FOLPETTI: A Novel Multi-Armed Bandit Smart Attack for Wireless Networks

Emilie Bout, Alessandro Brighente, Mauro Conti, Valeria Loscrì

FOLPETTI: A Novel Multi-Armed Bandit Smart Attack for Wireless Networks

Emilie Bout
Inria, Lille, France
emilie.bout@inria.fr

Alessandro Brighente
University of Padova, Italy
alessandro.brighente@unipd.it

Mauro Conti
University of Padova, Italy
mauro.conti@unipd.it

Loscri Valeria
FUN - Inria, Lille, France
valeria.loscri@inria.fr

ABSTRACT

Channel hopping provides a defense mechanism against jamming attacks in large scale Internet of Things (IoT) networks. However, a sufficiently powerful attacker may be able to learn the channel hopping pattern and efficiently predict the channel to jam.

In this paper, we present FOLPETTI, a Multi-Armed Bandit (MAB)-based attack to dynamically follow the victim's channel selection in real-time. Compared to previous attacks implemented via Deep Reinforcement Learning (DRL), FOLPETTI does not require recurrent training phases to capture the victim's behavior, allowing hence a continuous attack. We assess the validity of FOLPETTI by implementing it to launch a jamming attack. We evaluate its performance against a victim performing random channel selection and a victim implementing a MAB defence strategy. We assume that the victim detects an attack when more than 20% of the transmitted packets are not received, therefore this represents the limit for the attack to be stealthy. In this scenario, FOLPETTI achieves a 15% success rate for the victim's random channel selection strategy, close to the 17.5% obtained with a genie-aided approach.Conversely, the DRL-based approach reaches a success rate of 12.5%, which is 5.5% less than FOLPETTI. We also confirm the results by confronting FOLPETTI with a MAB based channel hopping method. Finally, we show that FOLPETTI creates an additional energy demand independently from its success rate, therefore decreasing the lifetime of IoT devices.

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**; • **Networks → Denial-of-service attacks**.

KEYWORDS

Internet of Things, Jamming, Channel hopping

1 INTRODUCTION

Guaranteeing security in IoT represents a challenging task due to the large number of connected devices and their largely distributed nature [10]. The attack surface of this technology is expected to further expand, due to the introduction of massive IoT networks [8]. Furthermore, these devices are mostly connected via wireless communications, whose openness represents an additional security challenge. A common strategy to mitigate attacks is to move the communication from the attacked channel to a free one. Each user is hence given a set of different channels to use for communication. Channel hopping is the process by means of which a user selects, upon detecting an attack or a communication failure, a new channel for transmission. This is particularly useful in attacks such as jamming, where a malicious user intentionally degrades the quality of the victim's channel preventing the successful delivery of packets [7]. In this case, the victim selects an alternative free channel according to a predefined strategy.

Most of the research on channel hopping focused on the design of strategies to optimally select the new channel. In particular, given different attacker capabilities, researchers focused on the design of the best channel hopping strategy to guarantee continuous service availability. Relevant examples include sequential channel hopping (i.e., move to the next available channel), random channel hopping (i.e., select a random channel from those available) [13], game theory approaches [17, 21], and machine learning approaches [9, 23, 27]. However, given the increasing availability of cheap smart and powerful devices, an attacker can design an optimal strategy too. In particular, given a certain defence strategy, the attacker can learn the victim's channel hopping pattern to predict the channel to jam. To this aim, an attacker might use DRL, where she first observes the victim's behavior, and then attacks according to the inferred channel hopping pattern [31]. In response, the victim selects a new channel hopping strategy to reduce the effectiveness of the attacker. In this case, the attacker needs to periodically interrupt the attack to learn the new victim's channel hopping pattern. The DRL strategy also relies on the assumption that, during the learning phase, the victim's channel hopping strategy remains static. Therefore, learning-based methods do not represent an efficient solution against channel hopping. Based on these observations, an effective attack strategy does not need to rely on a particular victim's behavior, and should provide attack continuity.

In this paper, we propose FOLPETTI, a novel attack strategy against channel hopping that guarantees the continuity of the attack. We exploit a MAB approach to resolve the exploit-explore dilemma at the attackers' side. Thanks to our online approach, we guarantee both the attack continuity and the independence from the assumption of a victim's static channel hopping strategy. As a relevant example, we test FOLPETTI in a jamming scenario considering two defence strategies: i) random channel hopping, and ii) the smart channel hopping strategy implemented in [22]. We compare FOLPETTI against three other attacking strategies: i) Random strategy, where the attacker randomly selects a channel to attack; ii) DRL strategy, where the attacker uses DRL to predict the channel to attack; and iii) Optimal strategy, i.e., a genie aided method. We

assume that the victim detects an attacker when the percentage of correctly delivered packets falls below 80%. This hence represents the limit that the attacker needs to achieve to be stealthy. Our results show that, in case of random channel hopping, FOLPETTI reduces the Packet Delivery Ratio (PDR) to 82%, close to the 80% PDR attained with a genie aided method (optimal solution). The FOLPETTI attack is continuous, therefore causing a higher impact on the network compared to that of the DRL-based attack, which needs to periodically stop and learn the best strategy. Indeed, we show that the DRL based attack reduces the PDR to 85%, i.e 5% less than FOLPETTI. Also when the transmitter adopts the smart channel hopping strategy, we demonstrate that FOLPETTI is able to reduce the PDR more significantly compared to other methods. Moreover, we prove that FOLPETTI significantly increases the number of retransmissions and consequently the energy demand of the victims.

The main contributions of the paper can be summarized as follows.

- We design FOLPETTI, a new smart attack strategy against channel hopping. FOLPETTI is based on a MAB approach able to perform an optimal channel selection.
- We integrate the FOLPETTI strategy with a jamming attack and demonstrate that the attacker is able to learn the victim's channel hopping strategy and attack online. This guarantees a continuous attack.
- We show that FOLPETTI increases the number of retransmissions and hence increases the nodes' energy demand, shortening device's lifetime.
- We develop a jamming module with the latest version of discrete event simulator NS-3 (Network Simulator-3). We add some essential needs for simulating modern jamming attacks, such as the machine learning-based approach.

