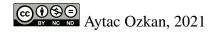# Adaptive Collaborative Autonomous Wireless Networks

by

Aytac OZKAN

MANUSCRIPT-BASED THESIS PRESENTED TO ÉCOLE DE
TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, "DEPOSIT DATE"

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Prof. Dr. Kim Khoa Nguyen, Thesis supervisor
Department of Electrical Engineering and University of Quebec

M. Pr. Louis Rivest, President of the board of examiners
PhD Program's Director

M. First Name Last Name, External examiner
Department and institution

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON "DEFENSE DATE"

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

FOREWORD

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

ACKNOWLEDGEMENTS

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

French title

Aytac OZKAN

RÉSUMÉ

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Mots-clés: mot-clé1, mot-clé2

Adaptive Collaborative Autonomous Wireless Networks

Aytac OZKAN

ABSTRACT

Due to the tremendous improvements of technology, the world is more connected than ever at the human history. Mobile devices, cell phones, smart home solutions, autonomous cars, etc. These vehicles are usually using IEEE 802.15.4 communication protocols, the devices which uses this protocol have the limited number of communication channels and low transmit power, are especially susceptible to jamming attacks. For example, some Internet of things (IoT) devices (e.g., brain and heart inculcated IoT devices), jamming attacks can cause serous consequences for human health

Within this concern, to prevent this kind of intentional interference against wireless networks, we are going to employ self-learning algorithms such as deep reinforcement learning to develop a resilient, intelligent, and self-supervised anti-jamming framework.

Since The DeepMind has been introduced the Reinforcement Learning (RL) and Q-Learning algorithm H., A. & D. (2016), this tools become one of the major toolkit to develop mitigation and intelligent deceptions strategies to prevent against reactive jamming attacks. Despite it is a subset of machine learning Kasturi, Jain & Singh (2020),it is no need for long training times and large datasets, and this future is the key of its success at the field.

Keywords: reinforcement-Learning, transfer-learning,wireless-networks,anti-jamming,multi-agent, collaborative-learning

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

LIST OF ALGORITHMS

Page

LIST OF ABBREVIATIONS

ETS   École de Technologie Supérieure

ASC   Agence Spatiale Canadienne

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

a   Première lettre de l'alphabet

A   Première lettre de l'alphabet en majuscule

INTRODUCTION

The last decade has witnessed the rapid growth of Machine Learning (ML) applications in wireless networks thanks to its agility and efficacy, especially in dealing with uncertainty and dynamics in large-scale problems Sun, Peng, Zhou, Huang & Mao (2019) Bkassiny, Li & Jayaweera (2013) However, some recent studies have revealed that conventional ML solutions have shortcomings, especially when they are applied to solve emerging problems in wireless networks, due to the special characteristics of wireless communications, such as high mobility, dynamic environments, diverse connections, and interference. However, some recent studies have revealed that conventional ML solutions have shortcomings, especially when they are applied to solve emerging problems in wireless networks, due to the special characteristics of wireless communications, such as high mobility, dynamic environments, diverse connections, and interference.

Moreover, the performances of ML techniques mainly rely on the availability of training data, but acquiring a sufficient amount of data might be costly and time-consuming. Even if the training data are sufficient, conventional ML techniques usually require a long training time, which makes them impractical for many latency-sensitive applications. Apart from the training time issues, many wire- less devices, e.g., IoT devices, are constrained by their limited computing capacity, and thus they are unable to run high-complexity ML tasks. Moreover, many ML techniques actually create more wireless traffic demands because data have to be sent to a central node for training and processing. Besides causing higher communication overhead, sending raw data may also threaten network users' privacy because sensitive information, e.g., healthcare, is sent via the wireless networks.

To address these challenges, Transfer Learning (TL) has recently emerged as a highly effective solution. Unlike conventional ML techniques that are trained to solve a specific problem, TL leverages valuable knowledge from similar tasks and previous experiences to significantly

enhance the learning performance of conventional ML techniques. As a result, TL possesses various advantages over traditional ML approaches, which can be summarized as follows;

- Enhance the quality and quantity of training data: One of the most challenging tasks for conventional ML approaches is finding sufficient and high-quality data for the training process. TL can easily overcome this problem by selecting and transferring knowledge from similar domains with a large amount of high-quality data. As a result, TL has been considered a highly effective solution for ML-based wireless networks in the future.

