Reinforcement Learning Enabled Intelligent Energy Attack in Green IoT Networks

Long Li, Yu Luo, Jing Yang, and Lina Pu, *Member, IEEE*

***Abstract*— In this paper, we study a new security issue brought by the renewable energy feature in green Internet of Things (IoT) network. We define a new attack method, called the malicious energy attack, where the attacker can charge specific nodes to manipulate routing paths. By intelligently selecting the victim nodes, the attacker can "encourage" most of the data traffic into passing through a compromised node and harm the information security. The performance of the energy attack depends on the charging strategies. We develop two reinforcement-learning enabled algorithms, namely, Q-learning enabled intelligent energy attack (Q-IEA) and Policy Gradient enabled intelligent energy attack (PG-IEA). Through interacting with the network environment, the attacker can intelligently take attack actions without knowing the private information of the IoT network. This can greatly enhance the adaptability of the attacker to different network settings. Simulation results verify that the proposed IEA methods can considerably increase the amount of traffic traveling through the compromised node. Compared with the network without attack, an additional 53.3% data traffic is lured to the compromised node, which is more than 4 times higher than the performance of Random Attack.**

***Index Terms*— Green IoT networks, security, malicious energy attack, reinforcement learning.**

I. Introduction

THE rapid development of the Internet of things (IoT), body area network (BAN), and smart infrastructure involves an ever-increasing number of sensors and actuators. Powering a large number of low-power devices in these applications is a great challenge, as battery replacement is time-consuming and cost-inefficient. This encourages us to utilize renewable energy to meet the clean and self-sustainable requirements of the coming green revolution [1], [2].

Through scavenging energy such as sunlight, wind, electromagnetic waves, and biothermal energy from the surrounding environment, an energy harvesting node (EHN) in a green IoT network can run semi-perpetually without any battery replacement [3]. Although the efficiency of energy harvesting greatly depends on external sources, which is susceptible to atmospheric changes and physical obstacles, still, the energy harvesting technique offers a promising solution for extending the lifespan of green IoT network. Consider that radio frequency (RF) energy radiated from cellular base stations, TV towers, and Wi-Fi access points is widely available in both indoor and outdoor environments. In this work, we mainly focus on RF energy harvesting.

The energy harvesting ability greatly extends the sustainability and scalability of green IoT networks; nevertheless, it causes some new security issues. In [4] [5] [6], eavesdropping attacks are implemented by taking advantage of the absence of encryption in RF energy harvesting network due to power constraints. Other research like [7] [8], implement some traditional attack methods (e.g. DoS, jamming) to disrupt the availability of EH networks by making use of RF energy harvesting features. Most attack methods are implemented at the physical layer, but still utilize the traditional way to compromise the network. Those issues are less threatening since tons of research has been studied on detection and prevention from those traditional attack methods [9]. Beyond that, in some researches [5], the author attacks the energy source rather than sensor nodes in a wireless power transfer network. The goal is to trick the energy source to transfer more energy to the attacker so that other nodes drain their energy faster. This new type of attack method needs the network to include an energy source that intentionally transfers energy over the air. But still, this kind of attack is easy to detect since it is a direct attack on the network.

In this paper, we propose a new attack method, called *malicious energy attack*, which exploits the energy-aware properties of routing protocols in green IoT networks [10]. Energy-aware routing protocols have been widely adopted in power-constrained networks to extend the lifespan of wireless networks with a limited energy supply [11]. In malicious energy attack, the malicious energy source (MES) manipulates routing paths in the network layer by consciously charging specific EHNs [12]. The infected nodes that receive extra energy from the energy attacker will become more active than ordinary nodes to work as data forwarders or information aggregators. As shown in Fig. 1, if the MES is able to select the infected nodes properly, it can manipulate the routing path and "encourage" most data traffic to pass through the compromised node that deviates from the source and the destination.

Different from the traditional *primary* attack methods, such as black hole, wormhole, selective forwarding, Sybil attack, and acknowledgment spoofing, that directly attack the routing protocol (i.e., on information plane), our proposed malicious energy attack is considered as a *secondary* attack incurred

Manuscript received August 23, 2021; revised December 13, 2021 and January 21, 2022; accepted January 23, 2022. Date of publication February 4, 2022; date of current version February 18, 2022. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Alexey Vinel. *(Corresponding author: Lina Pu.)*

Long Li and Lina Pu are with the Department of Computer Science, The University of Alabama, Tuscaloosa, AL 35487 USA (e-mail: lli90@crimson.ua.edu; lina.pu@ua.edu).

Yu Luo and Jing Yang are with the Department of Electronics and Communication Engineering, Mississippi State University, Starkville, MS 39762 USA (e-mail: yu.luo@ece.msstate.edu; jy599@msstate.edu).

Digital Object Identifier 10.1109/TIFS.2022.3149148

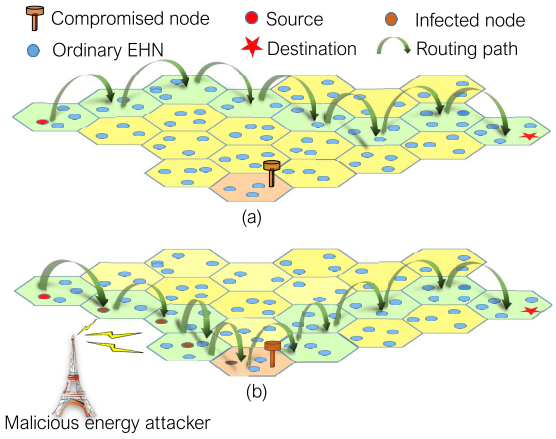


Fig. 1. An example of malicious energy attack. (a) An ordinary green IoT networks without energy attack chose a shorter path from Source to Destination. (b) An green IoT networks attacked by the MES chose a path through the Compromised node.

on the energy plane. It does not fabricate messages or cause harmful interference to the routing protocol. Instead, the attacker actively provides extra energy to selected nodes in order to attract interested data traffic to a specific IoT node. We call the energy attack secondary attack since the MES can not directly profit from the energy attack, but can greatly improve the efficiency of other primary attack methods. In this article, we use eavesdropping as an example to illustrate how the malicious energy attack creates opportunities for eavesdroppers to sniff confidential data from any target node. The malicious energy attack can greatly threaten the security of green IoT networks and has not been fully studied.

The amount of traffic that can be "lured" to the compromised node heavily depends on the energy distribution of EHNs with the energy-aware routing. To optimize the efficiency of malicious energy attack, MES needs to wisely decide which nodes in a green IoT network should be attacked. This question may not be difficult to answer if we know the global network status of green IoT networks (i.e., network topology, energy harvesting rate and the instant remaining energy of each node, traffic rate on each node, etc.) and the parameters of the routing protocol in path selection. The optimal energy attack can be formulated as a deterministic optimization problem [13]. Unfortunately, in practice, an attacker is very likely to face an unknown network environment. For this reason, we need to develop some strategies for the attacker to make intelligent attack decisions without knowing network and routing configuration.

The recent development of the reinforcement learning (RL) technique provides a promising solution to tackle the above challenges [14]. Inspired by the powerful ability of RL to interact with an unknown environment, we propose an RL-enabled intelligent energy attack strategy in this paper. The Q-learning algorithm and Policy Gradient algorithm are implemented on the MES to help find an optimal attack strategy that can maximize the data traffic lured to the compromised node. The malicious energy attacker will train itself intelligently to improve its attack pattern by interacting with the green IoT network.

The main contributions of this paper are three folds:

- First, we identify a new attack method in energy-aware green IoT networks, called malicious energy attack. The malicious energy attack manipulates the routing path at the network layer by intentionally charging specific EHNs. As an emerging attack method, the malicious energy attack is immune to many existing security mechanisms since it is an indirect attack method that disrupts the network protocols through energy.
- Secondly, we study how to enhance the efficiency of malicious energy attack via the RL. By applying reinforcement learning, the MES can attack the network intelligently without knowing the global network settings or the routing protocols. The reinforcement learning-enabled intelligent energy attack significantly outperforms the energy attack methods without learning ability.
- Thirdly, we conduct a simulation to verify the effectiveness of the malicious energy attack as well as the performance of the proposed RL-based algorithm. The simulation results show the effectiveness and the efficiency of our proposed RL-based algorithm. From the simulation results, the proposed algorithm lures an additional 53.3% data traffic travel through the compromised node compared to the original network without attack. The performance of the proposed algorithm is more than 4 times higher than the Random Attack method.

The rest of this paper is organized as follows: Section II provides some background. The implemented system and attack model are presented in Section III. A glance at the malicious energy attack and preliminary work are in Section IV. The intelligent energy attack policy is proposed in Section V, and a critical proposed training trick Pre-Train is discussed in Section V-C. Simulation results and analysis are discussed in Section VI, followed by the conclusion in Section VII.

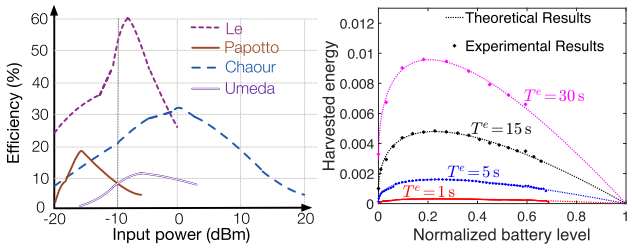
II. Background

In this section, we will introduce some related background on RF energy harvesting, energy-aware routing and routing security. In addition, we will briefly introduce RL and discuss in detail the two RL algorithms that we implemented on MES.