The rest of the paper is organized as follows. In Section 2 we present the representative works on attacks against channel hopping. In Section 3 we detail the FOLPETTI strategy and how this approach can be combined with a jamming attack. Section 4 describes the approach used for detecting jamming attacks and the channel hopping strategy implemented after an attack detection. Section 5 provides an analysis of the behavior of the FOLPETTI strategy. Then, in Section 6 we compare the performance of FOLPETTI with those of other relevant attacks in two different scenarios. In Section 7 we provide some insights into the implementation of such a kind of attack in a wireless communication network. We discuss in Section 8 possible countermeasures to prevent FOLPETTI. Finally, we conclude the paper and provide some future research direction in Section 9.

2 RELATED LITERATURE

In this section we review the literature discussing channel hopping schemes. We first discuss the defence perspective, to gain insights into how victims select their hopping strategies. Then we discuss the attack perspective to clearly show the novelty of our attack strategy.

Defence Perspective. Channel hopping solutions range from the most naive (i.e., sequential channel hopping) to the most elaborated and smart (e.g., via Reinforcement Learning (RL)). We focus on the second class, and in particular on RL where devices can decide on the next channel based on the positive and negative feedback received from the network. The authors in [9] propose a framework where, in case of heavy jamming, a user might leave the network and connect to another access point. The RL framework exploits the past observations to decide whether to leave the network or select a new channel. A similar RL-based mobility concept has been exploited by the authors of [27] to decide whether a user should combat the jammer or leave. Both [9] and [27] used quantized Signal to Interference plus Noise Ratio (SINR) values as RL states, therefore being limited in real world scenarios. To deal with the problem of infinite SINR states, the authors in [15] proposed to use the spectrum waterfall as RL states. Furthermore, their solution does not require the knowledge of the jamming pattern, and extends to the case where the attacker implements an intelligent and dynamic jamming strategy. The authors in [26] proposed a multi-agent RL framework to account for both the jamming attacker and the mutual interference caused by multiple legitimate nodes. A similar concept has been proposed by the authors in [32] to cope with the increasing number of devices in ultra-dense networks. All these works however did not account for the power-limited capabilities of IoT devices. The authors in [28] proposed a Markov decision process-based RL framework, showing the power consumption of their framework against three types of jammers: sweep, random, and sensing-based. However, they did not consider the effects of an intelligent jamming strategy.

Attack Perspective: As demonstrated by the fervent literature described above, most of the literature focuses on channel hopping solutions for the mitigation. In this paper, we focus on the attacker side and we hence review available proposals for smart jamming strategies. The authors in [18] proposed a game-theoretic approach, where both the attacker and the victim choose the channel to hop solving for the optimal strategy. However, authors resolved to Q-learning to compute the solution of the game, hence requiring periodical learning phase. The authors in [31] proposed a deep RL framework, where the attacker observes the victim activity to learn the activity pattern. The attacker employs an actor-critic neural network model, where the actor tests the channel and reports result to the critic which decides on the next action. Authors assume that the victim pattern is static during the learning phase. An extension to simultaneously attack multiple channels has been proposed in [25], where the authors consider multiple dynamic channels. Their approach is based on a Deep Q-Network (DQN) approach with the main objective to reduce the sum-rate of the victim node. In order to implement an effective attack, the authors foresee a listening phase where the attack is in stand-by.

Unlike the previous attacks, our solutions is able to attack in a continuous fashion, without requiring a listening phases. We hence provide a novel and efficient attack strategy.

3 METHODOLOGY

In this section, we describe our attack methodology. We first describe FOLPETTI, our novel attack strategy in Section 3.1. Then, as a relevant example, we show its implementation for a jamming attack in Section 3.2.

3.1 FOLPETTI: a Novel Attack Framework

An efficient attacks against channel hopping has two fundamental requirements: i) it should not depend on specific assumptions on the victim's hopping pattern, and ii) it needs to be continuous in time. We here explain how FOLPETTI satisfies both these requirements.

In FOLPETTI, the attacker follows the victim's channel selection using a MAB framework. Several MAB algorithms have been implemented at the defence side, where the goal is to find the optimal non-attacked communication channel [4, 22]. We adapted the framework of these solutions to implement an attack strategy. To select the optimal channel, i.e., that where the attack will have the most effect, the attacker needs to solve the Exploit-Explore dilemma. This problem can be modeled with via MAB as a Markov Decision Process (MDP). The MDP can be described via five tuples $\langle S, A, P, R, \gamma \rangle$, where:

- S is a finite set of states s;
- A is a finite set of actions a;
- $P_a(s^n, s^{n+1})$ is the probability that an action a in state s^n in time $n+1$;
- $R_a(s^n, s^{n+1})$ is the expected immediate reward received after transitioning state s^n to state s^{n+1} due to action a;
- $\gamma \in [0, 1]$ is a discount factor.

The main objective of a MDP is to find a policy π that associates an action to each state $\pi : S \rightarrow A$ to maximize the reward. Therefore, the agent tries to maximize his reward by selecting the optimal channel. At each time instant n, the attacker may stay in the previously selected state s_i, or move to another state s_j. Therefore, we define the possible actions as the set $S = s_1, s_2, ..., s_S$ of the available channels. We assume that the reward is one $R_a(s^n, s^{n+1}) = 1$ whenever the newly selected channel is used by a victim. Otherwise, the reward is zero $R_a(s^n, s^{n+1}) = 0$.

To solve this online decision problem, inspired by the work in [22], we apply the Thompson sampling algorithm as a policy. The prior distribution beta(α_j, β_j) for each access trial is a Beta distribution with parameters α_j and β_j. We denote as μ_j the event where the attack is successful ($R_a = 1$). For action s, the probability of success is given by

$$p_j(\mu_j|S_j) = \frac{\Gamma(\alpha_j + \beta_j)}{\Gamma(\alpha_j)\Gamma(\beta_j)}(\mu_j)^{\mu_j - 1}(1 - \mu_j)^{\beta_j - 1}; \tag{1}$$

where $\Gamma(\cdot)$ is the gamma function. If the state s is selected in round t and returns a reward R_a, the prior distribution for the mean reward of arm s can be updated via the Bayes rule. By utilizing the conjugacy properties, the posterior distribution for the mean reward of each arm is also a beta distribution with parameters updated based on the following rules [19]:

$$(\alpha_s, \beta_s) \leftarrow \begin{cases} (\alpha_s, \beta_s) & \text{if } a_t \neq s; \\ (\alpha_s + R_a, \beta_s + 1 - R_a) & \text{if } a_t = s. \end{cases} \tag{2}$$

3.2 FOLPETTI Combined with Jamming Attack

To validate the effectiveness of FOLPETTI, we show its application to a jamming attack. Jamming attacks have the purpose of causing a denial of service by degrading the channel's quality and preventing the exchange of packets between legitimate nodes in the network. The jammer has hence the option of voluntarily occupying the channel or causing collisions to corrupt the packet and force the node to retransmit. Furthermore, the attacker needs to act in a stealthy fashion, to avoid being detected.

