Capodoglio: Tackling Multi-Armed Bandit Jamming Attacks

Shuo Wang, Alessandro Brighente, Valeria Loscri, Junqing Zhang, Mauro Conti

Capodoglio: Tackling Multi-Armed Bandit Jamming Attacks

Shuo Wang*, Alessandro Brighente*, Valeria Loscri†, Junqing Zhang‡, Mauro Conti*§
* University of Padua, Department of Mathematics, Padua, Italy
† FUN, Inria, Lille, France
‡ Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool, United Kingdom
§ Örebro University, Sweden

shuo.wang@studenti.unipd.it, {alessandro.brighente, mauro.conti}@unipd.it,
valeria.loscri@inria.fr, junqing.zhang@liverpool.ac.uk

***Abstract*—Jamming attacks present a significant security threat to wireless networks by exploiting the open wireless medium, causing not only a denial-of-service, but also overloading the network and interfering with signals. The jamming attack can enable more sophisticated disruptions, such as protocol-aware and learning-based jamming, where adversaries transmit selectively upon detecting legitimate network activity, and this selective transmission conserves attacker resources and complicates detection. Channel hopping is widely adopted as a mitigation strategy, allowing networks to move away from interfered channels dynamically. However, recent studies have demonstrated that intelligent jammers, such as the Multi-Armed Bandit (MAB)-based attack, can effectively learn and predict channel hopping patterns, thereby continuously jamming communications and severely degrading network performance. Designing robust countermeasures against such intelligent jamming remains an open and critical challenge. In this paper, we analyze a kind of MAB-based attack methodology and propose two refined variants employing online and offline paradigms. To counteract these advanced threats, we introduce three defense strategies: (i) *speeding up*, which modifies the frequency of channel-hopping decisions to avoid the jammer's attack, (ii) deploying a *helper node*, diversify the communication patterns to affect the attacker's learning process, and (iii) employing a mirror Multi-Armed Bandit (*mirror MAB*) approach to predict and bypass channels of highest jamming probability. Comprehensive evaluations demonstrate the effectiveness of our proposed defenses, significantly mitigating MAB-based attacks. Specifically, our strategies achieve a PDR exceeding 90% under persistent attack conditions, while effectively preserving robust and dynamic channel hopping behaviors.**

***Index Terms*—Wireless Security, Jamming, Channel Hopping, Multi-Armed Bandit.**

I. Introduction

Guaranteeing security in the wireless communication, such as WiFi protocols based on the IEEE 802.11 family of standards, is challenging due to the vast number of interconnected devices and their inherently distributed nature [1], [2]. Typically, wireless networks are widely used in information acquisition, command transmission, drone patrol, construction site communication, and emergency communication. The wireless network's infrastructure relies on wireless connections, whose openness inherently exposes communication to security and privacy threats [3], [4]. Among these classical threats, *jamming* is particularly harmful, leading to denial-of-service, overloading the network. Unlike higher-layer threats, jamming operates at the physical layer, which will disrupt the communications directly. A promising countermeasure against jamming is *channel hopping*, which dynamically switches communication channels to mitigate interference [5], [6].

Various defensive strategies have been explored, including reputation-based threat detection and isolation mechanisms [7], decentralized Multi-Armed Bandit (MAB) frameworks [8], and coded rendezvous schemes such as SPADE [9]. Nevertheless, existing solutions continue to face limitations, such as reactive-only responses, added communication overhead, or incomplete protection against smart attackers [10], [11]. In [12], [13], the algorithm employs Q-learning to finalize the selection of the communication channel. These strategies vary from basic sequential or random approaches to sophisticated game theory-based and machine learning-driven techniques [14]–[18]. Recently, Reinforcement Learning (RL)-based methods have attracted significant attention due to their adaptability. Specifically, MAB algorithms have demonstrated substantial efficacy in dynamically managing noisy wireless environments [19], [20]. For instance, the authors in [17], [18] utilized quantized Signal to Interference plus Noise Ratio (SINR) values as input states for RL frameworks. More adaptive approaches, such as those proposed by Liu *et al.* [21], consider spectrum waterfalls, thus removing the dependency on prior knowledge of the jamming pattern. Furthermore, multi-agent RL-based strategies

have been proposed to address both jamming and mutual interference [22], [23]. Nonetheless, most of these methods overlook the power limitations inherent to wireless communication networks' infrastructure, with only a few studies explicitly addressing this critical issue [24]. Advancements in inexpensive and accessible hardware have enabled attackers to also adopt increasingly intelligent and adaptive strategies. Recent studies [10], [25] demonstrated that an advanced MAB-based jammer, namely FOLPETTI, can efficiently disrupt communications despite sophisticated defensive channel hopping techniques (random or deep RL-based).