A. Nonlinear Charging Features in RF Energy Harvesting

The RF EHN harvests energy from RF energy sources (e.g., TV tower, cellular base station, etc.) and stores harvested energy in a battery. The harvested energy is then used for data transmission. As revealed in our previous work [15], due to the nonlinear circuit and nonlinear battery charging, the charging process of RF EHN has obvious nonlinearity, which will significantly impact the attack efficiency of malicious energy attack.

a) *Circuit nonlinearity:* The harvester circuit exhibits a nonlinear characteristic due to the nonlinearity of diodes and the parasitic influence of the used elements in the rectifier and voltage multiplier [15]. We redraw the Power Conversion Efficiency (PCE) of the harvester circuits designed by Le *et al.* [16], Papotto *et al.* [17], Chaour *et al.* [18], and Umeda *et al.* [19] in Fig. 2(a).



(a) Nonlinear PCE (b) Nonlinear battery charging

Fig. 2. Nonlinear charging features in RF energy harvesting [15].

It is clear that all harvester circuits exhibit a nonlinear PCE with regard to input power. The nonlinear PCE implies that in order to achieve good energy efficiency, the energy attacker needs to charge the victim EHNs at moderate power. In other words, it is inefficient to attack EHNs with excessive power just to increase the impact of MES on the energy distribution of EHNs.

b) *Nonlinear battery charging:* The nonlinear battery charging refers to the nonlinear relationship between the harvested energy and the battery level in the harvesting process. The theoretical and experimental results in Fig. 2(b) reveal that the amount of harvested energy is not constant, but varies nonlinearly with respect to the normalized battery level even when the same amount of energy is provided by the energy source. Due to this nonlinear battery charging feature, the charging current approaches zero when a battery is nearly full and it will take infinite time to fully charge the EHN.

Considering the circuit nonlinearity and nonlinear battery charging, an intelligent energy attack needs to balance the attack efficiency and energy efficiency. In this work, we assume that the MES charges EHNs at a power that maximizes the PCE. The amount of energy to be charged to EHNs is adjusted by controlling the charging time.

B. Energy Aware Routing and Routing Security

In power-constrained green IoT networks, energy awareness is an essential property of routing protocols to extend the lifetime of the network. The shortest path may not be optimal, especially if nodes on that path have insufficient energy. Excessively frequent transmissions on certain nodes will cause early depletion of energy and have serious consequences for network connectivity. Therefore, to prolong the network lifetime, packet forwarding is usually scheduled on nodes with sufficient energy. The low-energy nodes by contrast tend to stand by to conserve energy. Thus, the high-energy nodes are more likely to be selected as data forwarders [13] or information aggregators [20] in the energy-aware routing protocols.

In networks with energy harvesting capabilities, the harvest rate of EHNs is another important factor that affects route selection. The harvesting-aware routing protocols are designed to align the traffic load with the energy harvesting rate at different nodes: the higher the energy harvesting rate, the higher opportunity to be selected as data forwarder [21]. The representative harvesting-aware routing protocols used in RF energy harvesting networks include joint routing and charging (J-RoC) [22], routing-first heuristic algorithm [23], and energy-opportunistic weighted minimum energy (E-WME) routing [11]. All of those protocols aim to find an optimal balance between energy constraint and throughput.

In the rest of the paper, E-WME [11] is selected as our routing protocol to demonstrate the impact of malicious energy attacks. In E-WME, it uses (1) to calculate the forwarding cost for each EHN. It has both energy-aware and harvest-aware abilities. The forwarding cost is formulated as an exponential function of the node residual energy, λ, a linear function of the transmit and receive energies, e, and an inversely linear function of the harvesting rate, r. The parameter μ is an appropriately chosen constant. The E-WME will select the route path with the smallest sum forwarding cost for data transmission.

$$C = \frac{1}{r \log \mu}(\mu^{1-\lambda} - 1)e \tag{1}$$

Routing security plays a critical role in protecting data privacy and maintaining network stability. In the current literature, adversary users attempt to threaten network security through a variety of attack methods, such as the black hole, wormhole, selective forwarding, Sybil attack, and acknowledgment spoofing [24]. In the traditional network layer attack method, the attacker usually needs to gain full control of at least one legitimate node to insert illegal routing information into the network. Since the legitimacy of the information can be verified by inserting an artificial imprint (e.g., cryptography, packet identification, and preamble), the attack can be easily detected. The secure routing design of green IoT networks to protect the network security in the information plane [25], [26] has been extensively studied in the literature.

However, in a malicious energy attack, the routing path is intentionally manipulated, not by injecting bogus routing information or creating artificial high-quality links, but by changing the energy level of EHN. Cryptography cannot prevent such attack, because the energy attack occurs on the energy plane and routing information will not be modified or fabricated. In addition, unlike the wormhole attack, which transmits the packets to a distant node in the network and thus can be detected by measuring the distance of a single hop, the geographic information does not help detect the energy attack.

Even if a defense mechanism is implemented, as long as EHN obtains energy from the environment, the attacker only needs to retrain the RL model without knowing the setting changes in the network, since the network environment is a black box for the RL algorithm. Due to the learning ability, the retrained RL model will find an optimal attack strategy with the presence of countermeasure. However, the performance of malicious energy attack will decrease as the complexity of the security mechanism increases. But it should be noted that the IoT devices are more concerned about resource limitation than the attacker. These features make the malicious energy attack immune to many security mechanisms and difficult to defend or prevent.

C. Reinforcement Learning

RL is becoming more and more popular because of its ability to learn the optimal policy through a trial-and-error search with delayed feedback [27]. A standard learning process of RL starts with the initial state s perceived from the environment. Then based on this state, an action $a \in A$ will be decided by the agent following its current policy π [28]. After that, the reward r, as well as a new state s', will be obtained from the environment to optimize the agent policy. The goal is to find an optimal policy that produces the largest cumulative reward via a trial-and-error manner [29].

Among the commonly used model-free RL algorithms, Q-learning and Saras are widely used for discrete and less computationally intensive tasks because of their simplicity and ease of implementation. For more complex tasks with large state and action spaces and high computational cost, a neural network is used to approximate the value function or the target policy. Representative algorithms include Policy Gradient, Deep Q Network (DQN), and Deep Deterministic Policy Gradient (DDPG).

In this article, we choose Q-learning and Policy Gradient algorithms to implement intelligent energy attack from the perspectives of computational cost and hardware requirements. Q-learning is a simple yet effective algorithm: it is expected to enhance energy attack performance at low computational costs and with low hardware requirements. Compared to a similar algorithm, Sarsa, Q-learning is an off-policy algorithm. It estimates the Q-value for state-action pairs based on the optimal greedy policy, independent of the agent's action selection policy. As an off-policy algorithm, Q-learning can converge to the optimal strategy much faster than Sarsa. We choose Q-learning because of its cost-effectiveness. Policy Gradient is more resource-consuming than Q-learning but also more efficient: it is expected to achieve higher performance in terms of attack efficiency as will be verified in Section VI-C. Compared with deterministic algorithms like DQN and DDPG, Policy Gradient can better solve the non-deterministic issue in our energy attack scenario. Due to the imperfect state information and limited knowledge of the global network, the optimal policy under a certain state should not be fixed. The Policy Gradient that generates a probability distribution of actions perfectly solves the uncertainty challenges in the intelligent energy attack.

In Q-learning enabled intelligent energy attack (Q-IEA), the continuous state (i.e., the energy level of nodes) is discretized into ten levels and the action space is 1 and 0 indicating whether the node is under energy attack or not. The Policy Gradient enabled intelligent energy attack (PG-IEA) is used to deal with the more practical situation with continuous state space (i.e., energy level) and larger action space (i.e., how much energy is charged). Next, we will introduce Q-learning and Policy Gradient algorithms in more details.

1) Q-Learning Algorithm: A Q-learning agent uses the Q value to evaluate the effectiveness of each $action$ in a specific $state$. The agent maintains a Q table to record the learned experience. As an off-policy algorithm, the Q-learning agent uses a decaying $\epsilon - greedy$ policy to fully explore the environment to approach the optimal target greedy policy. Compared with the on-policy algorithm Saras, Q-learning can approach the optimal policy faster. After training, the agent greedily chooses the action with the highest Q value to maximize the reward.

TABLE I
NOTATION TABLE

Symbol	Meaning
s	States
a	Actions
r	Rewards
t	Index of time slots
α	Learning Rate
γ	Discount Factor
θ	Policy Neural Network parameters
$\mathrm{R}(s_t, a_t)$	Reward under state s_t by taking action a_t
π_θ	Policy parameterized by θ
$J(\cdot)$	Cumulative Reward Function
$d(s)$	transition distribution of state s
E	Mathematical expectation
C	Cross entropy between action and pdf

A typical equation used to update the Q value is depicted in (2), where all notations can be found in Table I. The learning rate α and discount factor γ in (2) are two important hyperparameters that affect the training process. The learning rate controls the aggressiveness of learning. The higher the learning rate, the more the agent will rely on the current reward, and less on the knowledge learned from past experiences. The discount factor controls the prediction of the future rewards, allowing the agent to have a longer-term view.

$$Q(s_t, a_t) = (1-\alpha)Q(s_t, a_t) + \alpha[R(s_t, a_t) + \gamma * \max_a Q(s_{t+1}, a)] \quad (2)$$

2) Policy Gradient Algorithm: Different from Q-learning, the Policy Gradient algorithm is a so-called policy-based algorithm that directly uses a neural network instead of an intermediate Q table to approximate the target policy. Once $state$ is given, the Policy Gradient agent generates a probability distribution of the entire action space. The good actions with higher rewards will have higher probability, while the bad actions have lower chances to be chosen.