In this work, we implement a constant jamming attack to continuously jam the victim's channels. Based on FOLPETTI model, the attacker follows the victim's channel selection and jams them as described in Algorithm 1. Notice that in this work we assume that the attacker can jam a single channel per time instant. We will consider an attack jamming over multiple channels in future works.

Algorithm 1 FOLPETTI Algorithm

Require: j : channel index, c : total number of channel accesses, tj : number of successful transmissions so far
$\alpha_j = \beta_j = 1$
$t_j = c = 0$
while True **do**
  for `all j` **do**
    sample $r_j \sim$ beta $(\alpha_j + t_j, \beta_j + t_j)$
  end for
  m = argmax $\{\bar{r}_j\}$;
  c ++ ; JAM();
  if channel is occupied **then** t_m+ = 1
  end if
end while

We determine the success of the jamming attack based on the Received Signal Strength Indicator (RSSI). This metric, unlike other metrics such as the Packet Error Rate (PER) or PDR, does not require the attacker to spend a lot of time in listening mode. Therefore, the attacker can remain active for the whole duration of the attack. Indeed, we have observed a drop in the RSSI when an attack takes place whether on the side of the transmitter or the attacker.

4 CHANNEL HOPPING MODEL

In this section, we focus on the victim's side. We first describe in Section 4.1 the detection method implemented by the victim to detect an attack. Then, we describe in Section 4.2 the different channel hopping strategies.

4.1 Detection Method

To estimate the impact of FOLPETTI, we use the PDR. This type of metric is often used to identify jamming attacks in the literature [29], and can be computed as

$$\text{Packet Delivery Ratio} = \frac{\sum \text{Number of PSD}}{\sum \text{Number of PT}}, \tag{3}$$

where *PSD* is the number of packets successfully received at the destination and *PT* represents the number of packets transmitted by the source. Based on this metric, we implement a statistical detection framework based on the behavior of the network without attack [20]. Indeed, on the basis of a network without attack, it is possible to calculate the average PDR and thus to define a detection threshold δ. If the PDR is smaller than δ, then the channel is under attack and the mitigation method can start. The choice of PDR as detection approach is mostly based on the consideration

that it allows to detect different types of jamming attacks without increasing the computational overhead.

4.2 Channel Hopping

Channel hopping is both an interference mitigation method and a reaction strategy against jamming attacks. It consists of in dynamically changing the communication channel to mitigate interference or counteract to jamming attacks. Assuming that the jammer does not attack on all the channels simultaneously, channel hopping is an effective method to keep the communication active. However, there are some potential issues related to its implementation. For instance, the selection of the successive channel should be realized without prior negotiation to avoid leakage of information that could be exploited by the attackers. Indeed, if the attacker eavesdrops the hopping pattern, it can continuously jam the network in all channels. In this work, we implement the following two strategies.

- **Random channel Hopping** (Tx_{Random}) : upon attack detection, the transmitter node randomly chooses a new channel from the M available ones. After the selection, it verifies the availability of the channel. If this channel is already occupied, it selects a new one. Therefore, the transmitter has a probability $M_{available}/M$ of selecting a free channel, where $M_{available}$ represents the number of channel available at time instant t.
- **Smart Channel Hopping** ($Tx_{Optimal}$): This method uses a MAB approach, and follows the model proposed by authors in [22]. This method tries to converge to the best channel available in as few steps as possible by employing a Thompson sampling formulation.

5 PERFORMANCE EVALUATION

In this section, we first describe in Section 5.1 the system we consider for implementing the attack based on the FOLPETTI strategy. Before showing in Section 5.3 the performance obtained by running the FOLPETTI attack, we describe in Section 5.2 the other channel hopping strategies employed by the attacker.

5.1 Network Model

We consider a network based on wireless nodes with limited energy, i.e. battery equipped. All devices have similar features in terms of computational capacity and memory. In particular, we consider a wireless communication network composed of three legitimate nodes connected via an access point with the IEEE 802.11 protocol and capable of transmitting on 12 different channels [11]. We assume that the attacker has the same configuration as the legitimate nodes to reduce the probability of being detected. The attacker jams the communication between the access point and node 1 as depicted in Fig. 1. We assume that legitimate nodes in the network may be able to detect an attack if their performance falls below a certain value. Therefore, a stealthy attack is possible only if the attacker is able to keep its success rate below the identifiability threshold.

To follow the effects of the jamming attacks and compute the PDR, the three considered devices communicate via the Transmission Control Protocol (TCP). Indeed, TCP provides an acknowledgement (ACK packet) for each correctly received packet. Therefore, based on these ACK packets, the transmitter can judge whether the

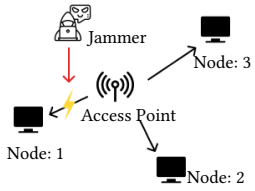


Figure 1: Network model

transmission is successful and consequently update the PDR metric. We assume that the access point constantly transmits packets every 0.1 s and begins its transmission at the start of the simulation ($t = 0$). After 10 s, the attacker starts its attack. The legitimate nodes and the attacker start their communication on the same channel. Table 1 summarizes the simulation parameters.

Parameter Name	Setting Used
Simulation Time	1790 seconds
Size of Legitimate Packet(octets)	1000
Start of jamming(seconds)	10
Start of channel hopping(seconds)	1
Energy Model	EnergyBasicModel
Distance Node - Access Point(m)	5
Threshold Detection (%)	80

Table 1: Simulation parameters

We implement the aforementioned attacker's and transmitter's channel hopping methods in the discrete event simulator NS-3 (Network Simulator-3). We modify and update the jamming module [12] with the latest version of NS-3 and integrated into it the *"ns3-gym"* module [5] allowing to implement MAB algorithms. The code is available[1].

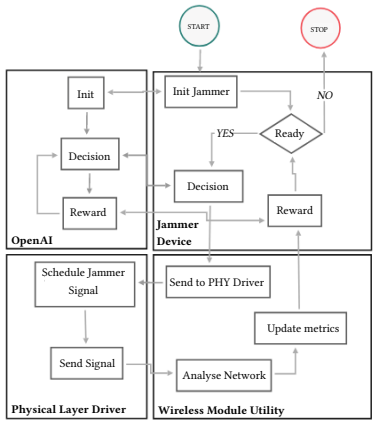


Figure 2: NS-3 Jammer work-flow

[1] If the reviewer wants to look at the code prior to publication, we can provide it through the program chairs. Due to the double-blind review process we cannot provide the link to the code at this stage.