In this paper, we analyze and develop the MAB-based attacks, and meanwhile address the limitations by explicitly focusing on defense strategies against the MAB-based jamming. We begin by describing the underlying MAB approach and differentiating two MAB-based attack implementations (i.e., online and offline attacks), clarifying the design for each attack. Subsequently, we propose three novel defense strategies: *speeding up*, *helper node*, and *mirror MAB*. Each strategy is detailed in terms of operational principles, highlighting its respective advantages and drawbacks. Our simulations demonstrate that these proposed countermeasures significantly improve network resilience, boosting the Packet Delivery Ratio (PDR) even under sophisticated MAB-based attacks. The main contributions of this paper are summarized as follows:

- We extend the analysis of the MAB attacker model, develop and evaluate the online and offline attack strategies against basic channel hopping methods, to provide the baseline for different countermeasures.
- We propose and thoroughly analyze three novel defense strategies (*speeding up*, *helper node*, and *mirror MAB*), discussing their operational characteristics and effects for the MAB-based attacks.
- We evaluate the proposed strategies by the metrics of packet delivery, and test various settings or parameters of each strategy, demonstrating their effectiveness against the MAB-based jamming algorithm we used in the paper.

The project focuses on exploring mitigations for MAB-based attacks. Capodoglio, aka the "sperm whale", represents a mammal that is deep-diving (simulating the exploration of jammers) and highly adaptable (using a variety of strategies for defense). This metaphor highlights the framework's ability to operate at varying levels of complexity while maintaining a single, coherent, and effective outcome.

The rest of the paper is structured as follows. Section II introduces the system overview used for our work, while Section III provides an overview of the MAB framework as a core methodology of attacks or defense strategies. Section IV discusses the online and offline MAB-based attacks. Next, Section V elaborates on our proposed defense mechanisms. Section VI presents the simulation setup and results, validating the effectiveness of the proposed solutions. Finally, Section VII concludes the paper and outlines potential directions for future research.

II. System Overview

In this section, we describe the system model considered in this paper, including the network model, threat model, and defense strategy.

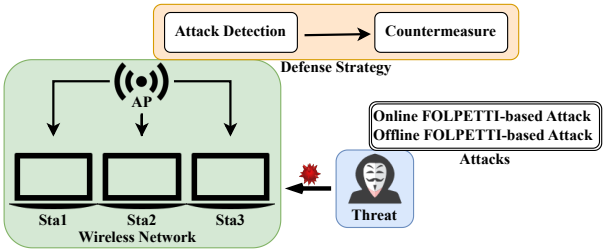


Fig. 1. System Model

A. Network Model

In this paper, the network model is a wireless network system, which includes an Access Point (AP), and three stations, as shown in Fig. 1. Specifically, the AP in our network model plays the critical role in running the defense strategy and reacting to the environment state changes. The wireless network system is used to test the performance of our proposed defense strategies.

B. Threat Model

As shown in Fig. 1, the threat model in our paper is the jammer, which aims to attack the target frequency channel under the guidance of the MAB-based attacks. In the field of wireless security, jammers typically emit continuous noise, mitigating the quality of communication at the expense of energy efficiency. Specifically, the jammer in our paper is a kind of intelligent jammer, which employs machine-learning techniques to infer channel hopping patterns and target them adaptively. Also shown in Fig. 1, the jammer runs online or offline MAB-based attacks to track the channel switching of the target wireless network and keep the jamming persistent and successful. The MAB-based attacks will be described in Section IV.

C. Defense Strategy

As shown in Fig. 1, the wireless network, specifically the AP, runs the defense strategies for avoiding compromise from attackers. In Section V, we proposed

three countermeasures and tested them in Section VI. Despite the countermeasures having different specific implementations, they are still based on the following two essential steps:

1) Attack Detection: In the wireless network, the AP runs a detection method to be aware of the outside attacks. To develop the detection method, we use the PDR as the performance metric for the detection method. This type of metric is often used to identify jamming attacks [26], [27] and can be computed as

$$\text{PDR} = \frac{\sum \text{Number of PSD}}{\sum \text{Number of PT}}, \tag{1}$$

where PSD is the number of packets successfully received at the destination, whereas PT represents the overall number of packets transmitted by the source.

We implement an effective detector based on the network status and PDR metric, which runs in the AP as a fundamental component. In particular, it computes the average PDR in the latest rounds and thus defines a detection threshold δ. If the AP detects PDR below δ for consecutive intervals, then it raises an alarm that the current channel is under attack. The choice of PDR as a detection approach is mainly based on the consideration that it allows the detection of jamming attacks without increasing the computational overhead.

2) Countermeasure: Once an attack is detected, the AP activates one of the following defense strategies:

- **Speeding Up** uses the random channel hopping algorithm, with an increasing frequency of channel hopping, to make the RL-based attacker hard to follow the network current channel.
- **Helper Node** introduces a third-party device to mislead the jammer into losing the channel change tracking. The random channel hopping algorithm runs for channel switching.
- **Mirror MAB** runs in the wireless network and predicts the vulnerable channels, updating the parameters alongside the system's running. The AP decides the next channel from the available channels, except for channels that are probably being attacked.

These defense strategies, i.e., speeding up, helper node, and mirror MAB, will be described in Section V.