We use $C * r$ as the loss function in PG-IEA, where C is the cross-entropy between the final executed action and the predicted probability distribution. Cross-entropy is widely used in supervised machine learning to derive the error between prediction and the ground truth. In our proposed PG-IEA, we use the final executed action as the ground truth, since the actual ground truth is not available to the MES. In this way, the PG-IEA algorithm always assumes the executed action is good and later updates the neural network to minimize the error so that the prediction can gradually approach the executed action. Considering that the executed $action$ may not always be optimal during training, we multiply it with $reward$, r, to indicate how much the neural network should update toward the selected $action$. If r is very large, it indicates a good

action, and a big step is taken to approach this $action$. If r is very small or even 0, the model will only take a very small step or even do not update the model. In this way, the model will finally converge to the optimal policy by minimizing the loss function.

With the defined loss function, the PG-IEA agent aims to search for an optimal parameterized policy π_θ, which maps the state to a probability distribution on the actions space [30]. θ represents the parameters of the Policy Gradient neural network. Given the instantaneous state s and the action a, the cumulative loss function can be written as (3).

$$J(\theta) = \sum_s d(s) \sum_a \pi_\theta(a, s) R(s, a) C(a). \tag{3}$$

Here, $d(s)$ represents the transition distribution of state s, and $\pi_\theta(a, s)$ denotes the probability of selecting action a under state s. All notations can be found in Table I. If we want to update the Policy Gradient neural network by a gradient, we will need to calculate the gradient of the target function, $\nabla_\theta J(\theta)$. The direct calculation of $\nabla_\theta J(\theta)$ is very difficult because it depends on both the action selection and the state distribution.

The $\nabla_\theta J(\theta)$ can be expressed as:

$$\nabla_\theta J(\theta) = \nabla_\theta \left(\sum_s d(s) \sum_a \pi_\theta(a, s) R(s, a) C(a) \right). \tag{4}$$

In (4), $d(s)$ is the state distribution following π_θ, which is indirectly determined by π_θ. It is hard to calculate the gradient of $d(s)$ since the global network environment is unknown to the attacker. Luckily, the intelligent energy attack is a one-step Monte-Carlo learning progress. Therefore, $d(s)$ is not a function of θ. Similarly, neither $R(s, a)$ nor $C(a)$ depends on θ. We can simplify the gradient of target function to be:

$$\nabla_\theta J(\theta) = \sum_s d(s) \sum_a \left(R(s, a) C(a) \nabla_\theta \pi_\theta(a, s) \right). \tag{5}$$

$\nabla_\theta \pi_\theta(a, s)$ can be rewrote as $\pi_\theta(a, s) \nabla_\theta \log \pi_\theta(a, s)$. Hence, the gradient of the target function can be expressed as:

$$\nabla_\theta J(\theta) = E \left(R(s, a) C(a) \nabla_\theta \log \pi_\theta(a, s) \right). \tag{6}$$

Here, we only need to calculate the gradient of the policy $\pi_\theta(a, s)$. Once the gradient is known, we can use this gradient to update the parameters θ according to (7).

$$\theta_{t+1} = \theta_t + \alpha \, r_t \, C(a_t) \nabla_\theta \log \pi_\theta(a_t, s_t), \tag{7}$$

where, a_t and s_t are the action and state at time t respectively, and r_t is the reward, which is the amount of interested traffic attracted to the compromised node in our energy attack scenario.

The Q-learning and Policy Gradient algorithms discussed above both aim to search for the optimal policy, but in different ways. In Q-learning, the agent uses the Q value to indicate the effectiveness of each $state - action$ pair. Therefore, as long as enough $state - action$ pairs are visited, the Q-learning can guarantee to find the optimal policy. However, this no longer holds in Policy Gradient, because it uses a neural network to approximate the optimal policy. Each weight update in the neural network will affect the output of all other states, which makes it achieve better training efficiency than Q-learning because fewer state visits are required to train the neural network. The drawback is that the neural network is only an approximation to the optimal policy [31]. Thus, the Policy Gradient algorithm can approximate but may never reach the optimal policy [31].

III. System and Attack Model

In this section, we introduce the system model and attack model of malicious energy attack. Specifically, we consider a green IoT network that uses RF energy harvesting as the energy source. In the attack model, we will introduce how the attacker implements the energy attack.

A. System Model

An example of an RF energy harvesting powered IoT network is shown in Fig. 1. Each EHN harvests ambient RF energy from the air and stores the harvested energy for future computation and communication usage. Since the ambient RF energy density is very thin, EHNs usually maintain a low energy level. We use uniformly distributed random variables to represent the fluctuations in ambient RF energy density over time. The average harvesting rates are also slightly different among EHNs, indicating the heterogeneity of the energy harvesting process in the network. We use Poisson–distributed random variables to represent dynamic network traffic. Due to the randomness of the energy consumption caused by the dynamic network traffic, and the randomness of energy harvesting, the battery levels of EHNs are highly dynamic, which further leads to changes in path selections.

We assume that E-WME routing is implemented at the Network layer. It selects the routing paths based on the energy harvesting rate of the node, the energy level of EHNs, and the hop number. The EHN with a higher harvest rate, more energy, and smaller hop number will have a lower forwarding cost. The path with the smallest cumulative forwarding cost will be selected to relay the traffic from the source to the destination. When there are no external energy sources (i.e., energy attackers), all EHNs in the network have a comparable energy harvest rate, so paths with fewer hops and shorter distances tend to be selected as the preferred forwarding route.

Each node in the green IoT network can work as a sender or receiver. But we assume that the attacker is only interested in the data traffic between a specific pair of source and destination nodes. The source node continuously sends data packets to its destination.

B. Attack Model

Considering that the energy attack is a secondary attack, we use eavesdropping as an example primary attack. Assume there exists a compromised node that performs eavesdropping attack and is interested in the network traffic from the source to the destination. The objective of energy attack is to attract targeted data traffic to the compromised node to improve the efficiency of eavesdropping.

The performance of the malicious energy attack is evaluated by the additional traffic attracted to the compromised node. It heavily depends on which nodes will be attacked and the energy distribution of the EHNs. If the attacker knows the global network status (i.e., network topology, energy distribution, harvest rate, etc.), the optimal energy attack can be easily solved by the deterministic optimization algorithms [13]. However, this assumption is unrealistic. In this paper, we consider a more practical scenario, where MES does not know the location information of each node, but only knows the area of the green IoT network. The attacker evenly divides the green IoT network area into many small cells containing about 0–2 EHNs. We assume that cell division is optimal, and each cell contains only one EHN. In each energy attack, the MES, which is equipped with beamforming antennas, charges specific cells directionally by increasing the RF density of the cells.

In order to assist the energy attack, we assume that some observer nodes are deployed by the attacker to monitor and record the traffic traveling through the local spied area. This information is used to construct the *state* space in RL-based energy attack and will be reported to the MES regularly. Since the attacker does not know the locations of the source and destination nodes, we assume that the attacker cannot find the optimal locations for the observer nodes. Therefore, we evenly place three observers around the compromised node in order to cover a large spied area. We assume the observer nodes have a larger communication range than ordinary EHNs. Therefore, the observer nodes can form a multi-hop link for reporting data to MES. The monitoring coverage of observers is set to 70% of the transmission range of EHNs.

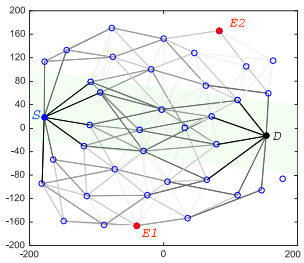
In addition, the energy harvest rate information is also valuable information to the attacker. We assume the attacker will manually measure the average RF energy density of each cell in the initial stage and then use this value as the estimated harvesting rate to construct the *state*. Even though the harvesting rate varies over time, the average harvesting rate is sufficient to perform intelligent energy attack. The compromised node will count the amount of traffic it receives/overhears and report this information (i.e., *reward* in RL) to the MES.

The reinforcement learning algorithm is implemented on MES to select the best nodes to attack in order to maximize the amount of traffic lured to the compromised node. We have developed two algorithms, namely, Q-IEA and PPG-IEA. The implementation details will be discussed in Section V.

IV. Glance at Energy Attack

Before diving into IEA, we conduct a preliminary experiment to investigate the impact of the malicious energy attack and also investigate the potential of developing IEA in an unknown network environment. Here, we simulate a small green IoT network containing 39 EHNs deployed in a 400m×400m area. The data generated by the source node, S, is sent to the destination node, D, as shown in Fig. 3(a).

Due to the energy and the size constraints of EHN, the transmission range of each node is relatively short, and the data is usually forwarded through multiple hops. Let E1 and E2 be two compromised nodes located at the edge of the network. At the network layer, we implement the E-WME routing method as we explained in Section II-B. We draw the routing paths without energy attack from node S to destination D in Fig. 3(a). The color of the path indicates the distribution of the traffic. The darker the path is, the more the traffic goes through. Although different paths can be selected for data delivery, due to fewer hop counts, most of the traffic passes through several main paths within the green band in Fig. 3(a). Consequently, in an ordinary network without attack, only 5.95% and 2.43% of generated target data packets travel through two compromised nodes, E1 and E2, respectively as shown in Fig. 3(b). This shows that in the ordinary network, E1 and E2 have very low possibility to capture the interested traffic since they are significantly deviated from the main path.