Fig. 2 describes the work flow of the jamming attack we implemented in the NS-3 module. Four main components are required in the development of a jamming attack on NS-3. The first is the *Jammer Device* class where the jammer strategy is defined. *Wireless Module Utility* class is the second element of this model, This class provided essential functions for the jammer class such as functions to compute metrics of network performance. Moreover, this latter is also used to connect *Jammer Device* and *Physical Layer Driver* classes. Physical Layer Driver inherits from the physical class already implemented in NS-3. Thank to the latter we can define new behavior of the physical layer without modifying all the basic process of NS-3. The last part is the *OpenAi* class which implements the logic of Multi-Armed Bandit algorithm.

Based on Fig. 2, we describe the process of the FOLPETTI attack combined with the constant Jamming attack. The first step is to initialize the jammer attributes and the MAB parameters, i.e., the number of arms (in this case this corresponds to the number of channels). Once this step is done, the jammer is ready and has to make the decision on which channel to jam. To this aim, we exploit the *OpenAi* class to return the arm chosen by the algorithm. Then, the *Jammer Device* class sends all the necessary data to the *Wireless Module Utility* class that transmits this information to the *Physical Layer Driver* element. The Physical Layer Driver schedules the next Jammer Signal. Once the signal has been sent, the *Physical Layer Driver* notifies the *Wireless Module Utility* which will then perform a network analysis. Then it returns an update of the different metrics of interest for FOLPETTI, i.e., the RSSI and the PDR. These metrics are communicated to the *Jammer Device* class. Then, the *Jammer Device* class receives the RSSI value and determines the reward. This value is delivered to the *OpenAi* class to complete the procedure of the MAB algorithm.

We developed this module to be as extensible as possible. Indeed, it is possible to extend all these classes to develop new jamming strategies, metrics, and new MAB algorithms.

5.2 Comparison with Other Attack Strategies

In order to assess the validity of FOLPETTI, we compare it with four other attack strategies:

- **Random Channel Hopping** (A_{Random}): As for the mitigation method, the attacker randomly selects the channel on which to perform its attack. The channel change takes place after each transmission of the jamming signal. Hence in this case, the attacker has a probability of $1/M$ of jamming the correct channel, where M represents the total number of channels.
- **Reactive Channel Hopping** ($A_{Reactive}$): One of the most developed strategies in the literature is the jamming reactive attack, which limits the attacker's energy consumption. The attacker jams the signal only when a communication takes place. This implies that the attacker must be able to react. The reaction time must be shorter than the packet transmission time for the latter to be an effect. Moreover, when the attacker senses no communication on a certain channel, he scans all the other channels in order to know the correct transmission channel.
- **DRL based Channel Hopping**(A_{DRL}): The attacker employs the strategy developed in [31] and uses a DRL algorithm consisting of an actor and a critic to make decisions on the channel to attack. In this method, the actor observes its environment to choose the action to optimize its policy. During this time, the critic evaluates the actions performed by the actor by calculating the difference between the expected outcome and the actual one and informs it of the quality of its choice. The temporal difference value is then used to update the actor and critic model. Moreover, to optimize this model and reduce the probability of being detected, the attacker alternates between two modes: attacking phase, and listening mode. Indeed, the two agents (actor and critic) are based on neural networks that must be trained beforehand. This is why, during the training phase, the attacker is in listening mode. Once the model is trained, the attacker switches to attack mode. In this phase, the attacker scrambles the channel and can decide to switch to listening mode to re-train its neural network when performance decrease.
- **Optimal Channel Hopping** ($A_{Optimal}$): We assume the attacker is omniscient and knows the victim's channel hopping pattern. After each diffusion of the jamming signal, the attacker verifies if it has to change channel. This approach represents the optimal solution, and we considered it as baseline in our simulations.

5.3 Performance of FOLPETTI-Based Jamming

To better understand the advantage of FOLPETTI, we initially describe its behavior and compare it with that of a DRL-based attack. In this part, we assume that the victim's channel hopping strategy follows a random choice.

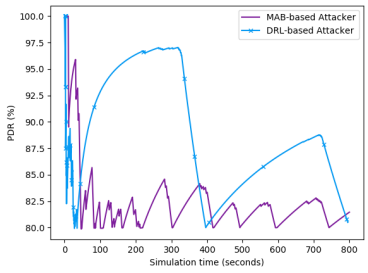
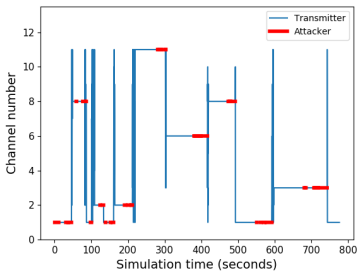
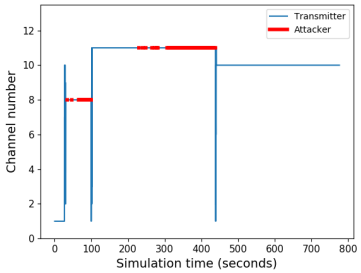


Figure 3: PDR attacker.

Fig. 3 shows the impact of FOLPETTI and DRL-based strategy on the PDR. The closer the PDR is to zero, the more effect the attack has on the network. We fix the attack detection threshold to 80% PDR. Indeed, based on [30], and the distance between the access point and the victim that we defined above, the percentage of the PDR observed in real-life in a network without attack varies between 82%- 100% in a normal behavior. Consequently, when the PDR drops at 80% a jamming attack occurs and the transmitter randomly hops into another channel. Fig. 4 shows the transition channel pattern for the transmitter and the attacker for the case of FOLPETTI attack and DRL-based attack. For reasons of readability,



(a) Channel Behavior with FOLPETTI attack



(b) Channel Behavior with DRL-based attack

Figure 4: Transition channel pattern for transmitter and attacker

we report the attacker's channel only when it is simultaneously used by the transmitter.

