{B}}\}$ is the action space of the IoT attack detection agent. Here, we split the threshold range for attack type θ_{a_i} into $\mathbb{B}$ discrete values. The action $\mathbb{A}_T^j (j \in \{1, ..., \mathbb{B}\})$ specifies the selected threshold for the attack detection in time period T.
- **Reward function:** $\mathcal{R} \triangleq \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ are the reward functions representing the utility of IoT attack detection agent obtained after the specific action is executed in a certain state. The agent attempts to choose the best action (namely threshold) based on its temporal state to maximize its expected discounted reward: $\sum_{d=0}^{\infty} \gamma^d r_{t+d}$. The reward r_t is defined as the utility when the agent executes the action a_t in state s_t.
- **State transition probability:** $\mathcal{P} \triangleq \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ is the transition probability of the system. $p(s_{t+1}|s_t, a_t)$ gives the probability of transiting to s_{t+1} given a action a_t is taken in the current state s_t.

In our model, the attack detection agent starts at some initial state $s^i \in \mathcal{S}$. After observing the current state, the agent selects the action $a^j \in \mathcal{A}$ and the agent will receive the corresponding rewards together with the new observation. At the same time, the system will transit to a new state $s' \in \mathcal{S}$ with probability $p(s'|s, a)$. The procedure is repeated at the new state and continues for a finite or infinite number of stages. The agent tries to find its optimal policies to maximize the expected long-term average rewards. In the context of IoT attack detection, the attack detection agent attempts to continuously optimize the threshold used for attack detection in consecutive time periods and strives to increase the detection rate and decrease the false alarm rate thereby maximizing the long-term average system utility.

Algorithm 1: IoT Attack Detection with Q-learning

```
Input: learning rate $\alpha$, discount factor $\gamma$, $\epsilon$-greedy,
  number of time slots $\mathbb{N}$ in a episode, initialize all the
  $Q$ table entries $Q(s, a)$ to zero, initialized state $s$.
repeat
    Start a new episode.
    Observe current system state $s = (\Upsilon, \Theta) \in \mathcal{S}$.
    Select a threshold $\theta^* = \arg\max_{a \in \mathcal{A}} Q(s, a)$.
    for $i \leftarrow 1$ to $\mathbb{N}$ do
        Compute the entropy value via equation 3.
        if $H_{ns}(X) < \theta^*$ then
            Let the traffic pass through.
        else
            Report the potential IoT attack alarm .
            Further screen the alarm by Defender.
        end
    end
    Obtain the reward $r$ and transform to a new state
      $s'$.
    Update the table entry for $Q_{t+1}(s, a)$ to
      $Q_t(s,a) + \alpha\big[r_t(s,a) + \gamma \underset{a'}{max} Q_t(s', a') - Q_t(s,a)\big]$.
    Replace $s$ with $s'$.
until Attack detection is terminated;
```

4) IoT Attack Detection via Q-Learning

In a reinforcement learning process, an IoT attack detection agent can learn its optimal policy (namely threshold) through interaction with its environment. In particular, the agent first observes its current state, and then takes action, and receives its immediate reward together with its new state. The current state is the attack detection rate and false alarm rate under the present detection threshold. The selection of a new threshold is executed as action and will lead to the updated system reward with the new detection and false alarm rate which is regarded as a new state. The observed information, i.e., the immediate reward and new state from the environment feedback, is also used to adjust the agent's policy, and this process will be repeated until the agent's policy approaches to the optimal policy. In reinforcement learning, Q-learning [16] is the most effective method and widely used in reinforcement learning. In the following, we will discuss how to use the *Q-learning* algorithm to implement IoT attack detection. In Q-Learning Algorithm, we aim to find an optimal entropy threshold θ^* for the attack detection agent to maximize the system utility. Accordingly, we first define value function $\mathcal{V}^\theta : \mathcal{S} \rightarrow \mathbb{R}$ that represents the expected value obtained by the threshold θ from each state $s \in \mathcal{S}$.

$$\begin{aligned} \mathcal{V}^\theta(s) &= \mathbb{E}_\theta\Big[\sum_{t=0}^{\infty} \gamma r_t(s_t, a_t)|s_0 = s\Big] \\ &= \mathbb{E}_\theta\Big[r_t(s_t, a_t) + \gamma \mathcal{V}^\theta(s_{t+1}|s_0 = s\Big] \quad (18) \end{aligned}$$

TABLE II: Description of IoT Attack Dataset.