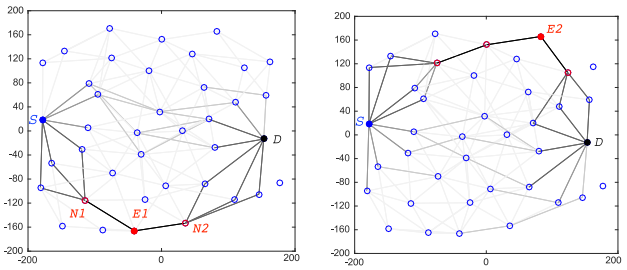


(a) Network without attack

Scenario \ Node	E1	E2
Ordinary	5.95%	2.43%
Scenario 1	54.98%	—
Scenario 2	—	54.01%

(b) Data traffic in E1/E2

Fig. 3. Network traffic distribution from S to D; the green band shows the main path; E1 and E2 are two compromised nodes.



(a) Scenario 1 (b) Scenario 2

Fig. 4. The impact of the malicious energy attack on data path selection in different scenarios.

However, the situation will immediately change once the attacker starts to assist the compromised nodes. As demonstrated in Fig. 4(a), when the MES intentionally provides extra energy to node N1 and N2, the path N1→ E1→ N2 is selected as a preferred route from node T to destination S. By comparing Fig. 4(a) and Fig. 3(a), we can observe dramatic changes in the traffic distribution caused by the malicious energy attack. Over 54.98% of target data goes through the compromised node, E1, almost ten times higher than that in the ordinary network without energy attack. Similar results can be observed in scenario 2, where the compromised node, E2, is further deviated from the main path than E1. In this case, we let MES simultaneously charge the three nodes marked by the red circle in Fig. 4(b). With the energy attack, over 54% of targeted traffic goes through E2, which is 22 times higher than that of the ordinary network.

These test results verify that the malicious energy attack is very effective in manipulating the data path in the green IoT network. Especially in a small network like Fig. 3(a), attacking one or two nodes is sufficient to attract remarkable network traffic to the compromised node. In this case, the MES can use brute force attacks to easily figure out the optimal attack strategy, so intelligent attack strategies are not very necessary. But in practice, when a green IoT network contains a large number of nodes, attacking one EHN will hardly lead to a visible increment in rewards. The simple brute force attack will be very inefficient. This motivates us to develop Intelligent Energy Attack (IEA) policies to help MES wisely select the nodes to attack in a large green IoT network, which will be introduced in the next section.

V. Intelligent Energy Attack Policy

In light of the unknown network and routing information, there are two main challenges in the design of an optimal energy attack strategy. First of all, without knowing the network state information (e.g., traffic distribution, routing protocols, and energy level of EHNs), it is difficult to mathematically model the relationship between the network status and the amount of traffic traveling through the compromised node. Second, the network environment is highly dynamic.

If MES attacks the green IoT network based on an instantaneous status of the network, the solution may be only optimal for a snapshot of the network. Therefore, how to adjust the attack pattern to accommodate the dynamics of the network is challenging. In order to tackle the above challenges, we propose two reinforcement-learning enabled attack methods, namely, Q-IEA and a PG-IEA. In this section, we will provide more details about the two proposed algorithms, including how to build the essential RL parameters and a thorough explanation of the neural network structure of the two algorithms.

A. Reinforcement Learning Parameters

One of the major challenges for IEA is how to construct the *state* space in light of the highly dynamic and unknown network status. A reasonable design of *state* space can greatly determine the performance of the agent. The ideal *state* information should contain the residual energy distribution of all nodes in the network. However, this information is private to each EHN and cannot be obtained by an attacker. Instead, we deployed some observer nodes that cooperate with MES. These nodes act as spy nodes to monitor the local traffic, then provide it to the attacker. The attacker already has the energy harvest information as we introduced in III-B. With the estimated harvest rate and traffic of spied nodes, we can roughly estimate the battery level[1] of nodes in the monitored areas of the spy nodes. The spy nodes will then calculate the average energy level of the spied area and report this value to MES. MES will use this information to construct a vector as the *state*, in which each element is the average energy of a spied area, as shown in Fig. 5.

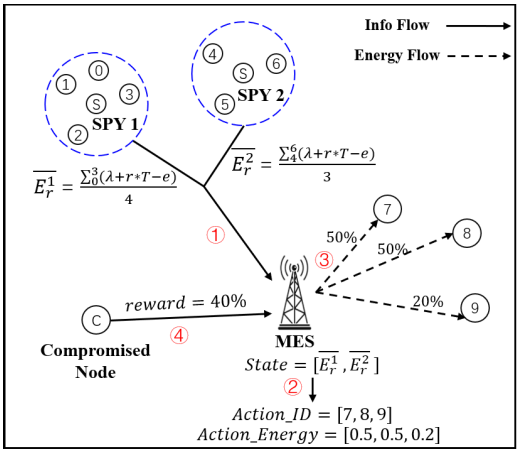


Fig. 5. IEA process.

In each period, the MES takes action a_t to lure the most target traffic to pass through the compromised node. The action of Q-IEA contains a list of nodes (e.g. Node 7, 8, 9) to be attacked. For PG-IEA, an action consists of two vectors, the list of nodes (e.g. Node 7, 8, 9 in Fig. 5) to be attacked and the amount of energy to be charged to correspond to nodes (e.g. 50%, 50%, 20% in Fig. 5). Then, the MES performs action to provide extra energy to the victim nodes. Finally, the compromised node calculates the amount of targeted data traffic travels through it (e.g. 40% in Fig. 5) as a reward and reports it to the MES. MES uses *states* and *reward* to update the model and then repeats this process until the end. We measure the amount of targeted traffic lured through the compromised node as the reward, r_t. In each period, the reward information is reported to the MES. The agent aims to find the best policy that maps the state to the most appropriate action, thereby maximizing the total reward.

B. Construction of IEA

Considering that Q-learning only deals with discrete states, but the energy level is a continuous variable, in the Q-IEA implementation, the energy levels from 0 to 100% are uniformly discretized into levels 0 to 9. In each episode, Q-IEA selects k number of victim EHNs to attack under a specific *state*. To charge the selected victim EHNs, we use a simplified on-off attack policy instead of training a new model to fine-tune the charging energy. In other words, the battery will be charged to nearly full when the nodes are under energy attack.[2] That is because Q-IEA is a greedy algorithm, it cannot find a policy for better balances energy efficiency and performance. Instead, it will always fully charge the victim EHNs to maximize the reward. Therefore, the output of Q-IEA is a vector of size k that represents the k number of victim EHNs to be fully charged.

[1] Although at the beginning, there will be errors in the estimations of battery level due to the heterogenous initial energy among different nodes. But as the experiment goes, the error will gradually decrease, which converges to $\leq 5\%$ in our simulation.

[2] Due to the nonlinear charging feature of the battery, it will take an infinite time to charge the EHN to full battery. In this paper, we assume the EHN is charged to over 90% when it is attacked in Q-IEA.

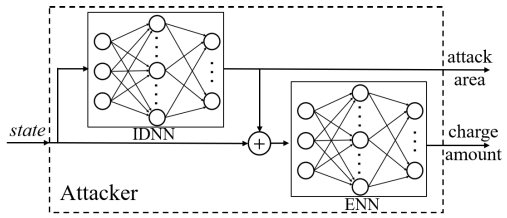
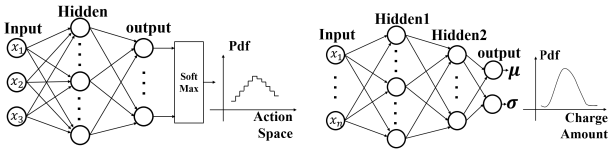


Fig. 6. PG-IEA Attacker Model Structure.



(a) Structure of IDNN (b) Structure of ENN

Fig. 7. Neural Network Structure used in PG-IEA.

The PG-IEA is composed of two independent decision-makers, one is used to select victim EHNs and the other is for deciding how much energy to charge, as shown in Fig. 6. In PG-IEA, the node ID selection neural network (IDNN) is used to select the k number of victim EHNs to attack based on the $state$ observed from the green IoT network. The output of IDNN together with $state$ is the input of the energy level adjustment neural network (ENN), which is used to determine the amount of charging energy.

The neural network structure of IDNN is depicted in Fig. 7(a). It has a hidden layer with 10 neurons and uses the $tanh$ function as the activation function. The number of neurons in the output layer is equal to the total number of candidate victim EHN. By using a softmax function, it converts the output of IDNN to a probability distribution of all candidate victim EHNs during the attack. It then samples from this distribution to select k number of victim EHNs. The output of IDNN is a vector of size k, which contains only 1 or 0 to indicate the attack state. In the training phase of IDNN, it uses the same on-off attack in Q-IEA to find the most appropriate policy.

ENN is trained to determine how much energy it should charge as shown in Fig. 7(b). The input of ENN is a combination of $state$ observed from the green IoT network and the output vector of the IDNN. ENN has two hidden layers with 20 neurons and 10 neurons respectively and uses Relu as the activation function. The output contains the mean value, μ, and variance, σ, which are used to generate a uniform distribution of 0-100%. Based on distribution, it randomly samples a value as the amount of energy that the attacker will charge the selected victim EHNs.

Usually, an exploration mechanism is needed in RL to avoid the agent falling into a local optimum. Since Q-IEA is a greedy algorithm, we set a decaying exploration rate to allow Q-IEA to explore more experience and converge to the optimal policy. In PG-IEA, the IDNN and ENN use the Adam optimizer function to update the neural network. The Adam optimizer can automatically adjust the learning rate to ensure the stability of the learning process. Also, PG-IEA itself is a combination of exploration and exploitation. Therefore, there is no need to set up an exploration mechanism for PG-IEA.