III. Multi-armed Bandit Framework

In this study, we could formulate the framework for both the jammer and the legitimate AP as an MAB algorithm and develop it based there own roles. Exploiting the adaptive nature of MAB algorithms, each agent continuously updates its channel selection, thereby improving network robustness under adversarial interference. Adaptive to our project, the algorithm can be described with a tuple $\langle S, A, R, P \rangle$, where:

- S is a finite set of states s;
- A is a finite set of actions a;
- $R(a_t, s_t)$ is the expected received under state s_t and action a_t at round t, and r_t is immediate reward at round t;
- $P(a_t, s_t)$ is the probability that an action a in state s_t and action a_t at round t;

The MAB is a classical RL algorithm, and it provides a framework for decision-making under uncertainty, where an agent selects arms (i.e., actions) from a set of available options. The goal of the MAB algorithm is to maximize cumulative rewards R; the higher reward observed at each round, the optimizer is selected by the MAB, which is equivalent to minimizing cumulative regret. The biggest difference between MAB problems and RL is that in MAB problems, the reward is only related to the action and has no relationship with other observations or states. In other words, in MAB, the reward r only depends on the choice of actions at round t.

In this paper, the MAB-based attacks or mirror MAB can be modeled via the MAB algorithm. The main objective of the algorithm is to identify a policy π associating an action with each state tuple (i.e., $\pi \colon S \to A$). We use the Thompson sampling algorithm as our policy to model the problems, inspired by [19]. Thompson sampling is one of the most effective algorithms for solving contextual MAB problems. In Thompson sampling, there are K arms (i.e., actions) and k indicates the arm index, the $\text{Beta}(\alpha, \beta)$ distribution for each action corresponds to one arm with parameters α_k and β_k [28], [29]. We denote by μ_k the event of success $(r_t(a_t) = 1)$ when selecting action a at round t. For each arm k, the probability of success is given by:

$$p(\mu_k) = \frac{\Gamma(\alpha_k + \beta_k)}{\Gamma(\alpha_k)\,\Gamma(\beta_k)} \left(\mu_k\right)^{\alpha_k - 1} \left(1 - \mu_k\right)^{\beta_k - 1}, \tag{2}$$

where $\Gamma(\cdot)$ is the gamma function [30]. Take these priors to be $beta$ distributed with parameters $\alpha \in (\alpha_1, ..., \alpha_K)$ and $\beta \in (\beta_1, ..., \beta_K)$, where K is the available actions number in the model. If the MAB algorithm applies arm m at round t and returns a reward r_t, the $Beta$ parameters for that channel are updated by Bayes' rule, taking advantage of the conjugacy properties:

$$(\alpha_k, \beta_k) \leftarrow \begin{cases} (\alpha_k, \beta_k) & \text{if } m \neq k, \\ (\alpha_k + r_t,\ \beta_k + 1 - r_t) & \text{if } m = k, \end{cases} \tag{3}$$

where $k \in K$ is the arm indicator in the paper. Hence, the MAB algorithm with Thompson sampling forms the core methodology in our project. A beta distribution with parameters (α_k, β_k) has mean $\alpha_k(\alpha_k + \beta_k)$, and the distribution becomes more concentrated as $\alpha_k + \beta_k$ grows.

The MAB with Thompson sampling flowchart is shown in Fig. 2. The $Beta$ distribution ensures that arms with higher observed success rates gradually receive larger α_k relative to β_k, increasing their probability of being selected, while still allowing occasional exploration of less-sampled arms. By updating the $Beta$ parameters in proportion to observed successes and failures, Thompson sampling naturally skews future samples toward arms with higher empirical performance, while preserving randomness for continued exploration.

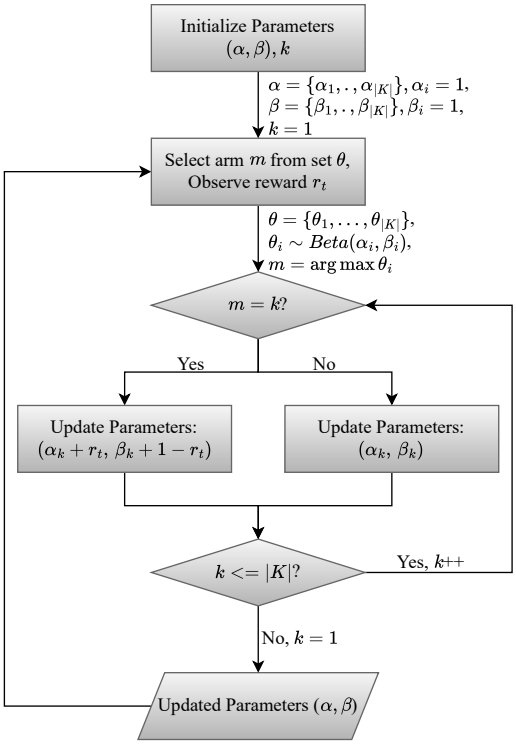


Fig. 2. MAB with Thompson Sampling Framework

IV. The MAB-based Attacks

In this section, we provide a detailed description of the proposed attacks. E. Bout *et al.* [10] have validated the effectiveness of MAB, showing its application to a jamming attack. Typically, an efficient attack against channel hopping has two fundamental requirements: (i) it should not depend on specific assumptions about the victim's hopping pattern, and (ii) it needs to be continuous in time.

In this paper, we developed two MAB-based attacks, i.e., online and offline, based on different considerations [31]. The main difference between the two approaches is the frequency of channel updates and the reward obtained in each round. Based on the approach proposed in [10], we implemented the online MAB-based attack and modified the original model to develop the offline MAB-based attacks. The following are the explanations of these attacks.