IoT Devices	WeMo motion, WeMo Switch, Samsung smartcam, TP-Link smart plug, Netatmo Camera Chromecast Ultra, Amazon Echo, Phillips Hue bulb, iHome smart plug, LiFX bulb			
Attack Types	Direct Attack		Reflection Attack	
	ARP spoofing, TCP SYN Flooding, UDP Flooding, and Ping of Death		SNMP, SSDP, TCP SYN, and Smurf	
Attack Parameters	Number of Attacks	200 times	Every Attack Time	10min
	Attack Rates	1, 10, 100 pkts/second	Time Span	16 days

If we denote $Q^*(s,a) \triangleq r_t(s_t,a_t) + \gamma\mathbb{E}_\theta[\mathcal{V}^\theta(s_{t+1})]$ as the optimal Q-function for all state-action pairs, then the optimal value function can be written by $\mathcal{V}^*(s) = \underset{a}{max}\{Q^*(s,a)\}$. Now the problem is reduced to find optimal values of Q-function, i.e., $Q^*(s,a)$, for all state-action pairs, and this can be done through iterative processes. In particular, the Q-function is updated according to the following rule:

$$Q_{t+1}(s,a) = Q_t(s,a) + \\ \alpha_t\Big[r_t(s,a) + \gamma\underset{a'}{max}Q_t(s,a') - Q_t(s,a)\Big] \quad (19)$$

In Formula 19, the learning rate α_t is used to determine the impact of new information to the existing Q-value. The learning rate can be chosen to be a constant, or it can be adjusted dynamically during the learning process. The algorithm yields the optimal policy indicating an action to be taken at each state such that $Q^*(s,a)$ is maximized for all states in the state space, i.e., $\theta^*(s) = \arg\underset{a}{\max}Q^*(s,a)$, which is displayed as a Q-Table. The detailed procedures are described in the Algorithm I. Our model continually updates the feature vectors for various attacks encountered such that it can recognize changing attack patterns. Our model also memorizes and retains the knowledge of previous attacks learned. Once the attacker changes to the attack pattern that appears before, our model can fast identify it without executing the reinforcement learning iterative process.

IV. Evaluation

In order to verify the effectiveness of our IoT attack detection framework, we design and conduct extensive experiments to thoroughly evaluate our approach over an IoT attack dataset from a real IoT platform [17]. First, we use the real IoT attack traces to analyze the performance of our proposed entropy-based anomaly detection features and models. Then, we use a real scenario that the attackers alter their attack patterns among 200 attacks to avoid intrusion detection. We apply our reinforcement learning-based attack detection model to this scenario and demonstrate our technique is capable of learning and recognizing changes in attack patterns, allowing it to adapt to intelligent or dynamically changing attacks unlike traditional static threshold models. We show the evaluation in detail as following.