C. Pre-Train Process

In a large green IoT network, charging most nodes, especially the ones far away from the compromised node, does not help to manipulate traffic through the compromised node. We call these EHNs invalid nodes. Since the size of the IEA structure is related to the number of candidate victim EHNs, as we discussed in Section V-B, a large number of invalid nodes in the action space leads to inefficient and lengthy training of the IEA. For this reason, we add a Pre-Train stage before training the IEA agent to reduce the action space and improve the training efficiency of IEA. In each Pre-Train attack period, we attack one EHN and record the cumulative reward of the EHN, which is then used as an indicator to eliminate invalid EHNs.

Aiming at efficiently exploring the action space and exploiting the learned knowledge, we test three Pre-Train algorithms, namely 1) Q-IEA, 2) One-step Q-learning, and 3) Random Attack. Q-IEA is the same as the algorithm we introduced in Section V. One-step Q-learning is a simplified stateless Q-learning [32]. Different from the time-varying s_t in (2), One-step Q-learning only has one global $state$, s, and Q table is updated according to (8). Another difference is that there is no decay parameter γ in the One-step Q-learning compared to typical Q-learning. In Q-IEA and One-step Q-learning, the actions are selected based on the $decay\ \epsilon\text{–}greedy$ policy. In Random Attack, it chooses a random EHN to attack in each attack period without learning anything.

$$Q(s, a_t) = (1 - \alpha)Q(s, a_t) + \alpha R(s, a_t) \tag{8}$$

In summary, Q-IEA and One-step Q-learning have the same action selection policy, but the learning process is different. Due to the larger Q-table, the Q-IEA will perform worse than One-step Q-learning. Because the goal of Pre-Train is to quickly eliminate invalid nodes instead of high performance in every $state$. So a large Q-table may lead to lower training efficiency in the perspective of Pre-Train. One-step Q-learning and Random Attack are only different in the action selection. One-step Q-learning will better exploit the learned knowledge than Random Attack which in turn leads to more efficient training. In Section VI, we will conduct an evaluation to verify our prediction and select the most appropriate Pre-Train algorithm and the best settings.

Given the cumulative reward, a simple way to select valid nodes is to sort the EHNs according to the cumulative reward, and then select a fixed number of nodes with the highest reward. But this is unreasonable since the number of valid nodes varies with the location of the source node and destination node. Since the network topology is unknown to the attacker, it is difficult to know how many valid EHNs we should select. In this paper, we use the mean value of the cumulative reward as the threshold. Only nodes whose cumulative reward is higher than the threshold will be selected as valid nodes.

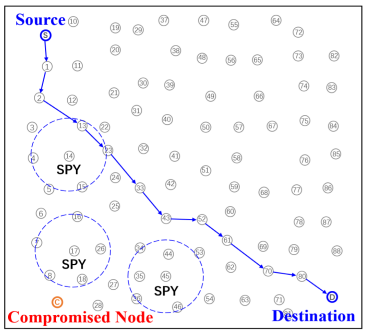


Fig. 8. An example of a large green IoT network with 90 EHNs.

VI. Simulation And Analysis

In this section, we evaluate the performance of the proposed IEA algorithm and compare the performance of IEA with benchmark solutions. Beyond that, we also evaluate the performance from different aspects to demonstrate the effectiveness and the efficiency of the proposed IEA algorithm.

A. Simulation Settings

Precision agriculture can be a potential application of the green IoT network. By deploying wireless energy harvesting nodes (EHNs) with sensors in farmland, people can monitor the state of the soil (e.g., the moisture, temperature, and pH value) in a large area for a long time [33]. In the simulations, we generate a large green IoT network with 90 EHNs deployed in 700 meters by 700 meters area, as shown in Fig. 8. The average distance between adjacent nodes is 70 meters, while the maximum transmission range of each node is 100 meters. In the test, each EHN generates data packets following the Poisson distribution with a mean value $\lambda = 0.2$ packets per slot. We assume the compromised node in the bottom left corner is interested in the target traffic from the source node at the top left corner to the destination node at the bottom right corner. The source node generates 0.5 data packets per slot and sends them to the destination node. The blue arrow lines in the graph constitute the preferred main path in the ordinary network without malicious energy attack. The green IoT network is built based on Python wsnsimpy, a dedicated simulator for wireless sensor networks. We modified the package accordingly to fully support energy harvesting and malicious energy attack.

The EHN in the network has a heterogenous average energy harvest rate, r_n, which follows a uniform distribution. On average, it takes 75 slots to charge the battery (i.e., supercapacitor) to nearly full.[3] Once fully charged, it can send roughly 17 packets before the battery reaches the low-energy threshold. The EHN in low-energy mode remains in inactive mode and avoids forwarding packets for neighboring nodes. Due to the randomness of the network traffic, the battery level of EHNs is highly dynamic, which further results in different path selections from the source node to the destination node. In addition, to simulate the dynamics of the radio environment, the harvest rate and Poisson distribution will slightly change over time. For every 100 slots, we re-generate the Poisson distribution of each node and let the harvest rate fluctuate 1% around its initial value.

We suppose the MES selects five victim EHNs to attack and only charges once at the beginning of each attack period. We set 20 slots as an observation episode. Three spy nodes are evenly placed around the compromised node. The spied area range is 70 meters, which is marked by the blue circle in Fig. 8. At the end of each period, the spy node uses the average value of the estimated energy of all monitored nodes as the average energy of the spied area.

In our simulation, we choose the random attack as the baseline. Random attack is an inefficient but simple attack method. In each attack period, it selects random nodes to attack without learning anything. As discussed earlier, if the routing information and the instant energy level of each EHN are known, the optimal energy attack can be easily solved using deterministic optimization algorithms. Although the optimal energy attack is infeasible to implement, we consider it as the upper bound of malicious energy attack. In the performance evaluation, we compare the performance of all proposed IEA algorithms with benchmark algorithms under different settings. In each period, the Upper Bound method will select the top five nodes with the lowest residual battery level to attack.

We construct the neural network on Tensorflow 2.2.0. The total training time for Q-IEA and PG-IEA is 8K and 10K episodes respectively. The performance of Q-IEA and PG-IEA is tested on Nvidia Jetson NANO 2GB dev-kit board without using GPU. The evaluation results will be discussed in the following sections.

B. Performance Evaluation for Pre-Train

In this section, we compare the performance of different Pre-Train algorithms to select the most appropriate one for IEA. The impact of training time, exploration rate, and learning rate on Pre-Train is analyzed. Each result is the average of ten independent tests. We evaluate the performance of Pre-Train in terms of accuracy and the number of valid nodes it extracts. The accuracy is calculated as the ratio of the number of correctly selected valid EHNs to the total number of claimed valid EHNs. It measures the ability of each algorithm to accurately select valid EHNs.

1) Training Time: In this test, We evaluate the performance of three Pre-Train algorithms with different training times, and the results are presented in Fig. 9 and Fig. 10

As shown in Fig. 9, the accuracy of the three algorithms increases as the training time increases. Because Pre-Train needs enough time for both exploration and learning process to obtain better accuracy. Since at the beginning of the Pre-Train, all methods are in the exploration stage. As the training episodes increases, the learning experience will gradually dominate the action selection, and the interference caused by the randomness of the energy distribution of EHNs will be reduced. Victim EHNs that receive extra energy obtain a higher

[3]Note that, due to the nonlinearity of the battery, the actual amount of energy that can be captured by the EHN depends on the instant residual energy of the battery [15]. Therefore, the amount of harvested energy in each slot will not be constant but calculated based on Equation (6) of [15].

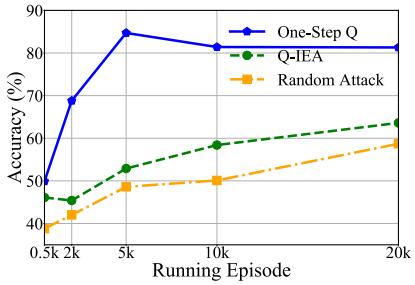


Fig. 9. Accuracy of different Pre-Train methods over training time.

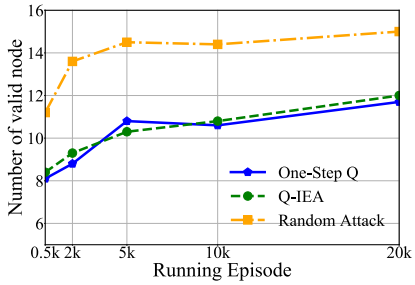


Fig. 10. Valid Node Number of different Pre-Train methods over training time.

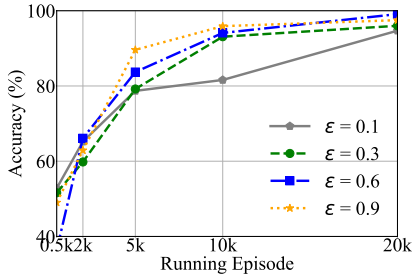


Fig. 11. Accuracy of One-step Q-learning with different exploration rate.

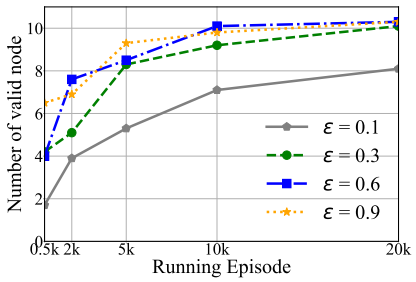


Fig. 12. Valid Node Number of One-step Q-learning with different exploration rate.

cumulative reward and a better accuracy is obtained as the training time increases.