- Speed-up learning processes: Instead of learning from scratch like conventional ML approaches, the training process in TL can be significantly sped up thanks to valuable knowledge shared from other similar domains and/or learned in the past. As a result, this can remarkably improve the learning rate, which is especially crucial for the development of ultra-low latency applications for future wireless networks.

- Reduce computing demand: Conventional ML approaches usually require a large amount of computing resources for the training processes. However, with TL, most of the data were trained by other source domains before the trained models are transferred to the target domain, thereby significantly reducing the computing demands for the training process at the target domain. This is particularly useful for wireless devices (e.g., smartphones and edge devices) as they usually have hardware constraints.

- Mitigate communication overhead: For TL approaches, instead of sending the raw data with large size, only knowledge, e.g., the weights of trained models, needs to be sent. As a result, the communication overhead can be significantly reduced for wireless networks.

- Protect data privacy: In TL, instead of learning from raw data from other domains, ones only need to learn from their trained models (expressed through weights), and thus data privacy can be protected. This feature of TL is very helpful for privacy-sensitive wireless applications such as healthcare and military communication networks.

CHAPTER 1

RESEARCH OBJECTIVES, MOTIVATION, RESEARCH QUESTIONS

1.1 Research Objectives

The main purpose of this research is to develop efficient, reliable, and relentless mitigation techniques against the jammer, particularly reactive and powerful ones by employing machine learning (ML) and wireless communication technologies.

To achieve the objective defined above, we are going to use the Frequency Hopping Spectrum Sensing (FHSS) technique. In the infinite time space, regarding the probability distribution of incoming signals, we are going to build the most accurate ML model to find out the optimal and feasible prediction for communication channels.

Of course, the first paragraph is specified only the core part of the communication node, but the particular contribution of this research is to develop multi-agent (multi-node) collaborative (transfer meta-learning) learning techniques to satisfy the constraints such as cost of learning (for each node), cost of transferring mitigation strategy. Predicate on these details, we would like to acquire an augmented Markov decision tuple for our ML model.

Therefore, we can divide our main objective into three subsections and each of them address a different research problem.

We assume in the wireless ad-hoc network, there are two communication node (CN) and one reactive jammer. Also CN1 and CN2 has data connection, which means they can transmit the data to each other. In addition, the ad-hoc network is multi-channel. Constraints, respectively CN's power storage (limit), channel bandwidth, and installed CPU or GPU power on the CN.

1.1.1 Objectives

- In the infinite time space, we are going to build a Machine Learning model which will take as input frequencies by sensing from the wireless network, and try to predict the

jammer next action (channel estimation), which literally means that try to determine the next communication channel will jam. And it will allow us to determine the jamming activity pattern in the ad-hoc network, and by analyzing these patterns we are going to concrete mitigation strategies. So, related research will propose novel techniques to answer the question below,

How to catalyze the jamming activity pattern and initialize effective and relentless mitigation strategies for communication nodes in wireless ad-hoc networks by using autonomous learning techniques ?

- In the first item, we have proposed a method which allows us to analyze the jammer behaviour pattern, and generate defence strategies for the CN based on this pattern. However, in the wireless spectrum we have two different CNs and we can have more nodes, in this case we may transfer the produced policy (or strategy) from one node to another when suitable conditions are satisfied. Moreover, collaborative learning can increase the chances to mitigate against reactive and strong jammers by saving time and energy of CNs. E.g, the cost of learning, cost of transmission, and accuracy of learning models have high importance roles in the decision model of the framework.

  Therefore, related research will address the question below, How the collaborative (transfer) learning can assist knowledge transmission between two different communication nodes in the wireless ad-hoc communication spectrum ?

- In second item we introduced collective (transfer learning) technique, but when a jamming attack hit the wireless network, there won't be any data transmission in this condition, each CNs have to run their own learning algorithm.

CHAPTER 2

LITERATURE REVIEW

CHAPTER 3

PROPOSED METHODOLOGIES

3.1   Preliminaries

3.1.1   Fundamentals of Transfer Learning

Transfer Learning, simply learned knowledge will be transferred from the source domain to the target domain to improve the learning process of the target task. Thus, in the following, we first present the definition of a "domain" and In our research problem, the source domain is the communication node (CN) that underwent the jammer attack, And the target domain is the likelihood closest unattacked communication node in the wireless network.