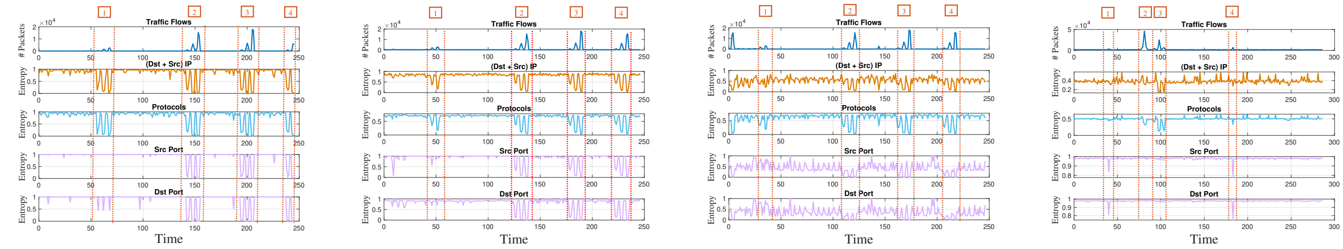
A. Data Set

The IoT dataset [17] is generated by a IoT platform comprising a TP-Link gateway connecting with ten IoT devices, such as TP-Link smart plug, Samsung smart-cam, WeMo motion sensor, etc. The vulnerabilities of these devices are investigated and identified first, and then a variety of attacks are launched including direct attacks, such as ARP spoofing and TCP flooding, and reflection attacks, such as SNMP and Smurf. A total of 200 attacks are launched to different IoT devices and that last for 16 days. The entire packet traces of benign and malicious traffic are stored in 30 *pcaps* files and each *pcaps* contains the traffic traces with a size of $2-3$ GB over a day. Also, the attacks are launched with three rates 1, 10, and 100 packet-per-second (pps), which can reveal the attack effect for both low-rate and high-rate. For each device, the attacks with different types and rates are launched. The detailed parameters about the dataset are described in Table II. The low-rate attacks confuse the distinction between benign and malicious traffic, and the transformation of attack type and rate also significantly boost the difficulty of attack detection. The properties of the dataset can thoroughly exam the performance of our IoT attack detection technique.

B. Entropy-based feature analysis

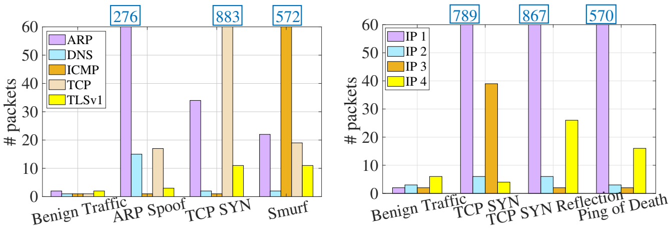
There exist some research works that apply entropy analysis to anomaly detection in cybersecurity. However, they use the traffic traces from traditional Internet mainly comprising of computers and servers to conduct analysis. There is no work that considers the unique characteristics of IoT networks and uses the real IoT dataset for attack detection via using the entropy-based technique. In this work, we revisit the entropy-based anomaly detection approach and explore the effect when it is applied to IoT attack detection.

We evaluate the detection effect of various features we propose in Section III-B. The IoT attacks can be launched in different layers that may impact the traffic for distinct protocols. So the distribution of protocol-related packets between normal and malicious traffic is likely to be distinct. If the distribution of packets concentrates on a certain protocol, that may indicate the presence of an attack. Figure 5a shows the distribution of packets for protocols *(ARP, DNS, ICMP, TCP, and TLSv1)* with and without attacks. The attacks include *ARP spoof, TCP SYN, and Smurf*. The data of the Figure is computed based on the trace set on 06/01/2019 for TP-Link smart plug. For the benign traffic, the distribution of packets about different protocols displays randomness resulting in the low entropy value. But the three attacks display the less randomness and more aggregation for some protocol, which will lead to the larger entropy value. Figure 5b displays the distribution of packets under four distinct IP addresses with and without attacks. The attacks include *TCP SYN, TCP SYN*



(a) TP-Link smart plug (b) Samsung smartcam (c) WeMo power switch (d) WeMo motion sensor

Fig. 4: Time series of entropy data for four IoT devices with different features.



(a) Protocols. (b) IP addresses.

Fig. 5: Distribution of transmitted packets from different protocols and IP addresses with and without IoT attacks.

Reflection, and Ping of Death. The Figure reveals that the attacks lead to the explosion of packets in some IP address, which also results in much larger entropy value and manifests potential anomaly.

Moreover, we compute the entropy values of features with 5 min period using the traffic traces from TP-Link smart plug, Samsung smart-cam, WeMo power switch on 06/01/2019 and WeMo motion sensor on 06/02/2019, which is shown in Figure 4. The four attacks, namely *ARP Spoof (layer: L2D), TCP SYN (layer: L2D), TCP SYN Reflection (layer: W2D2W) and TCP SYN Reflection(layer: L2D2L)*, are launched to the first three devices. Another four attacks, namely *UDP Flood (layer:L2D), SSDP (layer:D2W), SSDP (layer:L2D2L) and UDP Flood (layer:W2D)* are launched to the WeMo motion sensor.

Figure 4 labels all the launched attacks, and we have some findings from the figure. As shown, the entropy of IP and protocol are sensitive to most attacks and can achieve a good detection effect. However, for the UDP Flood attack, the difference of entropy values for benign and malicious traffic is not apparent, which is revealed in Figure 4d. Nevertheless, the entropy of port between benign and malicious shows a more substantial difference, which can be better utilized for detecting the attack. The benefit of entropy is that it can detect the attack even with the low-rate attack. For the ARP spoofing attack, we can find that the entropy of IP and protocol is still able to show the apparent fluctuation even with relatively lower transmission rates. For the WeMo power switch, the difference of entropy values between normal and malicious seems not to be much apparent than other devices, but the threshold can be still appropriately chosen to distinguish the normal and anomaly traffic. Therefore, we combine features and build the feature vectors to detect the attacks robustly. Next, we evaluate our reinforcement learning-based detection technique that dynamically chooses the threshold for feature vectors to determine the attacks.

