DDoS Attacks Mitigation in 5G-V2X Networks: A Reinforcement Learning-Based Approach

Badre Bousalem, Mohamed Anis Sakka, Vinicius Silva, Wael Jaafar, Asma Ben Letaifa, Rami Langar

DDoS Attacks Mitigation in 5G-V2X Networks: A Reinforcement Learning-based Approach

Badre Bousalem*, Mohamed Anis Sakka†‡, Vinicius F. Silva*, Wael Jaafar†, Asma Ben Letaifa‡, Rami Langar*†
*University Gustave Eiffel, LIGM-CNRS UMR 8049, F-77454, Marne-la-Vallée, France
†Software and IT Engineering Department, École de Technologie Supérieure (ÉTS), Montréal, QC H3C1K3, Canada
‡Higher School of Communication of Tunis (Sup'Com), Mediatron Lab., Tunis, Tunisia
E-mails: {badre.bousalem, vinicius.fonsecaesilva, rami.langar}@univ-eiffel.fr ; mohamed-anis.sakka.1@ens.etsmtl.ca ;
{wael.jaafar, rami.langar}@etsmtl.ca ; {mohamedanis.sakka, asma.benletaifa}@supcom.tn

***Abstract*—Vehicle-to-Everything (V2X) communication standards, which mainly rely on the 5G New Radio (NR) technology, can be subject to attacks such as Distributed Denial of Service (DDoS), which flood the network with non-expected control information. This causes network performance degradation and leads to accidents involving vehicles and/or vulnerable road users. A potential approach to mitigate DDoS attacks is to isolate the hijacked vehicular users in sinkhole-type slices that contain a small amount of network resources. Nevertheless, DDoS attacks may be unpredictable since it can modify its communication protocol for example, which makes it difficult to determine the proper moment to release mitigated users from the sinkhole-type slices once the security breach ceases to exist. In such a context, we propose a Reinforcement Learning-based approach that evaluates multiple types of DDoS attacks on sinkhole-type slices and estimates the optimal time to keep a mitigated user in such a slice before releasing it. The proposed approach is trained and tested with a dataset collected from a 5G-V2X testbed. Results show that our approach outperforms a benchmark of random actions, in terms of the mean cumulative reward and error over time.**

***Index Terms*—5G-V2X, attack mitigation, reinforcement learning.**

I. Introduction

Vehicular systems integrated into slice-powered fifth-generation (5G) and beyond networks bring a new set of applications and network services to mobile users. The communication channels between vehicles and roadside infrastructure, as well as vulnerable road users (VRUs) are particularly important to ensure their safety and improve the overall efficiency of our roads. Vehicle-to-Everything (V2X) communication standards come to fulfill these requirements [1], and an important ally to V2X is the use of network slicing, which creates multiple logical instances of the physical network, called "network slices".

From a security perspective, 5G-V2X communications can be prone to cyberattacks such as Distributed Denial of Service (DDoS), which floods V2X communication interfaces with different communication protocols (e.g., *TCP*, *UDP* and *ICMP*) that are not expected by any entity in the 5G-V2X network. Network performance degradation in terms of throughput, delay, or service availability, may happen as a consequence of such attacks.

Based on our previous work [2, 3], a potential approach to mitigate DDoS attacks is isolating the attack sources in sinkhole-type slices, where a small amount of physical resource blocks (PRBs) is reserved, thus limiting the attackers' actions. However, a smart DDoS attack behavior prevents network entities from easily estimating the proper time to release users from sinkhole-type slices, in cases where the mitigated device is used as a victim by external malicious entities. DDoS attacks have been widely discussed in the literature, however, their application in the 5G-V2X context has not been thoroughly explored [2–4].