By combining Fig. 3 and Fig. 4a, we see that our attack tracks the channel hopping mitigation method. Indeed, at $t = 10$ the PDR drops from 100% to 89%. This is explained by the fact that the attacker and the transmitter are positioned in the same channel. Then the attacker is in the exploration step and examines the availability of the other channels, hence increasing the PDR again to 95%. At the end of this phase, the attacker decides to turn to the exploitation period and jams channel 1 causing the PDR to drop to 80%. At this point, the transmitter detects a potential attack and changes the communication medium. The victim randomly chooses a channel and verifies in a second time if this is occupied. Therefore at time $t = 49$, the transmitter switches to channel 11. However, as this has a low RSSI value, the access point will re-select a new channel (in this case, 8). Simultaneously, the attacker receives negative rewards and decides to change the jamming channel. Consequently, at $t = 54$ the attacker chooses to jam channel 8 which will again cause the PDR to drop to 80%. We observe that this behavior pattern remains until the end of the simulation. Therefore, our proposed approach allows to quickly deduce the new communication parameters used by the transmitter. Indeed, after the transmitter changes channel, the attacker can find the optimal channel so that the PDR does not exceed 84%.

On the contrary, for the DRL-based attack, we notice in Fig. 3 that at the beginning of the simulation, the PDR rapidly drops to 80%. Consequently, the legitimate nodes of the networks react and change their transmission channel from 8 to 11, as we see in Fig. 4b. However, the attacker continues to jam channel 8 for a short time until it finds that it has no effect on it. As a result, it switches to listening mode to re-train its neural network with new observations. This period appears on the figure starting from second 150 and lasts about 100 s. During this time, the attacker does not impact the network, and the values of the PDR increase. At second 300, the attacker finds its new policy and jams the channel where the transmission is taking place. Hence, the PDR values drop to the detection threshold and the transmitter will react accordingly by changing channel again. We note that this behavior is repeated until the end of the simulation.

By analyzing the behavior of the two attacks, we see that the attack based on the MAB algorithm, i.e., FOLPETTI, reacts faster to policy updates than the one based on the DRL. This is confirmed by an overall lower PDR for FOLPETTI. Indeed, this mechanism does not require any learning time, so the attacker can remain active throughout the attack, hence steadily keeping a close-to-optimal PDR.

6 ASSESSMENT OF THE FOLPETTI ATTACK IN TWO SCENARIOS

In this section, we evaluate our new strategy in two scenarios. In the first Section 6.1, we consider a victim using random channel hopping. Then, we consider a victim exploiting a smart channel-hopping strategy in Section 6.2. For these two cases, we compare the different strategies in terms of a) PDR, b) success rate of the attack, c) number of retransmissions and d) number of detections. The PDR is computed via equation (3). The success rate of the attack corresponds to the number of successfully jammed packets by the attacker over the total number of transmitted packets by the victim. The results obtained below correspond to an average over 1000 simulations.

6.1 Scenario 1: FOLPETTI Against Random Channel Hopping

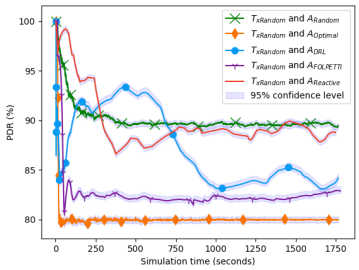


Figure 5: PDR for different strategies of attack against Random Channel Hopping Method

Fig. 5 shows the impact of the different attacks on the network considering the PDR evolution over time. As can be seen in this figure, for the FOLPETTI attack the PDR values are the closest to those of the optimal solution. Indeed, the average PDR of the optimal solution is 80% against 82% for our new approach and 85%

for the DRL-based attack. It is important to note that the channel change by the transmitter takes place when the PDR is below a threshold δ, set here at 80%. In the case of the optimal solution, the attacker is omniscient and knows in advance the future channel to jam. Therefore, once the PDR reaches the detection threshold, it will remain constant throughout the simulation. Conversely, for the attack following the strategy based on random channel hopping, the value of the PDR remains high at 90%. This can be explained by the fact that the attacker has a 1/12 probability of jamming the busy channel, therefore having limited effect on the transmission. The reactive attack has an efficiency close to the attack using a random channel strategy. Indeed, after 1 second of inactivity, the attacker re-scans all of the different channels in order to deduce the new transmission channel. Consequently, the attacker is inactive during this time and loses performance. Finally, the DRL-based approach has an average PDR value of 85% which is higher than of the FOLPETTI attack. In addition, the first learning time is considerable and the attack only manages to deduce a strategy after 600 s. In this approach, when the attacker considers itself ineffective, it has the possibility of re-training its two agents in order to re-adapt their strategies. As seen previously, during this phase, it cannot impact the network therefore having no effect on the PDR. On the other hand, the FOLPETTI attack has the ability to learn online and to remain active throughout the attack. Therefore the PDR obtained via FOLPETTI quickly decreases and closely approaches that of the optimal solution.

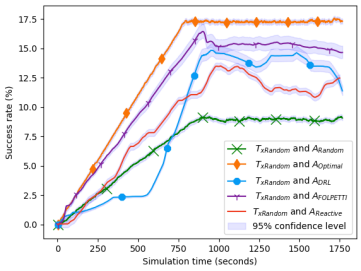


Figure 6: Success rate for different strategies of attack against Random Channel Hopping Method

The performances of the different attack strategies evaluated according to the success rate are presented in Fig. 6. When the attack is optimal at the end of the simulation, the success rate is around 17.5% against 15.5% for the FOLPETTI solution and 12.5% for the DRL-based solution. The FOLPETTI attack is more efficient than the other attack strategies and, in particular, twice as effective as that based on a random strategy. The difference between the FOLPETTI attack and the DRL-based attack can be explained by the fact that, in the first case, the attack needs fewer observations and therefore less time to converge towards the optimal solution.

By comparing the number of retransmitted packets in Table 2, we can discern that the strategy based on MAB with Thompson policy has an impact on the behavior of the network. On contrary, the random channel hopping strategy has no effect on the number of retransmissions. Indeed, the number of retransmissions is equal to

Attack	Number of Retransmissions	Number of detection
Random	74	0
Reactive	419	514
DRL-based Attack	641	1189
FOLPETTI	738	1483
Optimal	1070	1729

Table 2: Number of retransmissions and detection for different attack strategies against random channel hopping method

74 against 738 with the strategy based on the MAB and 1070 for the optimal solution. Moreover, the number of retransmissions for the DRL-based solution is 641, 13% less than for the FOLPETTI attack. The number of detections and consequently the number of channel hops at the transmitter side confirm these results. For the random solution, the PDR never drops below 80%, therefore no attack is detected and no channel hopping is performed. For FOLPETTI, the transmitter detects an attack 1483 times against 1729 times for the optimal solution. Therefore, this new type of approach produces a significant effect in terms of retransmissions and disturbance on the channel, hence decreasing the energy efficiency of the victim network. As confirmed by the results obtained previously, the attack based on DRL is less efficient than the optimal and FOLPETTI attacks and therefore leads to a lower number of detections. Indeed, this approach has a detection number of 1189, 294 detections less than with our approach.