C. *Performance of reinforcement learning model*

Our reinforcement learning-based attack detection technique is to optimize the feature threshold to determine the attack continuously. The new trend that the attacker is evolving to be more intelligent requires the defender to develop the capability to adjust their defense strategy based on the variation of attacks. The traditional method relies on data statistics to determine a proper threshold at a given time. However, such method requires more human interventions to choose a new threshold once the attacker changes their attack patterns. Our approach can smartly and automatically detect attacks with less human supervision, and also significantly lower the delay of detection.

We design experiments to analyze the performance of our proposed RL-based IoT attack detection and compare it with the traditional attack detection with static threshold models. For the experiments, we set the values of parameters P_0, P_1, C_0, C_1, C_2 in Table I to be 14, 12, 0, 3, and 15. The parameters define the penalty and reward that our reinforcement learning model can obtain. For the parameters about Q-Learning, the values of α, γ and ϵ are set to 0.1, 0.8 and 0.9 respectively. We stimulate the model to report the alarm accurately. If the model reports one alarm that is identified as an attack. The model can obtain the appropriate incentive. Otherwise, it receives the corresponding penalty.

We maintain a threshold vector for the same type of devices to store the threshold for each feature used for detecting the anomaly. Different types of IoT devices may have a distinct threshold vector due to the heterogeneity of their firmware, driver, and software. Figure 6 shows a snapshot of the feature threshold at some point for TP link plug and Samsung camera, respectively. The feature index denotes a part of the features we propose in Section III-B. For the same IoT device, the thresholds for each feature are also distinct. For static threshold models, the threshold vector is determined at the beginning and will not be changed unless human intervenes. Our learning

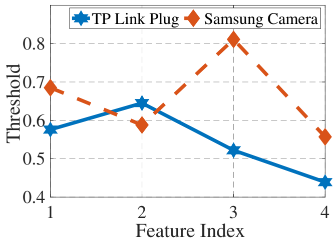


Fig. 6: Snapshot of thresholds for attack detection.

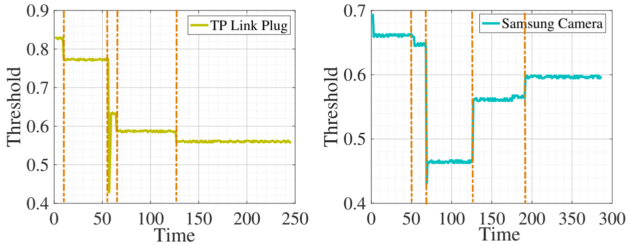


Fig. 7: Reinforcement learning-based feature threshold optimization for attack detection.

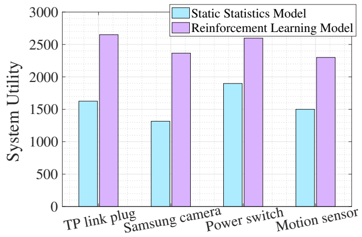


Fig. 8: Comparison of system utility with and without RL.

agent continuously optimizes and updates the threshold vector for attack detection with much less human intervention. The continuously changing threshold for the feature IP address over a day is displayed in Figure 7. Because we change the type of attacks and also adjust their attack rate, the corresponding boundary to distinguish benign and malicious traffic may be changed, leading to the variation of the threshold. The vertical dotted lines in the figure label the time point the attacker changes its attack type or attack rate. As shown, the reinforcement learning model can automatically adapt to the transformation of attacks and make adjustments for the threshold accordingly.

The utility of the IoT attack system is significantly improved because of the better detection and false alarm rate resulting from the continuous optimization of the threshold. We investigate the state-of-the-art work about IoT attack detection. We are the first to introduce the reinforcement learning technique to the detection model. Most research works like [8] use machine learning or deep learning to train an attack model and use it for future detection. We still regard them as the static statistics model because they do not continually update their model. We apply the static statistics model and our reinforcement learning model to detect the attacks for four devices. For each device, a certain number of attacks are launched with different attack patterns. Figure 8 shows the utility comparison over a day for the four devices. The system utility can achieve an average 54.7% increase across four devices when using the reinforcement learning technique compared to systems without it. With the change of the IoT attack pattern, the static threshold may gradually deviate from the optimal value resulting in bad detection rate and high false alarm rate, which will significantly reduce the system utility. Our reinforcement learning model is robust to the change of the IoT attack pattern and can dynamically update the feature threshold and significantly boost the attack detection rate to 98.5%. Our IoT attack detection framework can also be widely applied for other intrusion detection fields and facilitate the current research in IoT security and reinforcement learning.