To this end, we propose in this paper a Reinforcement Learning (RL)-based solution to evaluate multiple types of DDoS attacks in 5G-V2X networks, in the context of mitigated users in a sinkhole-type slice. Specifically, by evaluating the mitigated users' behavior, our RL agent defines the optimal time to release them from the sinkhole-type slice, once they do not represent a threat to benign users anymore. To do so, we first collect a dataset using our 5G-V2X testbed that is mainly focused on detecting and mitigating cybersecurity attacks, such as radio jamming and DDoS [2, 3]. Then, based on the obtained dataset, we train different RL agents. Our approach has shown promising results, outperforming a benchmark of random actions in terms of the mean cumulative reward and error over time, thus showcasing better convergence and stability through time.

The remainder of this paper is organized as follows. Section II discusses the related work. Section III elaborates on the 5G-V2X testbed used to collect data and train the RL agents. Section IV describes the methodology adopted to collect our dataset and provides details about the RL environment and the investigated algorithms. Finally, Section V illustrates the performance results, while Section VI concludes the paper.

II. Related Work

Several works have been proposed in the literature to optimize security solutions whilst considering 5G network requirements. However, there is a lack of works that rely on

slice-powered 5G-V2X networks and RL-based approaches to detect and mitigate cyberattacks, such as DDoS.

Authors of [5] proposed an optimization model to proactively mitigate DDoS attacks in the 5G Core Network (CN) through on-demand intra/inter slice isolation, to guarantee network performance requirements for 5G CN slices. This work focuses only on protecting 5G CN slicing, where an on-demand network service/function distribution between slices occurs. In our work, we rather consider slicing on the Radio Access Network (RAN), where physical resources could be shared between users. Similarly, authors in [6] proposed *DeepSecure*, a framework that used a Long Short Term Memory (LSTM) Deep Learning (DL)-based model to classify users' network traffic as DDoS attack or benign, as well as a model that predicted the appropriate slice for users previously classified as benign. The main drawback of the proposed method is the use of a dataset that is not directly related to a 5G-based environment. Finally, in [7], the authors proposed a hierarchical RL-based cooperative attack detection system to protect 5G wireless systems from advanced attacks such as jamming and DDoS attacks. The detection system is spread across key network components like access points, base stations, and servers. It is designed to identify the key characteristics associated with these attacks and detect them.

The aforementioned works did not consider the detection and mitigation of DDoS attacks in 5G-V2X communication interfaces. Additionally, none of them used data collected from a realistic 5G-V2X testbed, as we propose in this work to train our RL agent.

III. 5G-V2X Testbed Description

Fig. 1 shows the base hardware/software resources deployed in our 5G-V2X testbed. Our work mainly relies on a 5G standalone (SA) setup consisting of a 5G CN and a 5G base station (gNodeB) at the RAN. To emulate these components, we use *OpenAirInterface* (OAI) [8].

Our prototype also makes use of *FlexRIC* [9], a software-defined controller that allows users to flexibly monitor and control the RAN components over time, mainly allowing to share the network's resources among users in the shape of network slices, the latter being composed of one or more PRBs.

On top of *FlexRIC*, we also developed a northbound Software-Defined Network (SDN) application, a.k.a. "Slicing APP" application, which enables the network administrator to deploy network slicing policies in a user-friendly and abstracted manner. In this work, we consider two main policies that operate simultaneously. The first one is a DL-based attack detection agent that allows the RAN to detect DDoS attacks in real-time, then move the mitigated users to a sinkhole-type slice [2, 3]. The second policy consists of the RL-based approach proposed in this work, which focuses on the real-time evaluation of mitigated users within the sinkhole-type slices to decide when they can be moved back to the benign slice.