Benefiting from their efficient exploration scheme, both One-Step Q-learning and Q-IEA achieve better accuracy than the Random method. As a result, EHNs that have higher rewards have more chances to be visited in One-Step Q-learning and Q-IEA than in Random Attack. And One-Step Q-learning is further superior to Q-IEA in the perspective of extraction accuracy. Without the time-varying state s_t, One-Step Q-learning has a much smaller Q-table than Q-IEA, which improves the training efficiency as discussed in Section V-C. So as the training goes, it will focus on the valid nodes earlier than Q-IEA since the latter method needs more time to explore all the states. To achieve a comparable accuracy, Q-IEA will require a much longer training time than One-Step Q-learning.

Note that Random method achieves higher performance in term of valid node number as shown in Fig. 10. The Random Attack method can generally extract more valid nodes than the other two methods at the expense of lower accuracy. This result is intuitive, because all EHNs have the same chance to be selected in the Random Attack. This has two benefits for extracting more valid nodes. First, the rewards of all EHNs are more flattened in Random Attack. As a result, there will be more number of EHNs whose rewards exceeds the average reward. Second, valid EHNs are less likely to be missed. Generally, all nodes along the *source→compromised node→destination* path are valid nodes. However, in a large green IoT network (e.g., 90 nodes in our test), it is common that attacking one valid EHN, especially the one that is far away from the compromised node, will not obtain obvious excessive rewards. In Q-IEA and One-Step Q-learning, EHNs with any tiny reward have a chance to be captured. In Random Attack, by contrast, these valid EHNs have a high chance of being missed.

Note that more valid nodes come at the expense of the much lower accuracy of Random Attack. Due to the random energy distribution of EHNs in the large green IoT network, attacking an invalid EHN may also cause a positive reward when the *source→compromised node→destination* path happens to be the preferred path. By comparing the three Pre-Train methods, we select One-Step Q-learning with more than 5k training episodes for Q-IEA and PG-IEA. In the rest of the Pre-Train evaluation, we use One-Step Q-learning as the default Pre-Train algorithm to find its best settings.

2) Exploration Rate: In this test, we evaluate the impact of exploration rate on One-Step Q-learning and present the result in Fig. 11 and Fig. 12. In the training stage, the exploration rate allows the agent to explore as many new actions as possible to avoid missing valid nodes. As training time increases, the extraction result becomes more accurate. When the exploration rate is too small, $\epsilon = 0.1$, the accuracy and the number of selected valid nodes are low, since the agent is stuck at attacking the invalid EHNs that accidentally have positive rewards in the early stage. When ϵ is greater than 0.3, the accuracy and number of selected valid EHNs have no obvious difference. Therefore, in the rest of the tests, we use $\epsilon = 0.6$ as the default setting of the One-Step Q-learning Pre-Train algorithm.

3) Learning Rate: According to the Q-table updates equation showing in (8), the learning rate, α is the only parameter that affects the learning process in One-Step Q-learning. To evaluate how the learning rate impacts the Pre-Train result, we apply different learning rates on One-Step Q-learning and demonstrate the results in Fig. 13 and Fig. 14.

It can be seen from Fig. 13, the accuracy decreases as α increases. A small α leads to a higher accuracy at the cost

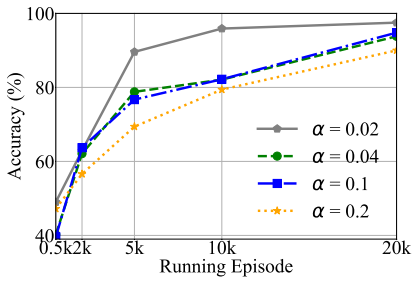


Fig. 13. Accuracy of One-step Q-learning with different learning rate.

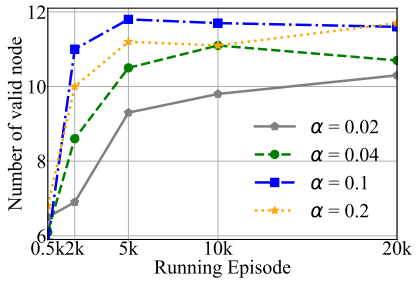


Fig. 14. Valid Node Number of One-step Q-learning with different learning rate.

of less number of selected valid nodes. According to (8), a low learning rate means the algorithm will more rely on learned experience. The interference of positive reward in the current period resulted from the accidentally high energy along the *source→compromised node→destination* path will be suppressed. The learning process will become slow but stable. And vice versa, the learning process will become aggressive but unstable when the learning rate is too large. However, the difference in accuracy can be neglected when the training episode is larger than 20k.

As shown in Fig. 14, the number of valid nodes with $\alpha = 0.02$ is less than others settings, as the learning is much slower and more episodes are needed to find a comparable number of valid nodes with other settings. When α is larger than 0.1, no obvious difference is observed in terms of accuracy and the number of valid nodes. In order to balance the accuracy and the number of selected valid nodes, we set $\alpha = 0.1$ and training time to 20k episodes in One-step Q-learning Pre-Train.

C. Performance Evaluation for IEA

In this subsection, we evaluate the performance of IEA and analyze the impact of the number of nodes attacked, the traffic rate, and Pre-Train accuracy on the performance of Random Attack, Q-IEA, and PG-IEA methods. Random Attack uses the Random Pre-Train to reduce the action space, while Q-IEA and PG-IEA adopt One-step Q-learning Pre-Train. Each result is the average performance of 30 independent tests.

1) Performance Comparisons: In order to evaluate the effectiveness of the different IEA methods, we run the random attack, Q-IEA, and PG-IEA with the same simulation setting and present their performances in Fig. 15. The reward (i.e., amount of targeted traffic captured by the compromised node) is normalized by the Upper Bound performance to

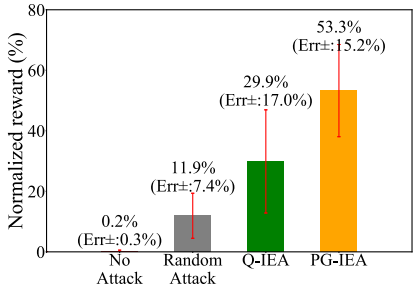


Fig. 15. IEA performance comparisons and improvements.

eliminate the impact of network randomness (e.g., randomness in energy harvesting rate and network traffic) on the attack. In the Upper Bound attack, we assume the MES knows the global and instantaneous network states and can choose the five optimal EHNs to maximize the attack performance. In a large network shown in Fig. 8, attacking five EHNs is not sufficient to attract all the targeted traffic through the compromised node. On average, Upper Bound manipulate 31.1% of total targeted traffic in our tests.

In the ordinary network without energy attack, the compromised node only capture 0.2% of normalized traffic since it largely deviates from the main *source→destination* path (i.e., blue arrow lines in Fig. 8). For Random Attack with the reduced action space from Pre-Train, an additional 11.7% of normalized reward is obtained. With the assistance of Q-learning, the Q-IEA achieves over two times higher performance than the Random Attack, which verifies the effectiveness of Q-IEA. The PG-IEA further improves attack performance to 53.3%. Different from the Upper Bound attack that knows global and instantaneous network states, Q-IEA and PG-IEA only have the average energy of the monitored nodes in the spied area. The imperfect state information and limited knowledge of the global network account for the lower performance of Q-IEA and PG-IEA than the Upper Bound attack.

As a greedy algorithm, Q-IEA cannot handle the stochastic optimal policy problem, that is, there may exist multiple optimal *actions* with probability under a certain *state*. In contrast, the goal of Policy Gradient is to find the best attack policy in each *state*. During the stochastic learning process, PG-IEA can find out a couple of actions with higher rewards under each *state*. After complete training, PG-IEA will output an action probability distribution by feeding a particular *state*. The higher reward an action can obtain, the higher probability it will be selected. By addressing the non-unique mapping relation issue, PG-IEA achieves 23.4% more normalized reward than Q-IEA on average.

In addition to performance of attack efficiency, we have also compared the energy efficiency of different attack methods in Fig. 16. Due to the nonlinear battery charging feature, the charging efficiency drops significantly after the battery level is higher than 20%, as illustrated in Fig. 2(b). Therefore, the most energy-efficient energy attack should charge the EHNs to a critical level so that it is just enough to lure the targeted traffic to the compromised node in one attack period, and providing more power to these EHNs will be a waste of energy.

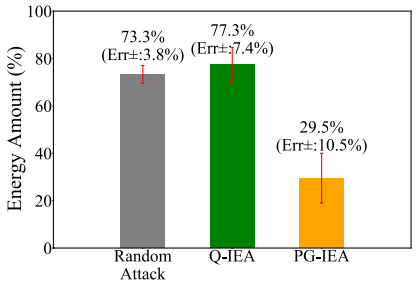


Fig. 16. IEA energy efficiency comparisons.

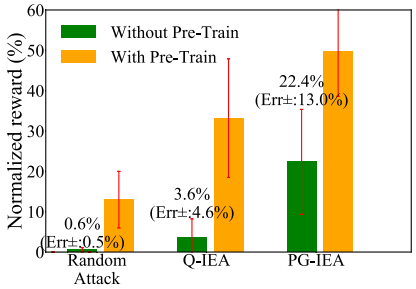


Fig. 17. Performance of IEA with and without Pre-Train.