V. Related Work

A. IoT security and privacy

As the development and broadly deployment of IoT networks, more and more vulnerabilities involved in IoT platforms, applications, protocols, and hardware are being exposed to the attackers, which causes the critical security and privacy issues. [18] proposes a mechanism to protect the users from insecure devices states via monitoring the IoT behaviors and their corresponding platform apps. Fine-grained context identification [19] is introduced in the IoT platform to analyze the sensitive actions, thereby providing effective access control. SmartAuth [20] uses Natural Language Processing (NLP) to extract the security information from an IoT app's description, code, and annotations, and then compare to the real actions that the app performs.

The communication protocols, such as ZigBee, Z-Wave, and BLE, that are used in IoT networks also bring out some security and privacy issues. For instance, the BLE protocol fails to hide the device's presence from the adversaries [21], which will lead to severe threats, such as user behavior tracking and inference of sensitive information. HoMonitor [22] monitors the encrypted wireless traffic and analyze to check if the behaviors are matched with the app's description. In the aspects of hardware and software, the fingerprinting technique [23] is proposed to secure the authentication and communication of IoT devices.

B. Machine learning in IoT networks

In order to develop a secure, robust, and optimized solution, machine learning is introduced and now is widely adopted to solve issues in IoT networks [24]. [9], [25] uses the machine learning technique to detect the attacks in IoT networks. IoT Sentinel [26] analyzes the networking traffic and uses the classification algorithms to identify the types of devices, which enables enforcement of rules for constraining the communications of vulnerable devices. A deep neural network-based framework is proposed in [27] to implement the real-time nodes authentication in wireless networks.

In addition, due to the excellent learning characteristics, reinforcement learning is more suitable for some networking scenarios. [28] proposes an RL-based scheduler that can dynamically adapt to traffic variation, and various reward functions set by network operators, to optimally schedule IoT traffic. [29] provides a security solution that uses deep reinforcement learning for signal authentication of IoT systems. [30] proposes a spoofing detection mechanism in wireless networks that also leverages reinforcement learning to increase detection accuracy. [11], [31] also propose a deep reinforcement learning-based intrusion detection model for the traditional network. As we can see, there is no research work considering the characteristics of intrusion in modern IoT networks. We explore the attack characteristics in IoT networks and propose a dynamical intrusion detection model that can adapt to the transformation of intrusion strategies from attackers.

VI. Conclusion

In this paper, we explore the unique characteristics and trends of attacks in IoT networks. We argue that the traditional intrusion detection approaches in cybersecurity cannot well adapt to the IoT scenario. Due to the heterogeneity of platforms, protocols, software, and hardware, diverse vulnerabilities are exposed to attackers resulting in the increased complexity of anomaly detection. Meanwhile, the new emerging low-rate attacks obfuscate the boundary between benign and malicious attacks. Furthermore, the attacker is evolving to have the capabilities of altering the attack strategies and even developing a new attack based on the environment feedback, which requires the defender to have a quick response. Therefore, we propose an entropy-based attack detection framework integrating the reinforcement learning model, which is more suitable for the IoT context. We conduct extensive experiments over a real IoT attack dataset and demonstrate the efficiency of our IoT attack detection approach.

References

[1] "Internet of things connected devices installed base worldwide from 2015 to 2025." https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/, 2019.

[2] Samsung, "Smartthings." https://www.smartthings.com, Accessed: 2015.

[3] Google, "Google home." https://developers.google.com/iot/, Accessed: 2015.

[4] Apple, "Homekit." https://developer.apple.com/homekit/, Accessed: 2015.

[5] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the internet of things: a survey of existing protocols and open research issues," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1294–1312, 2015.

[6] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *computers & security*, vol. 28, no. 1-2, pp. 18–28, 2009.

[7] X. Yuan, C. Li, and X. Li, "Deepdefense: Identifying ddos attack via deep learning," in *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 1–8, May 2017.