Definition 3.1. *"Domain: A domain $\mathcal{D}$ is defined by two parts: (i) a feature space $\mathcal{X}$ and (ii) a marginal probability distribution $\mathcal{P}(X)$ in which $X = \{x_{1,...,x_n}\} \in \mathcal{X}$ where n is the number of feature vectors in $\mathcal{X}$. As such $\mathcal{D} = \{\mathcal{X}, \mathcal{P}(X)\}$."*

Definition 3.2. *"Task: given domain $\mathcal{D}$, a task $\mathcal{T}$ is defined by two parts: (i) a label space $\mathcal{L}$ and (ii) a predictive function $f(.)$. The predictive function (or decision function) is learned from the feature vector and label space pairs $\{x_{i,\,l_i}\}$, with $x_i \in X$ and $l_i \in \mathcal{L}$. In other words, a task is defined by $\mathcal{T} = \{\mathcal{L},\ f(.)\}$."*

Definition 3.3. *"Transfer Learning: Given a source domain $\mathcal{D}_S$ with a corresponding source task $\mathcal{T}_S$ and a target domain $\mathcal{D}_T$ with a corresponding target task $\mathcal{T}_T$, the goal of TL is to learn the target predictive function $f_T(.)$ by leveraging the knowledge gained from $\mathcal{D}_S$ and $\mathcal{T}_S$ where $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$."*

3.1.2 Transfer Learning Techniques

The key fundamental of TL is utilizing knowledge learned from the source domain to improve the learning of the tar- get domain. To guarantee a good transfer performance, the following three main issues need to be taken into account.

- What to transfer: Deciding what will be transferred is the most critical step in TL. To address this issue, one needs to decide which part of learned knowledge will be transferred to improve the learning process of the target domain. This stems from the fact that not all learned knowledge from the source domain will be useful for the target domain in many scenarios. For example, some knowledge can be common in both the source and the target domains, whereas some knowledge is specific to the source domain but not the target domain.

- When to transfer: Transferring knowledge is not always helpful in speeding up the learning process of the target domain. Thus, one needs to know when the knowledge should not be transferred. For instance, if the target domain does not have anything in common with the source domain, i.e., they are not related, TL may not improve the learning process and even make the learning process less effective (i.e., negative transfer).

- How to transfer: Once the what and when questions have been addressed, one can proceed to transfer learned knowledge to the target domain. This process requires different techniques and designs to maximize the trans- ferring utilization at the target domain.

3.1.3 Transfer Learning in Deep Reinforcement Learning

In this section, we first provide fundamental knowledge for DRL and then discuss how TL can be applied to improve the performance of DRL algorithms.

1. Deep Reinforcement Learning:

   a. Markov decision process (MDP): MDP is widely used to formulate dynamic decision-making problems. Typically, an MDP is determined by four elements, including a state space $\mathcal{S}$, an action space $\mathcal{A}$, transition probabilities $\mathcal{P}$, i.e., the probability that a state s_t at time t moves to state s_{t+1} at time $t + 1$ after action a_t at state s_t. A mapping from the state space to the action a space is called a policy, denoted by π. The objective of

MDP is usually to find an optimal policy π^* that maximizes an expected discounted total reward, i.e., $\pi^* = \max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{T} \gamma^t r_t(s_t, \pi(s_t)) \right]$ where $a_t = \pi(s_t), \mathbb{E}[.]$ is the expectation function, and $\gamma \in [0, 1)$ is the discount factor representing the importance of future rewards. In particular representing the importance of future rewards. In particular, the larger the discount factor is, the more important future rewards are. In practice, due to outstanding abilities to deal with uncertainties in intelligent systems, MDP has been widely adopted to address various problems in dynamic wireless environments, such as spectrum management, cognitive radios, wireless security, and power control. Abu Alsheikh, Hoang, Niyato, Tan & Lin (2015)

b. Reinforcement learning (RL): RL is a unique class of ML. An RL agent operates in a dynamic environment formulated by an MDP framework, and its goal is to learn an optimal policy to maximize the expected discounted total reward. During the learning process, the agent interacts with the environment and observes results to gradually find its optimal policy. Specifically, at each time step t, the agent observes current state s_t, performs action a_t, according to its current policy, then receives an intermediate reward r_t and moves to new state s_{t+1}. After that, the agent adjust its policy based on the feedback of the environment, i.e., r_t and s_{t+1}.