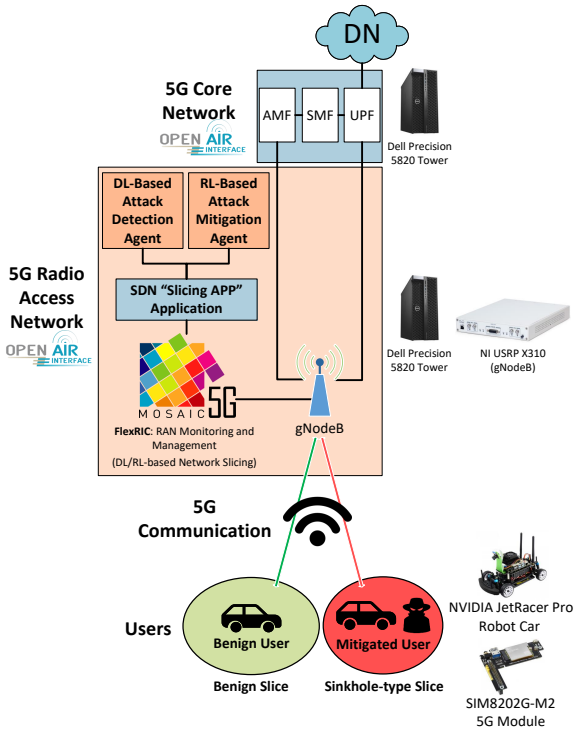

Fig. 1: 5G-V2X testbed's base hardware/software resources.

The 5G CN and 5G RAN components are executed in separate machines. Specifically, two Dell Precision 5820 (Intel Xeon W-2265 3.50GHz with 12 cores, 128GB of RAM) are used. The RAN machine is connected to a USRP X310 card, which is responsible for emulating the gNodeB, thus creating a communication interface between the 5G RAN and the User Equipments (UEs). Finally, the UEs are represented by two JetRacer Pro robot cars based on the NVIDIA Jetson Nano platform [10]. To connect with the 5G RAN, each robot car integrates a SIMCom SIM8202G-M2 5G module [11].

Our experiments generate benign traffic between the 5G CN and one of the UEs at a fixed throughput. To do that, we use the *iperf3* tool [12], where the 5G CN is set as the client, and one of the UEs is set as the server. On the other hand, we set up the other UE to be responsible for deploying the DDoS attacks using the *Mausezahn* tool [13], which targets the 5G CN's User Plane Function (UPF) module.

IV. Methodology

This section presents the adopted methodology to collect the traffic dataset, as well as to train our RL agents to evaluate the DDoS attacks. Specifically, we describe here the created RL environment for training, in order to estimate the duration of a DDoS attack and release time of mitigated users from the sinkhole-type slice. For the sake of comparison and to provide a robust insight regarding the addressed problem, we train multiple types of RL agents and evaluate their performances mainly in terms of cumulative rewards, and under different DDoS attack scenarios.

TABLE I: Data collection parameters for the DDoS scenarios.

Parameter	Value
Mausezahn's *c* value (duration of one attack burst)	1, 2, 3, 4, 5, 6
Interval between Two Attack Bursts	5, 7, 10 seconds
DDoS Source IP Address Range	12.1.1.1-12.1.1.62
DDoS Source Port Range	500-1000
DDoS Destination IP Address	12.1.1.1
DDoS Destination Port	80
Benign Traffic Throughput	30 Mbps
Data Collection Duration per Scenario	1 minute

A. Dataset Collection

To provide a complete and realistic DDoS traffic dataset, we defined several scenarios for traffic generation. Each scenario is characterized by its specific DDoS attack behavior, which in turn is defined by two main parameters: 1) The duration of one attack burst, and 2) The interval between two attack bursts. For each scenario, the UEs and the gNodeB are placed at fixed locations, and benign traffic generation is triggered simultaneously with the DDoS attacks. In Table I, we illustrate the main parameters' values defined for the different DDoS attack scenarios.

In addition to the DDoS attack scenarios, we include in our dataset two benchmark scenarios: 1) No DDoS attacks are executed, i.e., only *iperf3* runs between the 5G CN and the UEs, and 2) DDoS attacks are deployed randomly, i.e., with random burst duration and interval values.