[8] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," *NDSS'18*, 2018.

[9] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, "Network intrusion detection for iot security based on learning techniques," *IEEE Communications Surveys & Tutorials*, 2019.

[10] M. S. H. Ll, J. I. Vélez, L. Castillo, *et al.*, "Distributed denial of service (ddos) attacks detection using machine learning prototype," in *Distributed Computing and Artificial Intelligence, 13th International Conference*, pp. 33–41, Springer, 2016.

[11] G. Caminero, M. Lopez-Martin, and B. Carro, "Adversarial environment reinforcement learning algorithm for intrusion detection," *Computer Networks*, vol. 159, pp. 96–109, 2019.

[12] E. Hou, K. Sricharan, and A. O. Hero, "Latent laplacian maximum entropy discrimination for detection of high-utility anomalies," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 6, pp. 1446–1459, 2018.

[13] M. Xie, J. Hu, S. Guo, and A. Y. Zomaya, "Distributed segment-based anomaly detection with kullback–leibler divergence in wireless sensor networks," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 101–110, 2016.

[14] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[15] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Communications Surveys Tutorials*, vol. 21, pp. 3133–3174, Fourthquarter 2019.

[16] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *AAAI'16*, 2016.

[17] A. Hamza, H. H. Gharakheili, T. A. Benson, and V. Sivaraman, "Detecting volumetric attacks on lot devices via sdn-based monitoring of mud activity," in *Proceedings of the 2019 ACM Symposium on SDN Research (SOSR'19)*, pp. 36–48, ACM, 2019.

[18] Z. B. Celik, G. Tan, and P. McDaniel, "Iotguard: Dynamic enforcement of security and safety policy in commodity iot," *NDSS'19*, 2019.

[19] Y. J. Jia, Q. A. Chen, S. Wang, A. Rahmati, E. Fernandes, Z. M. Mao, A. Prakash, and S. J. Unviersity, "Contexlot: Towards providing contextual integrity to appified iot platforms.," in *NDSS*, 2017.

[20] Y. Tian, N. Zhang, Y.-H. Lin, X. Wang, B. Ur, X. Guo, and P. Tague, "Smartauth: User-centered authorization for the internet of things," in *26th {USENIX} Security Symposium Security 17)*, pp. 361–378, 2017.

[21] K. Fawaz, K.-H. Kim, and K. G. Shin, "Protecting privacy of ble device users," in *25th {USENIX} Security Symposium Security 16)*, pp. 1205–1221, 2016.

[22] W. Zhang, Y. Meng, Y. Liu, X. Zhang, Y. Zhang, and H. Zhu, "Homonit: Monitoring smart home apps from encrypted traffic," in *ACM CCS'18*, pp. 1074–1088, ACM, 2018.

[23] T. Gu and P. Mohapatra, "Bf-iot: Securing the iot networks via fingerprinting-based device authentication," in *2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pp. 254–262, IEEE, 2018.

[24] E. Benkhelifa, T. Welsh, and W. Hamouda, "A critical review of practices and challenges in intrusion detection systems for iot: Toward universal and resilient systems," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3496–3509, 2018.

[25] N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things," *IEEE Internet of Things Journal*, 2018.

[26] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma, "Iot sentinel: Automated device-type identification for security enforcement in iot," in *IEEE ICDCS'17*, pp. 2177–2184, IEEE, 2017.

[27] B. Chatterjee, D. Das, S. Maity, and S. Sen, "Rf-puf: Enhancing iot security through authentication of wireless nodes using in-situ machine learning," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 388–398, 2019.

[28] S. Chinchali, P. Hu, T. Chu, M. Sharma, M. Bansal, R. Misra, M. Pavone, and S. Katti, "Cellular network traffic scheduling with deep reinforcement learning," in *AAAI'18*, 2018.

[29] A. Ferdowsi and W. Saad, "Deep learning for signal authentication and security in massive internet-of-things systems," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1371–1387, 2019.

[30] L. Xiao, Y. Li, G. Han, G. Liu, and W. Zhuang, "Phy-layer spoofing detection with reinforcement learning in wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 10037–10047, 2016.

[31] M. Lopez-Martin, B. Carro, and A. Sanchez-Esguevillas, "Application of deep reinforcement learning to intrusion detection for supervised problems," *Expert Systems with Applications*, vol. 141, p. 112963, 2020.