- **Online Attack:** In [10], the authors proposed the original MAB model. We develop the online MAB-based attack on the original model, where the channel selection and rewards computation are performed on a per-round basis. This means that, for each round, if the attack selects the arm a_t and successful attacks on arm k (i.e., $a_t = k$), then $\alpha_k = \alpha_k + 1$; if not, $\beta_k = \beta_k + 1$.
- **Offline Attack:** An alternative approach is offline learning, i.e., we assume that rewards and channel updates are computed on batch rounds. Specifically, let us assume that the offline MAB-based attack updates its choice and parameter every N rounds. Then, the parameters are updated based on the number of successes and failures of the attack. In particular, assume the attacker pulls arm a_t for the current round t. By denoting as r_{succ} the number of successes for arm $a_t = k$ and as r_{fail} the number of failures for arm $a_t \neq k$ at a batch round N. Then, at the end of the batch, $\alpha_k = \alpha_k + r_{succ}$, and $\beta_k = \beta_k + r_{fail}$.

V. Defense Strategy Against MAB

In this section, we describe the proposed approaches to defend against MAB-based attacks. We propose a *speeding up* strategy in Section V-A, a *helper node* strategy in Section V-B, and a *mirror MAB* strategy in Section V-C.

A. Speeding Up

In wireless networks, the radio channel is ordinarily held for long intervals, such as minutes to hours, thereby the attacker has enough time to analyze the channel status. Thus, in real-world wireless networks, the channel hops only when external conditions or policy apply. MAB-based attacks rely on accurate feedback to update the $Beta$ parameters that govern their arm-selection rule. The speeding up countermeasure neutralizes this condition by controlling the legitimate transmitter (AP and stations) to hop more aggressively, thereby invalidating the jammer's accumulated knowledge.

The defense strategy, i.e., speeding up, is that once the victim detects the attack, it increases the rate at which it performs channel hopping. In particular, the selection of a new channel is not subject to a significant decrease in the PDR, but rather to a given timing. With the speeding up strategy, channel changes occur periodically: after every N transmitted rounds, the transmitter switches to the next new channel according to a specific policy. Consequently, the jammer observes

a non-stationary target and must continuously re-learn the reward landscape, markedly reducing the dwell time it can spend on any single channel.

B. Helper Node

Because MAB-based attacks require a certain number of steps to update their strategy and attack the targets, the victim can utilize this time, sacrificing part of its resources to defend against the MAB-based attack. To this aim, the victim can deploy an additional (third-party) node that runs a fixed hopping scheme based on the various scenarios.

As shown in Fig. 3, let us consider the round t_1 in which the victim detects the attack on the channel with index ch_i. The helper node will turn on the third-party device for transmitting noisy data on channel ch_i, while the legitimate station transmits on another channel ch_j, i.e., $ch_j \in M \setminus ch_i$, where M is the available channels list. The role of the helper node is thus to poison the decision-making capabilities of MAB: by creating dummy regular traffic on channels not used by the legitimate node, the helper node forces MAB to attack useless (i.e., information-less) channels. This strategy requires coordination between the helper node and the legitimate transmitting device to avoid interference.

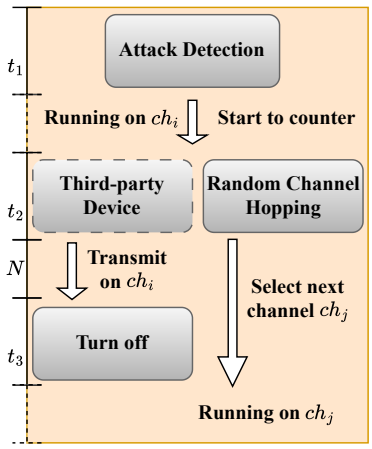


Fig. 3. Flowchart for Helper Node

C. Mirror MAB

In this section, we introduce the mirror MAB strategy (i.e., an MAB algorithm running in the device AP) to counteract the MAB-based attacks. Our objective is to safeguard WiFi communications by leveraging the MAB algorithms. When the network is active, the AP executes the mirror MAB algorithm, continuously updating Thompson sampling parameters based on observable environment states and returned rewards.

The mirror MAB algorithm runs a *mock* of the jammer's behavior in parallel. More precisely, it internally simulates the attacker's jamming patterns to anticipate which channels the attacker will attack. By modeling the jammer's state transitions, the mirror MAB maintains a set of parameters for each channel. These parameters are periodically updated using observed states, effectively learning how the jammer operates. The detailed steps are given below.

1) **Initialize Parameters:** At the beginning of the system, the mirror MAB initializes the relative parameters, e.g., $Beta(\alpha, \beta)$.
2) **Observe Reward:** Compute the reward r after a batch round.
3) **Parameters Update:** After observing the reward, the mirror MAB adjusts its internal parameters (i.e., α, β).
4) **Vulnerable Channels & Available Channels:** Using the updated parameters, the mirror MAB predicts a vulnerable channels list $B = \{ch_{b_1}, ..., ch_{|B|}\}$ most likely under attack. Meanwhile, the mirror MAB returns an available channels list $M = \{ch_{m_1}, ..., ch_{|M|}\}$ from the wireless network.
5) **Channel Switching:** Selecting the next channel ch, where we use the random policy to decide the new channel from the channels list $M \setminus B$.