This procedure us repeated until the agent's policy converges to the optimal one. In practice, Q-learning is one of the most widely used RL algorithms. This algorithm requires evaluating a state- action value function $Q^{\pi}(s, a)$, also called Q-function, that specifies how good of performing an action a at a state s under policy π. The value of each pair of state and actions is called Q-value.

The Q-function under the optimal policy π^* is called the optimal state-value function $Q^*(s, a)$. Suppose that values of $Q^*(s, a)$ for all state-action pairs (s,a) are known, the RL agent can obtain the optimal policy at state s by simply taking an action that maximizes $Q^*(s, a)$ Sutton & Barto (1998). The Q-learning employs a table, namely Q-table, store and update Q-values. Specifically, each cell in the Q-table stores an estimation of the Q-value for each state-action pair. Based on interactions between the agent and the environment at time t, i.e., action a_t, intermediate reward r_t, and

next state s_{t+1}, the Q-function is updated using the temporal difference (TD), which is difference between target Q-value, i.e., $Y_t = r_t(s_t, a_t) + \gamma \max_{a_{t+1}} Q_t(s_{t+1}, a_{t+1})$, and the current estimated Q-value, i.e., $Q_t(s_t, a_{s_t})$ as follows:

$$Q_t(s_t, a_t) \leftarrow Q_t(s_t, a_t) + \zeta_t[Y_t - Q_t(s_t, a_{s_t})]. \tag{3.1}$$

where ζ_t is the learning rate that represents the impact of new iteration, i.e., TD. If the Q-function is updated by (3.1) and the learning rate ζ_t satisfies condition in (3.2), it is proven that the policy learn by the Q-learning will converge to the optimal policy. Zhu, Lin & Zhou (2021)

$$\zeta_t \in [0, 1), \sum_{t=1}^{\infty} \zeta_t = \infty, \ and \ \sum_{t=1}^{\infty} (\zeta_t)^2 < \infty. \tag{3.2}$$

It is worth nothing that although the convergence of Q-learning is proven, this algorithm is inefficiency in the case of high dimensional state and action spaces since it uses a table for estimating $Q^*(s, a)$. Thus, DRL has been introduced recently as a highly-effective solution to address the current limitations of RL algorithms.

c. Deep reinforcement learning: In a high dimension environment, which has an enormous number of states and actions, the traditional RL methods, e.g., Q-learning, may not handle it effectively. For example, considering the problem of designing an RL-based agent to play various video games without knowing the rules in advance, states can be represented by images consisting of millions of pixels, making it impractical to construct a Q-table. Furthermore, it is inefficient to learn all state-action values in all states separately as that of the Q-learning algorithm. These challenges can be effectively addressed by leveraging DNN architecture for RL algorithms, i.e., DRL. The existing DRL methods can be grouped into value- and policy-based categories. In the first category, an agent first needs to learn a value function, e.g., the Q-function, then finds the optimal policy based on this function. Most of the value-based DRL

methods rely on the Q-learning algorithm, namely Deep Q-learning. Deep Q-learning employs a DNN instead of a Q-table to learn the Q-function

3.2   Problem Statement

Notation	Description
$P_s(k)$	Transmit power of the sender
$P_J^l(k)$	Jamming power of the lth jammer
R	number of transmit power levels
I	number of jamming power levels
L	number of jammers
N	number of channels
$x^{(k)}$	channel chosen by the sender
$y_l^{(k)}$	channel choosen by the lth jammer
h_s	channel power gain of the sender
h_l	channel power gain of the lth jammer

We consider a wireless communication scenario where a sender such as a wireless device transmits data to the receiver at time slot k with a transmit power $P_s(k)$. while there are L Jammers who can launch jamming attacks by injecting meaningless interference signals denoted as $P_j^l(k) \in \left\{ P_j^1(k), P_j^2(k), ..., P_j^L(k) \right\}$.

Each $P_j^l(k)$ has I different power levels.

In our model, each jammer is assumed to attack only one channel, At time slot k,the sender can choose one of N selectable frequency channels for transmitting denoted by $x^{(k)}$. Meanwhile, L jammers may select their frequency channels (denoted as $\left\{ y_1^{(k)}, y_2^{(k)}, \ldots, y_L^{(k)} \right\}$) for jamming.

h_s and h_l denote the channel power gains from the sender and the lth jammer to the receiver respectively.