B. Reinforcement Learning Environment

RL can be used to learn an optimal strategy for an operational environment through experience and rewards [14]. Specifically, a RL agent aims to learn an optimal strategy $\rho : S \rightarrow A$ to obtain a maximum reward [15, 16].

In our system, the main target is to estimate the attack duration, denoted d_t, and based on it, decide when to move a mitigated UE from the sinkhole-type slice to its operating benign slice. For the sake of simplicity, we assume that an attack duration d_t is within the discretized range of 1 to 100 seconds. In order to assess the RL agent's performance in different scenarios, we divide all possible values into four ranges D_i, $\forall 1 \leq i \leq 4$, where $D_1 = [1, 25]$, $D_2 = [25, 50]$, $D_3 = [50, 75]$, and $D_4 = [75, 100]$. At any time, an attack duration d_t is randomly selected from a range D_i based on the uniform distribution.

Subsequently, the main components of our RL system can described as follows:

- **State:** The set of states is defined as S, where a single state, denoted as s_t, corresponds to the duration between two consecutive attack bursts (in seconds). For the sake of simplicity, we assume that s_t is discrete and defined within the range 1 to 200 seconds. The value for each state is decomposed into two time durations, defined as $s_t = d_{t-1} + n_{t-1}$, where d_{t-1} is the duration of the previous attack, and n_{t-1} is the time spent in the benign slice before generating the new attack. The variable n_t follows the Poisson Point Process (PPP) with distinct values of rate λ. The value of λ defines the cadence of the attacks and is selected within the set $\{20, 40, 60, 80\}$ for different DDoS attack scenarios.
- **Action:** The set of actions is denoted A, where a single action, a_t, represents the agent's decision on when to restore the mitigated UE (victim) to the benign slice. A is defined with a set of regularly spaced values within the discrete range of 1 to 100 seconds. The spacing step between the action values in this set is defined as μ, where $\mu \in \{1, 2, 5, 10\}$. For instance, when $\mu = 1$, the actions set is defined as $A_1 = \{1, 2, \ldots, 100\}$, while for $\mu = 10$, it is defined as $A_{10} = \{1, 10, 20, \ldots, 100\}$.
- **State Transition Probability:** The expected probability of transitioning from current state s_t to the next state s_{t+1} after the execution of action a_t, is defined as $P(s_{t+1}|s_t, a_t)$; This probability depends on the used PPP with parameter λ to generate the attacks.
- **Reward:** After performing an action, the RL agent is given an immediate reward r_t. In our system, we define the value of r_t as follows:

$$r_t = \frac{1}{|a_t - d_t| + 1}, \tag{1}$$

where $|\cdot|$ is the absolute value operator. In other words, r_t is equivalent to the inverse of the absolute value of the difference between action a_t and duration d_t, where a_t represents the estimated attack duration and d_t is the attack's real duration. The denominator is increased by 1 in order to avoid division by zero. The reward increases when the agent's estimation is closer to the real duration, hence indicating a higher efficiency. The use of a RL agent aims to select the policy that maximizes the discounted cumulative reward R during the observation period T. It can be expressed by

$$R = \max_{\pi} \mathbb{E}\left(\sum_{t=0}^{T-1} \gamma^t r_t\right), \tag{2}$$

where π is the selected policy (action), $\mathbb{E}(\cdot)$ is the expectation function, and $\gamma \in [0, 1]$ is the discount factor that determines the importance of future rewards [17].

C. Reinforcement Learning Algorithms

In this work, we based our solutions on four RL algorithms, in combination with variations of the environment parameters λ, μ, and D_i. The considered RL algorithms are detailed below:

- ***Q*-learning:** It is a model-free RL algorithm that seeks the optimal policy for a RL agent by learning the optimal action-value function, or Q-function [18]. The Q-function estimates the expected future reward of taking a particular action in a given state.