Both Random Attack and Q-IEA have no control on how much energy to charge and use on/off policy. When the attack is on, the selected EHN will be provided excessive energy and charged to a very high power to guarantee the success of the energy attack. For this reason, we observe significant energy waste in Q-IEA and Random Attack algorithms. As shown in Fig. 16, their energy wastes are 73.3% and 77.3%, respectively. By contrast, the PG-IEA adopts an ENN to decide how much energy should be provided to each selected EHN, which remarkably reduces the energy waste in PG-IEA. However, due to the imperfect state information and limited knowledge of the global network, PG-IEA still has 29.5% of the energy waste compared to the optimal attack strategy. Fig. 15 and Fig. 16 demonstrate that PG-IEA significantly outperforms Random Attack and Q-IEA in terms of attack efficiency and energy efficiency.

2) Resource and Computation Cost Comparison: Even though the MES is not severely constrained by resources, resource utilization is still an important metric to evaluate the performance of Q-IEA and PG-IEA. Therefore, we conduct a comparative analysis on resource demands and computational costs of two IEA algorithms.

We use the time spent on optimal action decisions as the computation cost. In each episode, Q-IEA and PG-IEA consume 0.79ms and 6.21ms, respectively, to calculate an optimal action. Due to the complexity of the neural network, the computation cost of PG-IEA is 7.86 times higher than Q-IEA. The gap will expand to 12.27 times during the training since the backpropagation computation of the neural network is even more complex. To be noticed, the total training time for PG-IEA is 20K episodes which needs more time to train the model compared to 8K of Q-IEA. Beyond that, the demanding RAM for Q-IEA and PG-IEA to implement attacks is 18 MB and 162.8 MB, respectively. And the storage consumption for Q-IEA and PG-IEA is 7.47 KB and 190 KB, respectively. The communication overhead is the same to both algorithms since the information they interact with the IoT network is the same.

To summarize, Q-IEA requires much fewer resources and can be implemented on an STM32 dev-board [34]. On the contrary, PG-IEA requires more hardware and computation resources, so it must be implemented on a monoboard microcomputer (e.g. Raspberry pi, Nvidia Jetson series). Therefore, Q-IEA is recommended to be implemented when the attacker has very limited resources. PG-IEA is advocated for higher performance when the computation resources are sufficient.

3) Impact of Pre-Train: As discussed in V-C, Pre-Train can greatly improve the attack performance by eliminating invalid nodes. So as a critical part of IEA, we demonstrate two tests to further analyze the impact of Pre-Train on IEA performance, and the results are shown in Fig. 17 and Fig. 18, respectively.

In the first test, we remove Pre-Train from IEA, which means all IEA algorithms' action space contains all EHNs in the network. The performance comparison with and without Pre-Train are present in Fig. 17. The reward is normalized by Upper Bound to eliminate the impact of network randomness. From Fig. 17, the performance of all attack algorithms is significantly reduced. The Random Attack only obtains 0.6% of normalized traffic, that is 95.4% performance lost without Pre-Train. With the help of Q-learning, Q-IEA still achieves higher performance than the Random Attack. But Q-IEA only lures 3.6% of the normalized traffic, the performance dropped 90% without Pre-Train. However, the PG-IEA still performs better than other algorithms although it only obtains 22.4% of the normalized traffic as shown in Fig. 17. The performance of all algorithms is decreased significantly without the Pre-Train to reduce action space.

As discussed in II-C.1, Q-learning needs extensive valid attacks to guarantee the optimal action under every *state*. Without Pre-Train, the data efficiency is decreased since most attacks is invalid. Thus, the Q-IEA is struggling to find the optimal action under every *state* with so rare useful experience. Different from Q-IEA, PG-IEA can utilize such rare experience to directly update the entire neural network that affects policy under every *state*. Less data is required to achieve comparable performance to Q-IEA. Therefore, the gap between Q-IEA and PG-IEA is expanded by removing Pre-Train.

In addition to evaluating the impact of the existence of Pre-Train, another test is demonstrated to evaluate the impact of the accuracy of Pre-Train. We apply Pre-Train with different accuracy and present the result normalized by Upper Bound in Fig. 18. The Random Attack is removed from this test because the accuracy of the Random Attack Pre-Train algorithm remains consistent. As shown in Fig. 18, the performance of Q-IEA and PG-IEA gradually improves as the Pre-Train accuracy increases. With a more accurate Pre-Train result, the data efficiency will increase during the training since

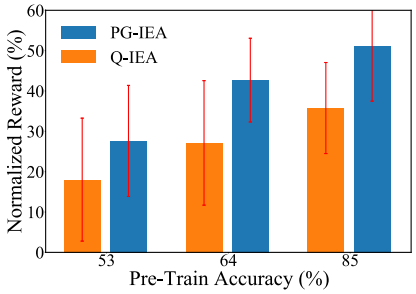


Fig. 18. Impact of Different Pre-Train Result.

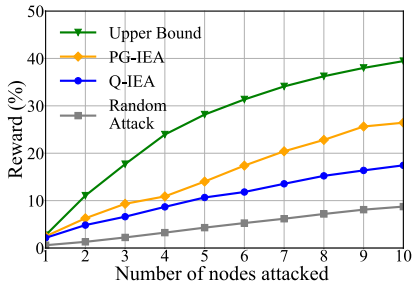


Fig. 19. Impact of the number of attacked nodes on energy attack.

more invalid nodes are eliminated. Fig. 18 demonstrates that the high accuracy of Pre-Train will improve the performance of both Q-IEA and PG-IEA, and PG-IEA always outperforms Q-IEA.

4) Number of Nodes Attacked: Intuitively speaking, the more nodes that are attacked (i.e., a larger k), the greater impact of the MES on the green IoT network, which can potentially bring more data traffic. Especially if all nodes along the *source→compromised node→destination* path are charged to nearly full, the most majority of the traffic can be lured to the compromised node. But it is neither practical nor efficient in reality. So to evaluate the relation between the number of nodes attacked, performance, and energy efficiency. We evaluate the performance of IEA with reduced action space by attacking the different number of nodes, and present the result in Fig. 19.

The simulation results confirm that with more nodes attacked, the MES gains stronger forces to lure the targeted traffic to the compromised node. From the Upper Bound curve, 30% of total traffic is encouraged to the compromised node with only five nodes attacked. But it only attracts 10% more traffic with another four nodes attacked. The Q-IEA curve also verifies that the performance improvement brought by charging more nodes significantly reduces when k is large. For this reason, we suggest attacking 3 to 5 nodes in the given green IoT network setting to balance performance between reward and energy efficiency. In this paper, MES attacks five victim EHNs in each episode. We also notice that no matter how many nodes are attacked, the performance of PG-IEA always outperforms the Q-IEA because of high data efficiency and appropriate policy searching strategy.

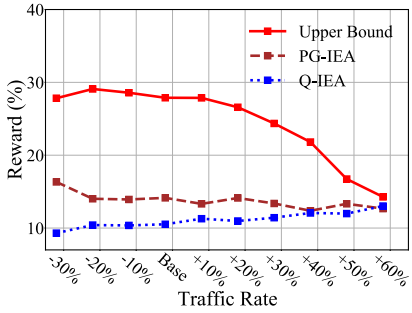


Fig. 20. Impact of the traffic rate on the performance of PG-IEA.

5) Main Path Traffic: To evaluate the adaptability of PG-IEA, we investigate the impact of traffic rate on the performance of proposed IEA algorithms. We set the default packet generation rate to 0.5 packets per slot and change the traffic rate based on the default value. The Upper Bound of 5 nodes attack and the performance of PG-IEA and Q-IEA are presented in Fig. 20. It shows that as the traffic rate increases, the performance of PG-IEA and Q-IEA is nearly the same. But on the contrary, the traffic lured to the compromised node by the optimal 5-node attack in Upper Bound decreases as the traffic rate grows.

As the traffic rate grows, the increased packet transmissions will drain the battery of EHNs in a faster manner. The nodes that are not being attacked will become the bottleneck in the energy attack. When the battery on those EHNs is drained, the *source→compromised node→destination* path is likely to become disconnected and the source node tends to switch to other routes where the compromised node is not involved. While other nodes along the *source→compromised node→destination* path become a severe limitation, attacking the top five nodes with the lowest residual battery level in Upper Bound may not be appropriate. The performance gap between the Upper Bound and two IEA algorithms is caused by more "wisely" selecting five nodes to attack as traffic increases.

Fig. 20 also shows that the gap between PG-IEA and Q-IEA decreases as the traffic rate increases. Especially when the traffic rate increases by more than 40%, the performance of the two IEA algorithms becomes the same. When the traffic rate is high, keeping the attacked node fully charged can maintain high performance at the cost of low energy efficiency especially when charged energy is more than an optimal threshold. But with the assistance of ENN, the PG-IEA will sacrifice some rewards to achieve a balance between rewards and energy efficiency as the traffic rate increase. Thus, PG-IEA only utilizes nearly half of the charged energy used by Q-IEA, and less than 10% of energy is wasted on average when the traffic rate increases by more than 40%.

VII. Conclusion

In this work, we introduced a new security issue in green IoT networks where an adversarial energy source can intentionally provide extra energy to specific nodes to manipulate the data path in the network layer. Malicious energy attack

is a brand new attack method in green IoT networks and is worth more investigations in the future. We have well-designed two reinforcement learning-enabled algorithm to implement the energy attack. The result shows that both algorithms outperform the Random Attack method.

However, the imperfect state design cannot accurately reflect the real network states and may lead to low energy efficiency during the attack. Because the current network state should be the superposition of the results of all historical actions. To alleviate the impact of this issue, we can utilize historical actions to improve the current state. Therefore, the requirements for memory ability are put forward. In future work, we will use Long Short-Term Memory (LSTM) neural network instead to further revise our algorithm. Since LSTM can make the decision based on the current state and history. Beyond that, we will also consider developing an algorithm for optimal cell division and evaluate the impact of cell division on the final attack performance.