To resist the jamming attack, the sender needs to choose an unblocked channel $x^{(k)}$ and an appropriate transmit power. In general, variable transmit power model is shown to be superior to the constant transmit power one under the constraint of the same average power. After the

receiver gets the signal at time slot k, the $SINR(k)$ is calculated by (1) and returned to the sender through the feedback channel.

$$SINR(k) \; = \; \frac{P_s(k)h_s}{\beta + \sum_{l=1}^{L} P_J^l(k)h_l f\left(x^{(k)} \; = \; y_l^{(k)}\right)}, \tag{3.3}$$

where β is the receiver noise power, $P_J^l(k)$ denotes the jamming power chosen by the lth jammer and $f(\xi)$ is an indicator function that equals 1 if ξ is true and 0 otherwise. If the jammer is completely blocked by the jammer at time slot k, the sender needs to retransmit the signal This will consume extra energy denoted as C_m.

It is reasonable that the channel is considered blocked if the jamming power takes the maximum value $P_J^L(k)$. In order to make a tradeoff between the energy saving and the communication performance, we define the utility $u_s^{(k)}$ of the sender by:

$$u_s^{(k)} \; = \; SINR(k) \; - \; C_m f\left(P_J^I(k) \; = \; P_j^L(k)\right) f\left(x^{(k)} \; = \; y_l^{(k)}\right) \; - \; \frac{C_s P_s(k)}{P_s^{max}} \tag{3.4}$$

where C_S and P_S^{max} denote the unit transmission cost and the maximum transmit power, respectively.

Algorithm 3.1 Algorithm example

Input: Gallery with initial templates $\mathcal{G} = \{\mathbf{r}_1, ..., \mathbf{r}_J\}$, unlabeled adaptation set
    $\mathcal{D} = \{\mathbf{d}_1, ..., \mathbf{d}_L\}$
Output: Updated Gallery $\mathcal{G}' = \{\mathbf{r}_1, ..., \mathbf{r}_{J'}\}$, $J' \geq J$

1   Estimate updating threshold $\gamma^u \geq \gamma^d$ from $\mathcal{G}$;
2   $\mathcal{G} \leftarrow \mathcal{G}'$;   `/* update gallery */`
3   **for** *all samples $d_l \in \mathcal{D}$ ($l = 1, ..., L$)* **do**
4   **for** *all references $r_j \in \mathcal{G}$ ($j = 1, ..., J$)* **do**
5         $s_j(\mathbf{d}_l) \leftarrow similarity\_measure(\mathbf{d}_l, \mathbf{r}_j)$;
6   **end for**
7   **end for**
8   $S(\mathbf{d}_l) \leftarrow \max_{j \in [1, J]} \{s_j(\mathbf{d}_l)\}$;
9   **if** $S(\mathbf{d}_l) \geq \gamma_d$ **then**
10   Output positive prediction;
11   **if** $S(\mathbf{d}_l) \geq \gamma^u$ **then**
12        $\mathcal{G}' \leftarrow \mathcal{G}' \cup \mathbf{d}_l$;
13   **end if**
14   **end if**

CHAPTER 4

PRELIMINARY RESULTS

CONCLUSION AND RECOMMENDATIONS

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

APPENDIX I

APPENDIX EXAMPLE

1.   First section of the appendix

1.1   Figures in annexes

Figure-A I-1   Figure in an appendix

In the annexes, the figures are declared in the same way. Their numbering changes automatically (e.g. Figure I-1).

1.1.1   Tables in annexes

Table-A I-1   Table in an appendix

titre	titre	titre	titre	titre	titre	titre	titre
blá	blá	blá	blá	blá	blá	blá	blá
blá	blá	blá	blá	blá	blá	blá	blá
blá	blá	blá	blá	blá	blá	blá	blá
blá	blá	blá	blá	blá	blá	blá	blá
blá	blá	blá	blá	blá	blá	blá	blá
blá	blá	blá	blá	blá	blá	blá	blá

Same behaviour for the tables (e.g., Table I-1).