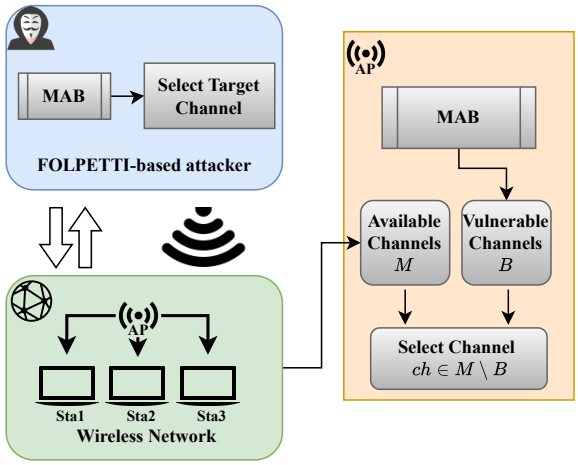


Fig. 4. Countermeasure: Mirror MAB

As shown in Fig. 4, the rightmost part stands for the mirror MAB method, where mirror MAB runs in the AP device. The mirror MAB is a trusted approach in the network, and it could collect the rewards and update the parameters according to the network status, and predict the vulnerable channels for the network.

VI. Performance Evaluation

We describe in Section VI-A the system we consider for implementing the evaluation settings. In Section VI-B, we describe the different MAB strategies

employed by the attacker, then show the performance by running the attack method in Section VI-C.

A. Evaluation Settings

We consider a wireless network composed of wireless devices, specifically, the wireless network includes an AP, three legitimate stations connected via an AP, and capable of transmitting on 12 different channels. We assume that the attacker aims to jam the communication between the AP and the legitimate nodes in the network.

To follow the effects of the jamming attacks and compute the PDR, we assume that the AP constantly transmits 12 packets every round and begins its transmission at the start of the simulation ($t = 0$). After 10 seconds, the attacker starts its attack. The legitimate stations and the attacker start their communication on the same channel. Table I summarizes the simulation parameters.

TABLE I
SIMULATION PARAMETERS

Parameter	Setting
Simulation Time (seconds)	1800
Start of jamming (seconds)	10
Start of channel hopping (seconds)	1
Threshold Detection (%)	80

We implemented this module using a custom-built network simulator, considering only the three kinds of fundamental nodes, i.e., the legitimate stations, the AP, and the attacker (jammer). For every setting, we run the simulation one hundred times to collect the results and calculate the average performance.

B. Attacks Performance

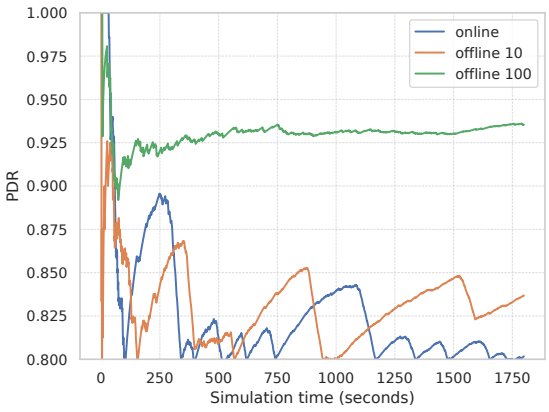
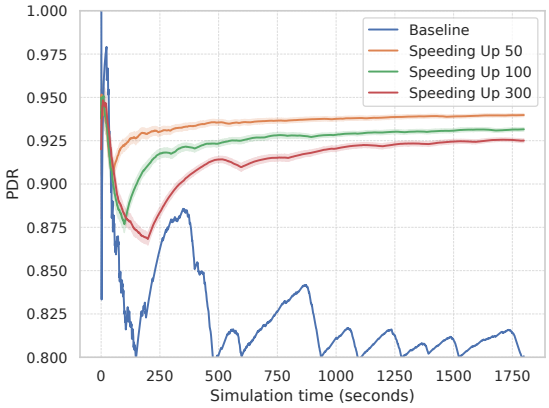


Fig. 5. PDR for different attack strategies

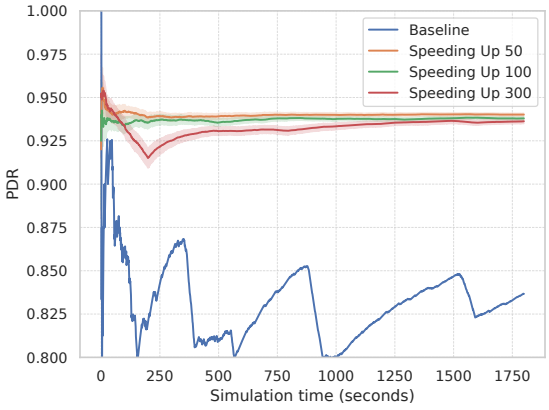
We develop and evaluate the two implementations of the MAB-based attacks, i.e., the online and offline attacks. The attack detection threshold to 80% PDR, i.e., when the victim detects a PDR below 80% it switches to the following channel. The closer the PDR is to the detection threshold, the more the attack affects the network.

In Fig. 5, the impact of online and offline MAB-based attacks on the PDR without defense strategies is shown. The results indicate that the online attacker consistently reduces the network's PDR more significantly than the offline attackers, where $10, 100$ stand for the parameters update delay. Under the same evaluation settings, the online attacker successfully jams the victim's channel approximately 39.7% of the time, compared to only 20.6% for the offline attacker. In particular, we use the online setting and offline 10 setting as the corresponding baselines for the proposed defense strategies.