- ***Deep Q-Network:*** Deep-Q-Network, a.k.a., DQN, is a model-free RL algorithm that combines Q-learning with Deep Neural Networks (DNNs) to learn a policy for an agent in a given environment [19]. The Q-function is approximated by a DNN that takes the state as input and outputs the expected future reward for each action.
- ***Advantage Actor-Critic:*** This algorithm, also called A2C, combines policy-based and value-based methods in RL. It utilizes two neural networks: 1) The Actor for action selection based on the current policy, and 2) The Critic for estimating action quality through a value function [20].
- ***Proximal Policy Optimization:*** This algorithm, also called PPO, is an iterative method that collects data through interactions with the environment, computes benefits for each action and optimizes the policy using multiple updates. It creates a balance between exploration and exploitation, resulting in effective learning and policy convergence over time [21].

V. Performance Results

In this section, we present the performance results of our solutions based on the previously defined RL algorithms, given different environment conditions. We also realize a performance comparison with a benchmark solution defined with an agent that randomly selects actions, i.e., without any learning, training, or decision criteria.

We consider two performance metrics: 1) The Mean Cumulative Reward, which is computed by summing up the immediate rewards over the episodes (in our case 1000 episodes), and 2) The Mean Error, or Loss Function, which quantifies the difference between the agent's selected action, i.e., the estimated duration (in seconds) of the current DDoS attack, and the real duration of the attack.

A. Simulation-based Results

We present here the results obtained with the considered RL algorithms (described in Subsection IV-C) for several DDoS attack scenarios, i.e., defined with different λ, μ, or D_i, $\forall i \in [1, 4]$. Unless stated otherwise, the RL-based solutions are run for 100,000 episodes, with a discount factor γ starting at 0.9 and then gradually decreasing to 0.1 through the episodes. Also, the replay buffer capacity is set to 1,000 and the mini-batch size is fixed to 64. Finally, for the DNN-based algorithms, the learning rate is set to 10^{-3}.

In Fig. 2, we present the mean cumulative reward performances achieved by the PPO, A2C, DQN, and Q-learning-based solutions, as well as the benchmark method, versus the episodes, and given A_5, D_2, and $\lambda = 40$. The cumulative rewards consistently increase over time, which shapes the convergence behavior of the solutions. Indeed, all RL-based solutions significantly outperform the benchmark performance, while presenting similar results. However, among them, Q-learning stands out for its fast convergence and superior stability through time. Given that the context of 5G-V2X is very dynamic, stability is an important criterion to consider for deploying adequate RL solutions in a real environment.

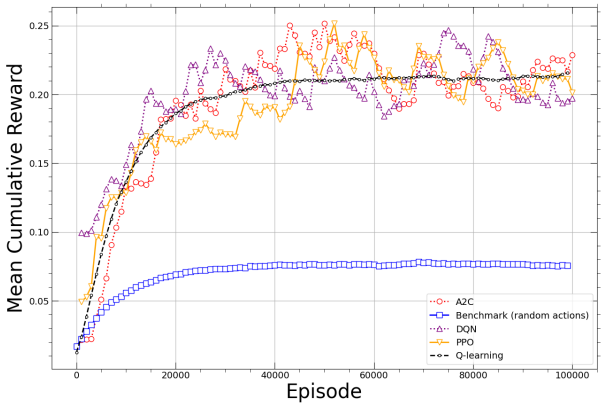

Fig. 2: Mean cumulative reward vs. episodes (different solutions).

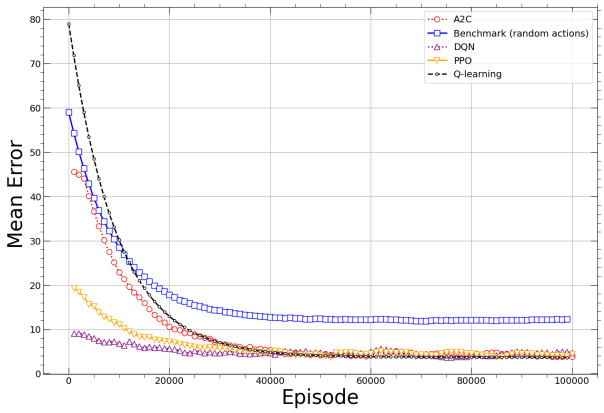

Fig. 3: Mean error vs. episodes (different solutions).