(a) first figure

(b) second figure

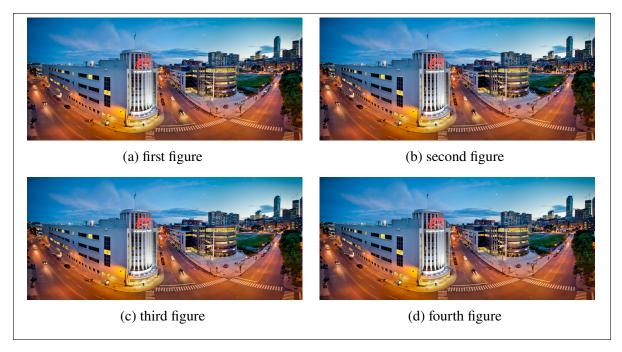(c) third figure

(d) fourth figure

Figure-A I-2   Subfig example

BIBLIOGRAPHY

Abu Alsheikh, M., Hoang, D. T., Niyato, D., Tan, H. & Lin, S. (2015). Markov Decision Processes With Applications in Wireless Sensor Networks: A Survey. *IEEE Communications Surveys Tutorials*, 17(3), 1239-1267. doi: 10.1109/COMST.2015.2420686.

Bkassiny, M., Li, Y. & Jayaweera, S. K. (2013). A Survey on Machine-Learning Techniques in Cognitive Radios. *IEEE Communications Surveys Tutorials*, 15(3), 1136-1159. doi: 10.1109/SURV.2012.100412.00017.

Dastangoo, S., Fossa, C. E., Gwon, Y. L. & Kung, H. (2016). Competing Cognitive Resilient Networks. *IEEE Transactions on Cognitive Communications and Networking*, 2(1), 95-109. doi: 10.1109/TCCN.2016.2570798.

Gwon, Y., Dastangoo, S., Fossa, C. & Kung, H. T. (2013, Oct). Competing Mobile Network Game: Embracing antijamming and jamming strategies with reinforcement learning. *2013 IEEE Conference on Communications and Network Security (CNS)*, pp. 28-36. doi: 10.1109/CNS.2013.6682689.

H., H., A., G. & D., S. (2016). Deep Reinforcement Learning with Double Q-Learning. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, (AAAI'16).

Han, G., Xiao, L. & Poor, H. V. (2017, March). Two-dimensional anti-jamming communication based on deep reinforcement learning. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2087-2091. doi: 10.1109/ICASSP.2017.7952524.

Huynh, N. V., Hoang, D. T., Nguyen, D. N. & Dutkiewicz, E. (2020). DeepFake: Deep Dueling-based Deception Strategy to Defeat Reactive Jammers.

Kasturi, G., Jain, A. & Singh, J. (2020). Machine Learning-Based RF Jamming Classification Techniques in Wireless Ad Hoc Networks.

Kwon, Y.-D., Choo, J., Kim, B., Yoon, I., Min, S. & Gwon, Y. (2020). POMO: Policy Optimization with Multiple Optima for Reinforcement Learning.

Li, W., Wang, J., Li, L., Zhang, G., Dang, Z. & Li, S. (2019). Intelligent Anti-Jamming Communication with Continuous Action Decision for Ultra-Dense Network. *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1-7. doi: 10.1109/ICC.2019.8761578.

Liu, S., Xu, Y., Chen, X., Wang, X., Wang, M., Li, W., Li, Y. & Xu, Y. (2019). Pattern-Aware Intelligent Anti-Jamming Communication: A Sequential Deep Reinforcement Learning Approach. *IEEE Access*, 7, 169204-169216. doi: 10.1109/ACCESS.2019.2954531.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S. & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533. Consulted at http://dx.doi.org/10.1038/nature14236.

Sabharwal, A., Schniter, P., Guo, D., Bliss, D. W., Rangarajan, S. & Wichman, R. (2014). In-Band Full-Duplex Wireless: Challenges and Opportunities. *IEEE Journal on Selected Areas in Communications*, 32(9), 1637-1652. doi: 10.1109/JSAC.2014.2330193.

Sun, Y., Peng, M., Zhou, Y., Huang, Y. & Mao, S. (2019). Application of Machine Learning in Wireless Networks: Key Techniques and Open Issues. *IEEE Communications Surveys Tutorials*, 21(4), 3072-3108. doi: 10.1109/COMST.2019.2924243.

Sutton, R. S. & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press. Consulted at http://www.cs.ualberta.ca/%7Esutton/book/ebook/the-book.html.

Zhu, Z., Lin, K. & Zhou, J. (2021). Transfer Learning in Deep Reinforcement Learning: A Survey.

LIST OF REFERENCES