REFERENCES

[1] T. Wu, F. Wu, J.-M. Redouté, and M. R. Yuce, "An autonomous wireless body area network implementation towards IoT connected healthcare applications," *IEEE Access*, vol. 5, pp. 11413–11422, 2017.

[2] F. Akhtar and M. H. Rehmani, "Energy harvesting for self-sustainable wireless body area networks," *IT Prof.*, vol. 19, no. 2, pp. 32–40, Mar./Apr. 2017.

[3] D. Niyato, D. I. Kim, M. Maso, and Z. Han, "Wireless powered communication networks: Research directions and technological approaches," *IEEE Wireless Commun.*, vol. 24, no. 6, pp. 88–97, Dec. 2017.

[4] X. Chen, D. W. K. Ng, and H.-H. Chen, "Secrecy wireless information and power transfer: Challenges and opportunities," *IEEE Wireless Commun.*, vol. 23, no. 2, pp. 54–61, Apr. 2016.

[5] Q. Liu, K. S. Yildirim, P. Pawełczak, and M. Warnier, "Safe and secure wireless power transfer networks: Challenges and opportunities in RF-based systems," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 74–79, Sep. 2016.

[6] V. N. Vo, T. G. Nguyen, C. So-In, and D.-B. Ha, "Secrecy performance analysis of energy harvesting wireless sensor networks with a friendly jammer," *IEEE Access*, vol. 5, pp. 25196–25206, 2017.

[7] A. Di Mauro, D. Papini, and N. Dragoni, "Security challenges for energy-harvesting wireless sensor networks," in *Proc. PECCS*, 2012, pp. 422–425.

[8] V. Shakhov, S. Nam, and H. Choo, "Flooding attack in energy harvesting wireless sensor networks," in *Proc. 7th Int. Conf. Ubiquitous Inf. Manage. Commun.*, 2013, pp. 1–5.

[9] P. Tedeschi, S. Sciancalepore, and R. Di Pietro, "Security in energy harvesting networks: A survey of current solutions and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2658–2693, 4th Qaurt., 2020.

[10] L. Li, Y. Luo, and L. Pu, "Q-learning enabled intelligent energy attack in sustainable wireless communication networks," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2021, pp. 1–6.

[11] L. Lin, N. B. Shroff, and R. Srikant, "Asymptotically optimal energy-aware routing for multihop wireless networks with renewable energy sources," *IEEE/ACM Trans. Netw.*, vol. 15, no. 5, pp. 1021–1034, Oct. 2007.

[12] J. Guo, X. Zhou, and S. Durrani, "Wireless power transfer via mmWave power beacons with directional beamforming," *IEEE Commun. Lett.*, vol. 8, no. 1, pp. 17–20, Feb. 2019.

[13] G. Han, Y. Dong, H. Guo, L. Shu, and D. Wu, "Cross-layer optimized routing in wireless sensor networks with duty cycle and energy harvesting," *Wirel. Commun. Mobile Comput.*, vol. 15, no. 16, pp. 1957–1981, Nov. 2015.

[14] Y. Li, "Deep reinforcement learning: An overview," 2017, *arXiv:1701.07274*.

[15] Y. Luo, L. Pu, Y. Zhao, W. Wang, and Q. Yang, "A nonlinear recursive model based optimal transmission scheduling in RF energy harvesting wireless communications," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3449–3462, May 2020.

[16] T. Le, K. Mayaram, and T. Fiez, "Efficient far-field radio frequency energy harvesting for passively powered sensor networks," *IEEE J. Solid-State Circuits*, vol. 43, no. 5, pp. 1287–1302, May 2008.

[17] G. Papotto, F. Carrara, and G. Palmisano, "A 90-nm CMOS threshold-compensated RF energy harvester," *IEEE J. Solid-State Circuits*, vol. 46, no. 9, pp. 1985–1997, Sep. 2011.

[18] I. Chaour, A. Fakhfakh, and O. Kanoun, "Enhanced passive RF-DC converter circuit efficiency for low RF energy harvesting," *Sensors*, vol. 17, no. 3, p. 546, Mar. 2017.

[19] T. Umeda, H. Yoshida, S. Sekine, Y. Fujita, T. Suzuki, and S. Otaka, "A 950-MHz rectifier circuit for sensor network tags with 10-m distance," *IEEE J. Solid-State Circuits*, vol. 41, no. 1, pp. 35–41, Jan. 2006.

[20] Y. Dong, J. Wang, B. Shim, and D. I. Kim, "DEARER: A distance-and-energy-aware routing with energy reservation for energy harvesting wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3798–3813, Dec. 2016.

[21] A. Kansal, J. Hsu, M. Srivastava, and V. Raghunathan, "Harvesting aware power management for sensor networks," in *Proc. 43rd Annu. Conf. Design Autom.*, 2006, pp. 651–656.

[22] Z. Li, Y. Peng, W. Zhang, and D. Qiao, "J-RoC: A joint routing and charging scheme to prolong sensor network lifetime," in *Proc. 19th IEEE Int. Conf. Netw. Protocols*, Oct. 2011, pp. 373–382.

[23] A. H. Coarasa, P. Nintanavongsa, S. Sanyal, and K. R. Chowdhury, "Impact of mobile transmitter sources on radio frequency wireless energy harvesting," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Jan. 2013, pp. 573–577.

[24] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A survey on wireless security: Technical challenges, recent advances, and future trends," *Proc. IEEE*, vol. 104, no. 9, pp. 1727–1765, Sep. 2016.

[25] J. Tang, A. Liu, J. Zhang, N. Xiong, Z. Zeng, and T. Wang, "A trust-based secure routing scheme using the traceback approach for energy-harvesting wireless sensor networks," *Sensors*, vol. 18, no. 3, p. 751, Mar. 2018.

[26] T. Zhu, S. Xiao, Y. Ping, D. Towsley, and W. Gong, "A secure energy routing mechanism for sharing renewable energy in smart microgrid," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Oct. 2011, pp. 143–148.

[27] R. Nian, J. Liu, and B. Huang, "A review on reinforcement learning: Introduction and applications in industrial process control," *Comput. Chem. Eng.*, vol. 139, Aug. 2020, Art. no. 106886.

[28] Y. Yao and Z. Feng, "Centralized channel and power allocation for cognitive radio networks: A q-learning solution," in *Proc. Future Netw. Mobile Summit*, 2010, pp. 1–8.

[29] J. Yan, H. He, X. Zhong, and Y. Tang, "Q-Learning-Based vulnerability analysis of smart grid against sequential topology attacks," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 1, pp. 200–210, Jan. 2017.

[30] C. Qiu, Y. Hu, Y. Chen, and B. Zeng, "Deep deterministic policy gradient (DDPG)-based energy harvesting wireless communications," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8577–8588, Oct. 2019.

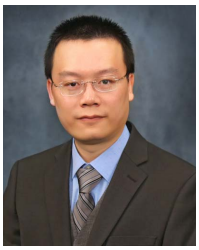
[31] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1992.

[32] K. Spiros and K. Daniel, "Reinforcement learning of coordination in cooperative multi-agent systems," in *Proc. 18th Nat. Conf. AI*, Edmonton, AB, Canada: ACM Press, 2002, pp. 326–331.

[33] Y. Luo and L. Pu, "WUR-TS: Semi-passive wake-up radio receiver based time synchronization method for energy harvesting wireless networks," *IEEE Trans. Mobile Comput.*, early access, Mar. 8, 2021, doi: 10.1109/TMC.2021.3064374.

[34] N. Lin, Y. Dong, and D. Lu, "Fast transparent virtual memory for complex data processing in sensor networks," *SENSORNETS*, vol. 24, p. 34, Feb. 2012.

Long Li received the B.S. degree in automation from Chang'an University, Xi'an, China, in 2017. He is currently pursuing the Ph.D. degree in computer science with The University of Alabama, Tuscaloosa. His research interests include RF energy harvesting wireless networks, reinforcement learning, and edge computing.



Yu Luo received the B.S. and M.S. degrees in electrical engineering from Northwestern Polytechnical University, China, in 2009 and 2012, respectively, and the Ph.D. degree in computer science and engineering from the University of Connecticut, Storrs, in 2015. He is currently an Assistant Professor with Mississippi State University. His research interests include the sustainable wireless networks for emerging the IoT, RF energy harvesting hardware, security in RF energy harvesting wireless networks, and underwater wireless networks. He was a co-recipient of the Best Paper Award in IFIP Networking in 2013 and Chinacom in 2016.

Jing Yang received the B.S. degree in mechanical engineering from the Lanzhou University of Technology, Lanzhou, China, in 2015, and the M.S. degree in mechatronic engineering from the Hefei University of Technology, Hefei, China, in 2018. She is currently pursuing the Ph.D. degree in electrical and computer engineering from Mississippi State University, Starkville. Her research interests include energy harvesting wireless sensor networks and the Internet of Things.

Lina Pu (Member, IEEE) received the B.S. degree in electrical engineering from Northwestern Polytechnical University, Xi'an, China, in 2009, and the Ph.D. degree in computer science and engineering from the University of Connecticut, Storrs. She is currently an Assistant Professor with The University of Alabama. Her research interests include edge computing, RF energy harvesting wireless networks, security in the sustainable IoT, and underwater acoustic/VL networks. She owned IFIP Networking 2013 Best Paper Award.