Fig. 3 shows the mean error performance (in seconds) for the same approaches and system conditions as Fig. 2. As it can be seen, for any method, the mean error is very high at first, then it decreases exponentially to be stabilized after approximately 40,000 episodes. Indeed, the benchmark stabilizes at a mean error value of 14 seconds, while the RL-based solutions are capable of reducing the mean error to approximately 3.8 seconds. Since the considered scenario uses D_2 with a mean attack duration of 37.5 seconds, then the error represents an average 10% for the RL-based approaches, but more than 37% for the benchmark.

In the aforementioned experiments, we built the RL environment, in particular the states, by generating the latter based on the PPP for the time of arrival of a new DDoS attack (n_t) and based on the uniform distribution in D_i for the attack duration (d_t). In contrast, in the remaining, we will base our experiments on the collected data from our 5G-V2X testbed. For the sake of simplicity, we decided to deploy only the Q-learning-based approach, given its proven superior performances, stability, and suitability for the dynamic 5G-V2X environment.

TABLE II: Attack duration estimation in 5G-V2X testbed (in seconds).

Estimated attack duration of RL-based method	5G-V2X attack real duration
6.1	5.7
10.4	10.6
8.5	8.1
12.1	12
11.6	11.4
5.0	4.9
3.0	3.3
14	13.8
11.3	11.5
9.7	9.6

B. 5G-V2X Testbed-based Results

Next, we built an experiment where ten bursts of DDoS attacks with random durations are sequentially generated by a malicious UE in the 5G-V2X testbed. With our Q-learning-based solution deployed on the platform (D_1, $\mu = 0.1$), we attempt to estimate the attacks' real durations and trigger the moving of the mitigated UE back to the benign slice. Obtained results are presented in Table II. Clearly, the Q-learning-based approach is capable of accurately estimating the attack duration. On average, the error is less than 3.2%. Hence, the proposed approach is suitable for practical deployments in 5G-V2X environments.

VI. Conclusion

In this work, we introduced novel RL-based solutions to mitigate DDoS attacks in 5G-V2X networks, which consider users isolated in sinkhole-type slices as victims of external attack sources. To evaluate the effectiveness of our proposed methods, we built an OAI-based 5G-V2X testbed to collect the necessary dataset for training and testing our approaches. The performances of the RL-based solutions were compared with a benchmark of randomly selected actions, in terms of the mean cumulative reward and mean error. The obtained results show that our RL-based methods outperform the benchmark and that the Q-learning-based approach exhibits high stability in action selection for different environment parameters. Hence, the Q-learning-based solution is seen as the optimal choice for real 5G-V2X environments.

In future work, we intend to explore novel DL and RL-based techniques to detect and mitigate inter-slice and intra-slice DDoS attacks. We also plan to extend the scope of the proposed RL-based solution by making it distributed using Federated Learning. Finally, we will extend our dataset by adding new features and DDoS-based scenarios, then make it publicly available to enable the research community to reproduce our experiments.

Acknowledgment

This work was supported in part by the ANR 5G-INSIGHT project (Grant no. ANR-20-CE25-0015), and in part by the Innovation for Defence Excellence and Security (IDEaS) program of the Department of National Defence (DND) Canada.

References

[1] A. Alalewi, I. Dayoub, and S. Cherkaoui, "On 5G-V2X Use Cases and Enabling Technologies: A Comprehensive Survey," *IEEE Access*, vol. 9, pp. 107 710–107 737, 2021.