C. Performance of Our Proposed Defense Approaches



(a) PDR speeding up for online attack



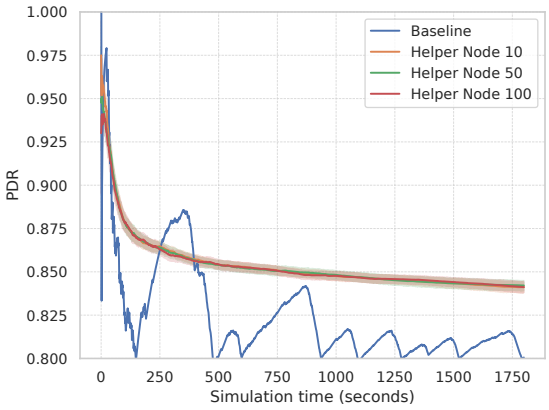
(b) PDR speeding up for offline attack

Fig. 6. PDR for Speeding Up parameters.

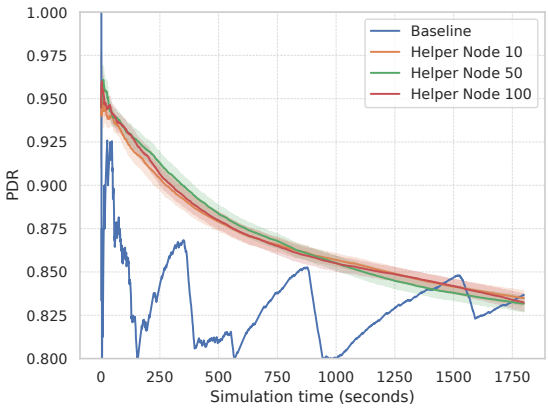
1) Performance of Speeding up: To better understand the advantage of the speeding up defense strategy,

we initially describe its behavior and compare it with that of online and offline MAB attacks. The speeding up strategy envisions channel hopping occurring on a predefined time basis or detection of a drop in the PDR. In Fig. 6, the parameter (e.g., 50) is the interval of channel change, and when the speeding up parameter value is set to 0, it means that no speeding up is deployed.

In Fig. 6, the PDR of the network is shown when considering random channel hopping with different speed-up parameters. In particular, we compare the various speeding up settings values, e.g., $50, 100, 300$, which mean the update intervals for each test, and those without speeding up (i.e., baseline). We can see that the speeding up strategy can improve the PDR of the network, and for update interval $= 50$, the network achieves a PDR of 0.93 under online and offline attacks. Meanwhile, the PDR for the baseline setting is lower than 0.9. The speeding up of 100 and 300 is all better than the baseline.



(a) PDR Helper Node for online attack

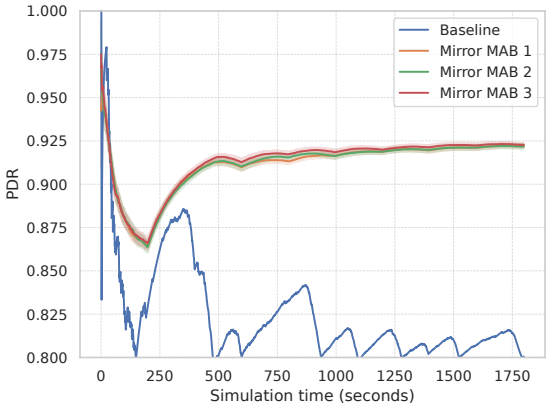


(b) PDR Helper Node for offline attack

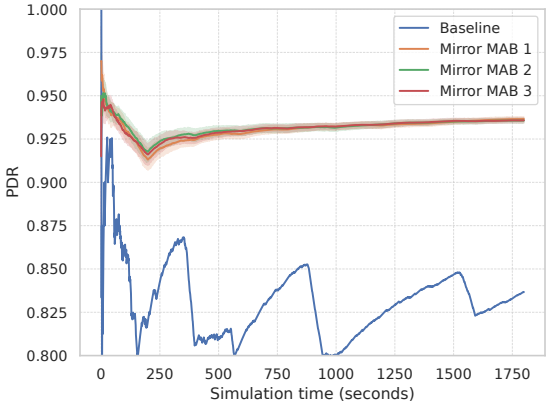
Fig. 7. PDR for Helper Node

2) Performance of Helper Node: In this section, we show the results of the helper node strategy with different parameters. We notice that the helper node defense does not depend on the specific values from Fig. 7, as the victim just needs to move to the successive channel, regardless of how the successive channel index is selected.

As shown in Fig. 7a and Fig. 7b, the PDR curves are shown when considering different helper node values for the online and offline attacks, respectively. We see that the helper node strategy is helpful in mitigating the attack effects, as the victim can improve the PDR with the defense strategy. From Fig. 5, Fig. 7a and Fig. 7b, we notice that under online and offline attackers, using helper node with different helper node settings, e.g., $10, 50, 100$, can improve the performance of PDR, and keep the PDR around $0.825 \sim 0.85$.



