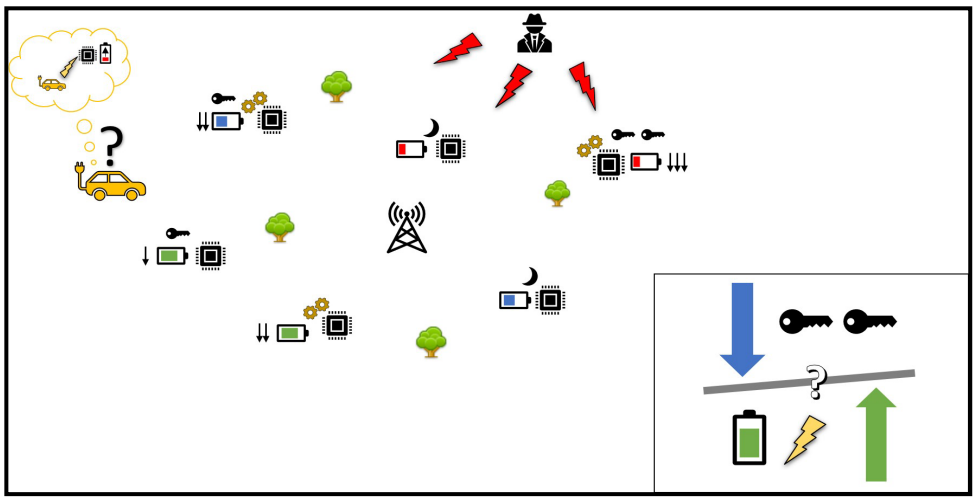
Secure and robust models for energy-efficient communications for the Internet of Things

Michaël Mahamat

Par **Michaël MAHAMAT**

Secure and robust models for energy-efficient communications for the Internet of Things

Thèse présentée
pour l'obtention du grade
de Docteur de l'UTC



Soutenue le 14 novembre 2023
Spécialité : Informatique et Sciences et Technologies de l'Information et des Systèmes : Unité de recherche Heudyasic
D2772

Thèse présentée pour l'obtention du grade de Docteur

UNIVERSITÉ DE TECHNOLOGIE DE COMPIÈGNE

Secure and Robust Models for Energy-Efficient Communications for the Internet of Things

Spécialité : Informatique et Sciences et Technologies de l'Information et des Systèmes

Par **Michaël MAHAMAT**

Soutenue le 14 novembre 2023 devant le jury composé de :

Directeurs de thèse :			
	Abdelmadjid Bouabdallah,	Professeur des universités,	Université de Technologie de Compiègne
	Ghada Jaber,	Maître de conférences,	Université de Technologie de Compiègne
Rapporteurs :			
	Romain Laborde,	Maître de conférences HDR,	Université Toulouse III - Paul Sabatier
	Pascal Lorenz,	Professeur des universités,	Université de Haute Alsace
Examinateurs :			
	Valeria Loscri	Advanced researcher HDR,	Inria Lille
	Aziz Moukrim,	Professeur des universités,	Université de Technologie de Compiègne
Membre invité :			
	Sara Berri,	Maitre de conférences,	ENSAE Cergy-Pontoise

Remerciements

Premièrement, je remercie mes directeurs de thèse Abdelmadjid Bouabdallah et Ghada Jaber pour m'avoir choisi pour mener ce projet doctoral. Les réunions ont été fructueuses, leur expertise tant que sur les domaines étudiés ainsi que la façon de faire de la recherche m'ont grandement aidé durant ces trois années. Egalement sur le plan humain, leurs conseils m'ont permis de surmonter les épreuves qu'on peut rencontrer durant une thèse. Ils ont permis la réussite de cette thèse.

Je remercie en parallèle l'initiative des Systèmes Technologiques Sûrs et Durables (MSTD) de l'Alliance Sorbonne Université, menée par M. Jérôme Favergeon et Mme. Laurie Herlin, pour le financement apporté. Le cadre apporté par les orientations de l'initiative ont été un moteur important dans la direction choisie pour les travaux effectués.

Je remercie ensuite messieurs P. Lorenz et R. Laborde pour avoir accepté d'évaluer mon manuscrit de thèse ainsi que pour les remarques apportées. Je remercie également monsieur A. Moukrim d'avoir accepté d'être le président de mon jury de soutenance ainsi que mesdames V. Loscri et S. Berri d'avoir fait partie du jury. Les discussions durant la soutenance ont été très engageantes intellectuellement.

Je n'oublie également pas les personnels administratifs du laboratoire Heudiasyc, notamment Bérengère, Nathalie et Véronique pour la gestion administrative de mon doctorat et leur gentillesse.

Il y a aussi tous les doctorants du laboratoire Heudiasyc ainsi que ceux d'autres laboratoires que j'ai côtoyés, qu'ils aient commencé avant, en même temps que moi, ou peu avant la fin. Je vous remercie chaleureusement pour les moments passés en salle de pause à jouer, à boire du café, à refaire le monde, ou à tout simplement discuter. Nous nous sommes accrochés mutuellement dans ce navire en eaux profondes qu'est le doctorat. Vous êtes tellement nombreux que j'ai peur d'oublier des noms, merci à vous tous.