Consequently, when the mitigation method is based on a random strategy, the proposed solution is effective and the results obtained are almost similar to those provided by an optimal solution.

6.2 Scenario 2: FOLPETTI against Smart Channel Hopping

In order to estimate the robustness of the FOLPETTI model, we evaluated it against a smart channel hopping method. We exploit the channel hopping method proposed in 4, that tries to predict the best communication channel available without interference.

The results obtained in Fig. 7 reveal different points. First, the smart method obtains better performance in terms of defense compared to a random defense strategy. In this case, the PDR is equal to 100% unlike 90% when the channel hopping strategy is based on random selection. The same behavior is also visible for the reactive attack. Indeed, the PDR remains high around 97.5%. For the two smart attackers, we notice that in the beginning of the simulation the PDR is around 94%. Therefore, this new types of attack decrease the PDR by 6% compared to a more basic method. As our attack does not need training time and converges faster towards the choice of the optimal channel, the PDR tends to drop little by little to arrive at the end of the simulation at 92.5%. Thus, we notice that our approach has an impact on the network even if the mitigation method is more efficient. This is different from the attack based on the DRL algorithm, which at convergence reaches an average PDR value rise around 94.5% due to the time needed to re-train the model.

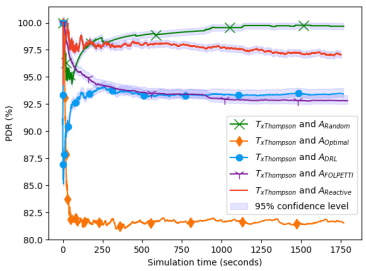


Figure 7: PDR for different strategies of attack against Smart Channel Hopping Method

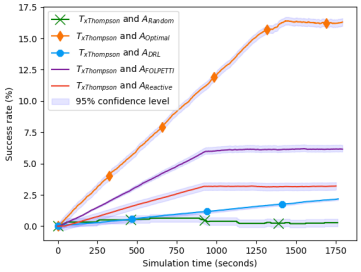


Figure 8: Success rate for different strategies of attack against Smart Channel Hopping Method

These results are also confirmed by the success rate, shown in Fig. 8. Indeed, our new attack has a success rate 7.5% higher than the one based on random channel hopping strategy, whose success rate is 0%. Moreover, if we compare the success rate between the two smart attacks, our approach has a higher success rate that converges around 7.5% after 900 s of simulation. During the same simulation time, the DRL-based attack has a success rate of 2%, i.e., 3.75 times lower than that of the FOLPETTI attack.

By looking at the number of retransmissions reported in Table 3, the attack based on MAB algorithm has an important effect on the network as it allows to generate 280 retransmission. The attack based on agent-critic method causes 212 retransmissions, i.e., 68 less than FOLPETTI attack. In addition, the legitimate nodes must change channel 513 times against 485 times when the attack is based on DRL method and 0 times with a random strategy.

Simulations demonstrated the efficiency of this new type of attack to select the optimal channel to jam even if the channel hopping method is more robust.

7 DISCUSSION

Based on the results obtained in the two scenarios considered, we observe that the proposed method has better performance than a basic strategy like a random method. In addition, FOLPETTI is also effective when the channel hopping strategy is based on a more advanced approach such as the MAB algorithm. The main advantage of the FOLPETTI based attack is that it operates without having prior knowledge of its victim. Indeed, the apprenticeship is completely online, the attacker does not alternate between long

Attack	Number of Retransmission	Number of detection
Random	38	0
Reactive	87	124
DRL-based Attack	212	485
FOLPETTI	280	513
Optimal	470	751

Table 3: Number of retransmission and detection for different attack strategies against smart Channel Hopping method

transmission and listening phases in order to discern its impact on the victim. This is possible thanks to the RSSI metric which does not require any listening time, allowing the attacker to remain active throughout its attack. Another main benefit of the Multi Armed Bandit algorithm with Thompson policy is that it requires less computing resources and can be executed in small devices as evidenced in [22]. Therefore, the attacker can use common devices whilst being mobile like a Raspberry Pi, and execute attacks in any environment.

By analyzing the number of retransmissions that FOLPETTI generates, we can deduce that this type of attack strongly impacts on the network performance. An increase in the number of retransmissions not only causes a delay in the delivery of information but also a considerable loss of energy for the transmitter. Indeed, the energy consumption E_{total} of the transmitter in 802.11 protocol can be computed with this formula:

$$E_{\text{total}} = PW_{Tx} \times T_{Tx} + PW_{Rx} \times T_{Rx} + PW_{IDLE} \times T_{IDLE} + PW_{SLEEP} \times T_{SLEEP}, \quad (4)$$

where $PW_{TX}, PW_{RX}, PW_{IDLE}$ and PW_{SLEEP} represent the power consumption of the transmitter node when it is in the transmitting, listening, idle, and sleep states respectively. T_{TX}, T_{RX}, T_{IDLE} and T_{SLEEP} correspond to the total time spent by the transmitter in each respective state. In addition, the total amount of energy for one retransmission E_{Retx} can be described as:

$$E_{Retx} = R(T_{Tx} \times PW_{Tx}) + R(T_{Rx} \times PW_{Rx}), \quad (5)$$

where R corresponds to the total number of packets transmitted during one retransmission such as data or acknowledgements packets. Diverse measures of power consumption for several type of network interface controller are provided in [2]. The authors show that transmitting and listening are the two most energy demanding modes. Combining this consideration with equations (4) and (5), we notice that FOLPETTI drastically reduces the lifetime of a device. Indeed, if the attacker is equipped with an Alfa AWUS036h network interface controller compatible with a Raspberry Pi, the energy consumption for each step can be determined as shown in table 4. These information are provided by [5].

Based on the previous simulations and by considering the values given in Table 5 corresponding to the transmission T_{Tx} time and reception time T_{Rx} of the various packets involved for one retransmission, we can compute the additional network energy consumption under a jamming attack.