(a) PDR mirror MAB for online attack



(b) PDR mirror MAB for offline attack

Fig. 8. PDR for mirror MAB

3) Performance of Mirror MAB: Fig. 8 presents the PDR over time for both online and offline attack

scenarios, comparing the baseline approach with the proposed mirror MAB strategy with different settings, e.g., $1, 2, 3$, which means the number of vulnerable channels predicted by the mirror MAB algorithm. In Fig. 8, the baseline approach suffers from significant PDR degradation, dropping below other settings before stabilizing. This instability is due to its inability to adapt to jamming dynamically. In contrast, the mirror MAB strategies maintain a higher and more stable PDR, around 0.92 and 0.93 under the online and offline attack, respectively. However, the initial drop is less severe, as offline attacks allow for a more predictable jamming pattern. The mirror MAB strategies consistently outperform the baseline, leveraging historical data for more informed channel selection. While not as adaptive as online MAB, the offline approach still provides a robust defense against attacks. Across both settings, mirror MAB implementations achieve close improvement over the baseline.

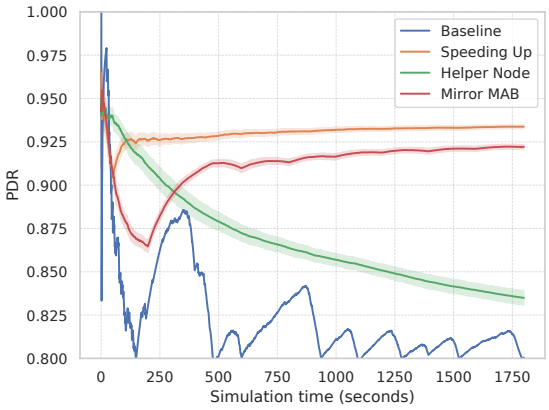


Fig. 9. PDR for different strategies

4) Summary: In Fig. 9, we illustrate the PDR performance over time for different strategies against the MAB attack: *baseline*, *speeding up*, *helper node*, and *mirror MAB*. The horizontal axis represents the simulation time in seconds, while the vertical axis shows the achieved PDR.

Overall, we observe that the *speeding up* strategy consistently surpasses the other methods in reaching higher PDR values more rapidly. Its aggressive channel-switching mechanism allows it to evade jamming attempts sooner, translating into a faster recovery of communication reliability. Meanwhile, *helper node* employs additional resources in the network to counteract jamming, helping to maintain a robust PDR once the system stabilizes. The initial recovery slope for *helper node* is slower than *speeding up*, which means a long period of development of the defense strategy to mitigate smart attacks.

By contrast, the *mirror MAB* approach benefits from a dynamic learning framework that updates its defense strategy based on observed rewards. Despite a slightly slower reacting phase compared with *speeding up*, *mirror MAB* proves effective in converging to a high PDR, reflecting its ability to track and counter evolving jamming tactics adaptively. In all cases, once the strategies stabilize, the victim sustains near-optimal PDR levels despite ongoing jamming. Thus, the results confirm that both *speeding up* and *mirror MAB* yield rapid improvements.

VII. Conclusion

Jamming attacks represent a serious threat to wireless communications security. In particular, MAB-based attackers leveraging smart jamming strategies to follow channel hopping patterns can undermine the effectiveness of some defense mechanisms. Our study demonstrated that both *online* and *offline* variants of MAB-based attacks can rapidly learn the hopping behavior of typical IEEE 802.11 deployments. To address this challenge, we proposed and evaluated three countermeasures: *speeding up*, *helper node*, and *mirror MAB*, and demonstrated their effectiveness in mitigating the sustained impact of MAB-based attacks.

Although the results are encouraging, several avenues are invited for further investigation. In future works, we could implement the proposed defenses on a hardware testbed in order to assess their practicality, overhead, and robustness under real-world propagation conditions. Furthermore, we will explore the possibility of using more RL algorithms as a friendly jamming mechanism to exclude possible network intruders.

References

[1] W. H. Hassan *et al.*, "Current research on internet of things (iot) security: A survey," *Comput. Netw.*, vol. 148, pp. 283–294, 2019.

[2] F. Guo, F. R. Yu, H. Zhang, X. Li, H. Ji, and V. C. M. Leung, "Enabling massive iot toward 6g: A comprehensive survey," *IEEE Internet of Things J.*, 2021.

[3] Z. Zhang and M. Krunz, "Sigtam: A tampering attack on wi-fi preamble signaling and countermeasures," in *Proc. IEEE Conf. Commun. Netw. Security (CNS)*, 2022, pp. 1–9.

[4] W. Xu, "Jamming attack defense," in *Encyclopedia of Cryptography, Security and Privacy*. Springer, 2025, pp. 1318–1325.

[5] K. Grover, A. Lim, and Q. Yang, "Jamming and anti-jamming techniques in wireless networks: a survey," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 17, no. 4, pp. 197–215, 2014.

[6] A. Gil-Martínez, M. Poveda-García, J. A. López-Pastor, J. C. Sánchez-Aarnoutse, and J. L. Gómez-Tornero, "Wi-fi direction finding with frequency-scanned antenna and channel-hopping scheme," *IEEE Sensors J.*, vol. 22, no. 6, pp. 5210–5222, 2022.

[7] E. Staddon, "Threat detection, identification and quarantine in wireless iot based critical infrastructures," Ph.D. dissertation, Université de Lille, 2022.