[2] B. Bousalem, V. F. Silva, R. Langar, and S. Cherrier, "Deep learning-based approach for DDoS attacks detection and mitigation in 5G and beyond mobile networks," in *Proc. IEEE Int. Conf. Netw. Softwar. (NetSoft)*, 2022, pp. 228–230.

[3] B. Bousalem, V. F. Silva, R. Langar, and S. Cherrier, "DDoS attacks detection and mitigation in 5G and beyond networks: A deep learning-based approach," in *Proc. IEEE Glob. Commun. Conf.*, 2022.

[4] A. Boualouache and T. Engel, "A survey on machine learning-based misbehavior detection systems for 5G and beyond vehicular networks," *IEEE Commun. Surv. Tuts.*, pp. 1–1, 2023.

[5] D. Sattar and A. Matrawy, "Towards secure slicing: Using slice isolation to mitigate DDoS attacks on 5G core network slices," in *Proc. of IEEE Conf. Commun. Netw. Sec. (CNS)*, 2019, pp. 82–90.

[6] N. A. E. Kuadey, G. T. Maale, T. Kwantwi, G. Sun, and G. Liu, "DeepSecure: Detection of distributed denial of service attacks on 5G network slicing—deep learning approach," *IEEE Wireless Commun. Lett.*, vol. 11, no. 3, pp. 488–492, 2022.

[7] H. Sedjelmaci, "Attacks detection approach based on a reinforcement learning process to secure 5G wireless network," in *Proc. IEEE Int. Conf. Commun. Wrkshps. (ICC Wrkshps.)*, 2020, pp. 1–6.

[8] Eurecom, "OpenAirInterface." [Online]. Available: https://openairinterface.org/

[9] R. Schmidt, M. Irazabal, and N. Nikaein, "FlexRIC: An SDK for next-generation SD-RANs," in *Proc. Int. Conf. Emerg. Network. Exper. Technol.*, New York, NY, USA, 2021, p. 411–425. [Online]. Available: https://doi.org/10.1145/3485983.3494870

[10] Waveshare, "JetRacer Pro Robot Car based on the Jetson Nano Platform." [Online]. Available: https://www.waveshare.com/jetracer-pro-ai-kit.htm

[11] Waveshare, "SIMCom SIM8202G-M2 5G module." [Online]. Available: https://www.waveshare.com/sim8202g-m2-5g-for-jetson-nano.htm

[12] ESnet, "iperf3." [Online]. Available: https://downloads.es.net/pub/iperf/

[13] Netsniff-ng, "Mausezahn." [Online]. Available: http://netsniff-ng.org/

[14] J. Parras and S. Zazo, "Learning attack mechanisms in wireless sensor networks using Markov decision processes," *Expert Systems with Applications*, vol. 122, pp. 376–387, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417419300235

[15] K. Maneenil and W. Usaha, "Preventing malicious nodes in ad hoc networks using reinforcement learning," in *Proc. Int. Symp. Wireless Commun. Syst.*, Sep 2005.

[16] J. Yang, H. Zhang, C. Pan, and W. Sun, "Learning-based routing approach for direct interactions between wireless sensor network and moving vehicles," in *Proc. IEEE Int. Conf. Intelli. Transport. Syst. (ITSC)*, Oct. 2013, pp. 1971–1976.

[17] A. Moussaid, W. Jaafar, W. Ajib, and H. Elbiaze, "Deep reinforcement learning-based data transmission for D2D communications," in *Proc. Int. Conf. Wireless Mob. Comput. Network. Commun. (WiMob)*, 2018.

[18] B. Jang, M. Kim, G. Harerimana, and J. Kim, "Q-learning algorithms: A comprehensive classification and applications," *IEEE Access*, Sep 2019.

[19] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing atari with deep reinforcement learning," *CoRR*, vol. abs/1312.5602, 2013. [Online]. Available: http://arxiv.org/abs/1312.5602

[20] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," *CoRR*, vol. abs/1602.01783, 2016. [Online]. Available: http://arxiv.org/abs/1602.01783

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347