Combining formula (5) and the information from Table 4, we deduce that the energy consumption for one retransmission E_{Retx} is 0.0050431 J. For the scenario 1, the number of retransmission

Operating Mode	Sleep	Idle	Tx	Rx
P(W)	0.001	0.30	0,67	0,34

Table 4: Power Consumption for 2.4 GHz Operation

Type of packet	Transmission time (s)	Reception time (s)
Data	0,00397	0,00695
Ack	0,00002	0,00003

Table 5: Average time spent by each state during a retransmission according to the type of packet

is 738 when the strategy attack is based on Multi Armed Bandit. Therefore the additional energy expended by the transmitter is 3.72 J for an attack of 30 minutes. For the same situation, the DRL-based attack generates an increase of energy consumption of 3.23 J, which is 0.49 J less than the FOLPETTI attack. With the random jamming approach, the supplementary energy consumed by the access point is 0.373 J, i.e 10 times less than for FOLPETTI method.

The same results are also observed for the second scenario. Indeed, for 30 minutes of simulations, the number of retransmissions is 280 when the jammer is based on the new approach and 38 when the attacker applies a random logic. Consequently, the emitter consumes 1.41 J with the FOLPETTI model against 0.19 J with a basic method. Our new approach has a greater effect on the victim's energy consumption than other attack strategies. This is confirmed comparing our attack to the DRL-based. Indeed, the DRL-based method causes an energy expenditure for the victim of 1.06 J, i.e. 0.35 J less than the FOLPETTI attack.

The main advantage of FOLPETTI attack is that it does not require training. Indeed, the update of its policy is done in real time. Unlike other works, we do not rely on ACK packets to define the success of our attack. Consequently, the attacker must not wait to receive this packet to recover a reward. The update of the policy is done in a smaller time and the attacker will converge faster to the optimal solution. As a result, FOLPETTI attack is more effective as we proved in the previous section.

To finish, our new type of attack can be considered a framework to perform several known attacks. Indeed, being able to predict the communication channel can be useful in replay attack or flooding attack in order to limit their probability of detection. Moreover, more and more attacks are based on machine learning algorithms and consequently data sets in order to deduce information [1, 16]. In particular, they require passive listening to the network in order to collect the maximum of data. This step is crucial because the effectiveness of the machine learning algorithm relies on the data set. In this case, FOLPETTI makes it possible to follow the channel hop and thus not to lose any information necessary for the creation of the data set. In addition, the mitigation method by channel hopping is not only present in Wi-Fi protocol. Indeed, this strategy is also employed by other communication protocols. Thereby, this new framework does not depend on the Wi-Fi protocol and can be effective in other use cases.

8 POSSIBLE COUNTERMEASURES

We have just shown that the different strategies employed in the channel hopping literature lose effectiveness when the attacker operates with a more elaborate algorithm. Indeed, even if the FOLPETTI attack has a smaller effect when a smart channel hopping is used, this new type of attack leads to a decrease in network performance. Several works aimed at creating more robust MAB algorithms exist in the literature and could be applied in MAB channel hopping strategies [24]. Moreover, it is also possible to evade the FOLPETTI attack by directly attacking the MAB model of the attacker. Indeed, this type of algorithm can be subjected to adversarial attacks, and two strategies could be designed to disturb the FOLPETTI model.

- **Poisoning attacks:** One of the countermeasures can be the poisoning attack [14]. The purpose of the transmitter is to deceive the attacker about its choice. The victim influences the attacker to jam an unused channel, having hence no effect. To this aim, legitimates nodes must falsify certain rewards in order to control the attacker's decision. In this situation, nodes can possibly voluntary transmit in an jammed channel to influence the number of rewards and therefore the attacker's policy.
- **Delayed feedback attack:** Another possible countermeasure could be to implement delayed feedback in order to deceive the attacker's choice. The goal of the victim is to delay the transmission of feedback information that serves as a reward to the attacker. In this case, the complexity of the MAB algorithm for the attacker will increase significantly as demonstrated in [6]. As a result, the attacker will need a significant time to converge to the optimal solution.

One of the main difficulties of these solutions is that the transmitter must completely know the parameters of the MAB algorithm employed by the attacker, such as the type of the rewards or the policy. In addition, the legitimates nodes of the network must be able to implement these methods without impacting the network performance. Finally, an important point to take into account when designing these countermeasures in the IoT context is the power consumption they require. Consequently, designing solutions against FOLPETTI attacks requiring few resources, knowledge, and not impacting the network remains an open challenge.

9 CONCLUSION AND FUTURE WORK

In this article, we presented FOLPETTI, a novel smart attack that operates in an unknown network in the absence of a-priori information about the network itself. FOLPETTI has the ability to understand and predict the behaviour of victims and more particularly their channel hopping strategy. We evaluated FOLPETTI against two defence strategies, one more classic based on random channel hopping and one more advanced based upon MAB algorithm. In the first situation, FOLPETTI approaches an optimal attack solution in terms of PDR, success rate, and number of retransmissions. In the second case, our new attack is still able to enhance the success rate to 5% against 2.5% for the other smart attack. The simulations results demonstrate that the proposed strategy can improve the overall jamming performance. Moreover, our approach

can be considered a framework to be used for any multi-channel communication protocol. Although we used FOLPETTI to perform a jamming attack, it can also be used for other attacks that may benefit from channel following strategies such as replay attack. Our future works will evaluate this new solution in a testbed composed of several nodes and jammers in order to evaluate its performance in real life situation. Moreover, since the final goal of creating an attack is to improve the security system, we will add countermeasures in the smart channel hopping method against a FOLPETTI attack. This study will focus on the trade-off between the security, efficiency, and the network performance.

ACKNOWLEDGMENTS

This work was partially supported by a grant from CPER DATA and by the General Armament Direction, France and the Defense Innovation Agency, France

REFERENCES

[1] Abbas Acar, Hossein Fereidooni, Tigist Abera, Amit Kumar Sikder, Markus Miettinen, Hidayet Aksu, Mauro Conti, Ahmad-Reza Sadeghi, and Selcuk Uluagac. 2020. Peek-a-boo. *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks* (Jul 2020). https://doi.org/10.1145/3395351.3399421

[2] Salvatore Chiaravalloti, Filip Idzikowski, and Lukasz Budzisz. 2011. Power consumption of WLAN network elements. *TKN Technical Reports Series* (08 2011). https://doi.org/10.13140/2.1.4424.8005

[3] Atheros Communications. [n.d.]. *Single-Chip 2x2 MIMO MAC/BB/Radio with PCI Express Interface for 802.11n 2.4 and 5 GHz WLANs.* https://datasheetspdf.com/datasheet/AR9280.html

[4] Hiba Dakdouk, Erika Tarazona, Reda Alami, Raphaël Féraud, Georgios Z Papadopoulos, and Patrick Maillé. 2018. Reinforcement learning techniques for optimized channel hopping in IEEE 802.15. 4-TSCH networks. In *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. 99–107.