[8] H. Dutta, A. Kumar Bhuyan, and S. Biswas, "Using multi-armed bandit learning for thwarting mac layer attacks in wireless networks," *IEEE Trans. on Netw*, vol. 33, no. 1, pp. 327–339, 2025.

[9] D. Ryoo, Y. Yoo, J. Paek, and S. Bahk, "Spade: Secure periodic advertising using coded time-channel rendezvous for ble audio," in *Proc. Int. Conf. Distributed Comput. Smart Syst. Internet Things (DCOSS-IoT)*, 2023, pp. 39–46.

[10] E. Bout, A. Brighente, M. Conti, and V. Loscri, "Folpetti: A novel multi-armed bandit smart attack for wireless networks," in *Proc. ARES*, 2022, pp. 1–10.

[11] V. Loscri and M. Biagi, "Libero: Light bias as effective countermeasure against eavesdropper attacks," *IEEE Trans. Commun.*, vol. 72, no. 12, pp. 7882–7893, 2024.

[12] D. Ma, Y. Wang, and S. Wu, "Against jamming attack in wireless communication networks: A reinforcement learning approach," *Electronics*, vol. 13, no. 7, p. 1209, 2024.

[13] Z. Zhou, C. Dong, D. Mo, and P. Zheng, "Privacy-preserving decision making based on q-learning in cloud computing," in *2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2022, pp. 727–732.

[14] X. Tang, P. Ren, and Z. Han, "Jamming mitigation via hierarchical security game for iot communications," *IEEE Access*, vol. 6, pp. 5766–5779, 2018.

[15] N. Namvar, W. Saad, N. Bahadori, and B. Kelley, "Jamming in the internet of things: A game-theoretic perspective," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2016, pp. 1–6.

[16] B. Upadhyaya, S. Sun, and B. Sikdar, "Machine learning-based jamming detection in wireless iot networks," in *Proc. IEEE VTS Asia Pacific Wireless Commun. Symp. (APWCS)*, 2019, pp. 1–5.

[17] G. Han, L. Xiao, and H. V. Poor, "Two-dimensional anti-jamming communication based on deep reinforcement learning," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2087–2091.

[18] L. Xiao, X. Wan, W. Su, Y. Tang *et al.*, "Anti-jamming underwater transmission with mobility and learning," *IEEE Commun. Lett.*, vol. 22, no. 3, pp. 542–545, 2018.

[19] V. Toldov, L. Clavier, V. Loscrì, and N. Mitton, "A thompson sampling approach to channel exploration-exploitation problem in multihop cognitive radio networks," in *Proc. IEEE Int. Symp. Personal, Indoor and Mobile Radio Commun. (PIMRC)*. IEEE, 2016, pp. 1–6.

[20] F. Shi, Z. Zhou, J. Guo, R. Li, Z. Zhang, S. Li, Q. Liu, and X. Bao, "Lightrl-ad: A lightweight online reinforcement learning approach for autonomous defense against network attacks," in *2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2024, pp. 1614–1621.

[21] X. Liu, Y. Xu, L. Jia, Q. Wu, and A. Anpalagan, "Anti-jamming communications using spectrum waterfall: A deep reinforcement learning approach," *IEEE Commun. Lett.*, vol. 22, no. 5, pp. 998–1001, 2018.

[22] X. Wang, Y. Xu, J. Chen, C. Li, X. Liu, D. Liu, and Y. Xu, "Mean field reinforcement learning based anti-jamming communications for ultra-dense internet of things in 6g," in *Proc. IEEE Int. Conf. Wireless Commun. Signal Processing (WCSP)*. IEEE, 2020, pp. 195–200.

[23] Q. Zhou, Y. Li, and Y. Niu, "Intelligent anti-jamming communication for wireless sensor networks: A multi-agent reinforcement learning approach," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 775–784, 2021.

[24] J. Xu, H. Lou, W. Zhang, and G. Sang, "An intelligent anti-jamming scheme for cognitive radio based on deep reinforcement learning," *IEEE Access*, vol. 8, pp. 202 563–202 572, 2020.

[25] E. Bout, V. Bout, A. Brighente, M. Conti, and V. Loscri, "Evaluation of channel hopping strategies against smart jamming attacks," in *Proc. IEEE Int. Conf. Communications (ICC)*, 2023.

[26] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proc. ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2005, pp. 46–57.

[27] A. Aimi, F. Guillemin, S. Rovedakis, and S. Secci, "Packet delivery ratio guarantees for differentiated lorawan services," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2022, pp. 2014–2019.

[28] D. Russo, B. Van Roy, A. Kazerouni, I. Osband, and Z. Wen, "A tutorial on thompson sampling," *arXiv preprint arXiv:1707.02038*, 2017.

[29] S. Agrawal and N. Goyal, "Analysis of thompson sampling for the multi-armed bandit problem," in *Proc. Conf. on Learning Theory (COLT)*, 2012, pp. 39.1–39.26.

[30] J. Havil, *Gamma: Exploring Euler's Constant*. Princeton University Press, 2003.

[31] F. Liu and N. Shroff, "Data poisoning attacks on stochastic bandits," in *Proc. Int. Conf. Machine Learning (ICML)*. PMLR, 2019, pp. 4042–4050.