[5] Piotr Gawlowicz and Anatolij Zubow. 2019. ns-3 meets OpenAI Gym: The Playground for Machine Learning in Networking Research. 113–120. https://doi.org/10.1145/3345768.3355908

[6] Aditya Grover, Todor Markov, Peter Attia, Norman Jin, Nicolas Perkins, Bryan Cheong, Michael Chen, Zi Yang, Stephen Harris, William Chueh, and Stefano Ermon. 2018. Best arm identification in multi-armed bandits with delayed feedback. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 84)*, Amos Storkey and Fernando Perez-Cruz (Eds.). PMLR, 833–842.

[7] Kanika Grover, Alvin Lim, and Qing Yang. 2014. Jamming and anti–jamming techniques in wireless networks: a survey. *International Journal of Ad Hoc and Ubiquitous Computing* 17, 4 (2014), 197–215.

[8] Fengxian Guo, F Richard Yu, Heli Zhang, Xi Li, Hong Ji, and Victor CM Leung. 2021. Enabling massive IoT toward 6G: A comprehensive survey. *IEEE Internet of Things Journal* (2021).

[9] Guoan Han, Liang Xiao, and H Vincent Poor. 2017. Two-dimensional anti-jamming communication based on deep reinforcement learning. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2087–2091.

[10] Wan Haslina Hassan et al. 2019. Current research on Internet of Things (IoT) security: A survey. *Computer networks* 148 (2019), 283–294.

[11] S. Kawade, T. G. Hodgkinson, and V. Abhayawardhana. 2007. Interference Analysis of 802.11b and 802.11g Wireless Systems. In *2007 IEEE 66th Vehicular Technology Conference*. 787–791. https://doi.org/10.1109/VETECF.2007.174

[12] Network Security Lab. [n.d.]. *Wireless jamming model.* https://www.nsnam.org/wiki/Wireless_jamming_model

[13] Eun-Kyu Lee, Soon Y Oh, and Mario Gerla. 2010. Randomized channel hopping scheme for anti-jamming communication. In *2010 IFIP Wireless Days*. IEEE, 1–5.

[14] Fang Liu and Ness Shroff. 2019. Data Poisoning Attacks on Stochastic Bandits. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). 4042–4050.

[15] Xin Liu, Yuhua Xu, Luliang Jia, Qihui Wu, and Alagan Anpalagan. 2018. Anti-jamming communications using spectrum waterfall: A deep reinforcement learning approach. *IEEE Communications Letters* 22, 5 (2018), 998–1001.

[16] N. Msadek, R. Soua, and T. Engel. 2019. IoT Device Fingerprinting: Machine Learning based Encrypted Traffic Analysis. In *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. 1–8.

[17] Nima Namvar, Walid Saad, Niloofar Bahadori, and Brian Kelley. 2016. Jamming in the internet of things: A game-theoretic perspective. In *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1–6.

[18] Hossein Noori and Saeed Sadeghi Vilni. 2020. Jamming and anti-jamming in interference channels: a stochastic game approach. *IET Communications* 14, 4 (2020), 682–692.

[19] Daniel Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. 2017. A tutorial on thompson sampling. *arXiv preprint arXiv:1707.02038* (2017).

[20] Michael Spuhler, Domenico Giustiniano, Vincent Lenders, Matthias Wilhelm, and Jens B. Schmitt. 2014. Detection of Reactive Jamming in DSSS-based Wireless Communications. *IEEE Transactions on Wireless Communications* 13, 3 (2014), 1593–1603. https://doi.org/10.1109/TWC.2013.013014.131037

[21] Xiao Tang, Pinyi Ren, and Zhu Han. 2018. Jamming mitigation via hierarchical security game for IoT communications. *IEEE Access* 6 (2018), 5766–5779.

[22] Viktor Toldov, Laurent Clavier, Valeria Loscrí, and Nathalie Mitton. 2016. A Thompson Sampling approach to channel exploration-exploitation problem in multihop cognitive radio networks. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 1–6.

[23] Bikalpa Upadhyaya, Sumei Sun, and Biplab Sikdar. 2019. Machine learning-based jamming detection in wireless iot networks. In *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*. IEEE, 1–5.

[24] Daniel Vial, Sanjay Shakkottai, and R. Srikant. 2021. Robust Multi-Agent Multi-Armed Bandits. arXiv:2007.03812 [cs.LG]

[25] Feng Wang, M Cenk Gursoy, and Senem Velipasalar. 2021. Adversarial Reinforcement Learning in Dynamic Channel Access and Power Control. In *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 1–6.

[26] Ximing Wang, Yuhua Xu, Jin Chen, Chunguo Li, Xin Liu, Dianxiong Liu, and Yifan Xu. 2020. Mean field reinforcement learning based anti-jamming communications for ultra-dense internet of things in 6G. In *2020 International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 195–200.

[27] Liang Xiao, Xiaoyue Wan, Wei Su, Yuliang Tang, et al. 2018. Anti-jamming underwater transmission with mobility and learning. *IEEE Communications Letters* 22, 3 (2018), 542–545.

[28] Jianliang Xu, Huaxun Lou, Weifeng Zhang, and Gaoli Sang. 2020. An intelligent anti-jamming scheme for cognitive radio based on deep reinforcement learning. *IEEE Access* 8 (2020), 202563–202572.

[29] Wenyuan Xu, Wade Trappe, Yanyong Zhang, and Timothy Wood. 2005. The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks. In *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. Association for Computing Machinery, 46–57. https://doi.org/10.1145/1062689.1062697

[30] Wenyuan Xu, W. Trappe, Yanyong Zhang, and Timothy Wood. 2005. The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks. *Proceedings of the International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*. https://doi.org/10.1145/1062689.1062697

[31] Chen Zhong, Feng Wang, M Cenk Gursoy, and Senem Velipasalar. 2020. Adversarial jamming attacks on deep reinforcement learning based dynamic multichannel access. In *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 1–6.

[32] Quan Zhou, Yonggui Li, and Yingtao Niu. 2021. Intelligent Anti-Jamming Communication for Wireless Sensor Networks: A Multi-Agent Reinforcement Learning Approach. *IEEE Open Journal of the Communications Society* 2 (2021), 775–784.