Pour conclure, je remercie toute ma famille et mes amis qui m'ont épaulé durant ce parcours du combattant. J'ai commencé ce doctorat sous l'ère COVID, ce qui n'a pas été de tout repos. Malgré cette époque bizarre (devoir sortir masqué tel un certain justicier, ne pas pouvoir se retrouver en présentiel pour échanger et avancer, etc.), j'ai survécu à cette période ! Tout le monde m'a soutenu et a cru en moi pour la réussite de ce doctorat. Merci à vous !

List of publications

Thesis-related publications

Journal articles

- **M. Mahamat**, G. Jaber, and A. Bouabdallah, "**Achieving efficient energy-aware security in IoT networks: a survey of recent solutions and research challenges**", in *Wireless Networks*, Springer, 2023

International conference papers

- **M. Mahamat**, G. Jaber, and A. Bouabdallah, "**A Threat-Aware and Efficient Wireless Charging Scheme for IoT Networks**", in *International Wireless Communications & Mobile Computing Conference (IWCMC 2023)*, June 19 – 23, 2023, Marrakesh, Morocco.
- H. Souissi, **M. Mahamat**, G. Jaber, H. Lakhlef, and A. Bouabdallah, "**Analyses of Recent Advances on Machine Learning-based Trust Management for Mobile IoT Applications**", in *2022 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, September 2022, Split, Croatia
- **M. Mahamat**, G. Jaber, and A. Bouabdallah, "**A Deep Reinforcement Learning-Based Context-Aware Wireless Mobile Charging Scheme for the Internet of Things**", in *2022 IEEE Symposium on Computers and Communications (ISCC)*, June - July 2022, Rhodes, Greece.

Other publications

- **M. Mahamat**, G. Jaber, and A. Bouabdallah, "**Security and energy-efficiency in the Internet of Things: Challenges and solutions**", in *Colloque InterUT Systèmes sûrs et durables*, Université de Technologie de Compiègne [UTC], February 2023, Paris, France.

Seminar and workshop presentations

1. Michaël Mahamat, "**Energy-efficiency, robustness, and security in IoT networks: A state of the art and a first direction**", in 6th GdR RSD & ASF Winter School 2022, March 24th, 2022, Le Pleynet, France.
2. Michaël Mahamat, "**Modèles sécurisés et robustes pour une communication économe en énergie dans l'IoT à large échelle**", for the presentation of the multidisciplinary initiative "Mastery of safe and sustainable technological systems" of Sorbonne Université, May 12th, 2021, Paris, France.

Abstract

The Internet of Things (IoT) revolutionizes our lifestyle and companies, with real-time traffic analysis and environment monitoring, from anywhere on Earth. However, the adoption of IoT is still difficult due to the numerous challenges it comes with. The IoT faces two serious challenges: maximizing network lifetime and ensuring a high level of security. Although it is important to maximize the network lifetime to ensure continuous services, it is important to secure IoT networks against numerous threats. However, using security solutions increases computations which increases the energy consumption of IoT devices, thus, reducing network lifetime. These challenges are opposed, therefore, it is mandatory to find a trade-off to maximize network lifetime.

The first approach to reduce the impact of IoT security is to use solutions considering the limited energy of IoT devices. This approach is called energy-efficient security and it has attracted more and more attention in recent years. The first part of this thesis presents an extensive study of recent energy-efficient IoT security solutions. We analyze the different mechanisms they use to reduce the impact of security on the energy consumption of the devices, from energy management techniques to optimization and learning-based approaches. We after discuss the different approaches within the field along with the usefulness of Artificial Intelligence (AI) based approaches to reduce the energy consumption of IoT security.

The latter study outlined that wireless charging was not considered among energy management techniques used to alleviate the energy consumption of IoT security solutions. Wireless charging solutions use one or multiple charging devices that travel in the network to recharge the devices. Furthermore, we observed that the existing wireless charging strategies were not context-aware or able to adapt to critical changes in the network such as a network attack. Therefore, as a second contribution, we propose a context-aware charging strategy in which the context is modeled as a varying importance of the devices. The proposed approach represents this varying importance level via context prediction. Then, to determine which device to charge, we model the problem of context-aware charging with a Markov Decision Process (MDP) and propose the use of a Deep Reinforcement Learning (DRL) algorithm, namely Deep-Q learning, to solve it.

The proposed context-aware charging strategy is then modified to tackle the problem of threat-aware charging. This strategy considers the estimated threat level at a given

moment in the IoT network to determine which node will consume the most energy to defend itself against the detected threat. This strategy does not need prior information and data sets to function. This strategy, also based on DRL, learns from the observations it receives from the network to determine the device to charge next. The results obtained show that the consideration of the threat level improves network lifetime compared with non-intelligent approaches, as well as with an approach based on Deep Q-learning, which does not possess this knowledge of the threat level.

Keywords: Internet of Things (IoT), security, energy consumption, mobile chargers, wireless charging, threat-awareness.

Résumé

L'Internet des Objets (IdO) révolutionne notre mode de vie et les entreprises, notamment avec l'analyse en temps réel du trafic ou la surveillance de l'environnement depuis n'importe quel endroit sur Terre. Cependant, l'adoption de l'IdO reste difficile en raison des nombreux défis que comporte son déploiement. En effet, deux défis majeurs s'imposent à l'IdO : maximiser la durée de vie du réseau ainsi que garantir un niveau de sécurité élevé. Bien qu'il soit important de maximiser la durée de vie du réseau pour assurer un service continu, il est tout aussi important de sécuriser les réseaux IdO face aux différentes menaces. Cependant, l'utilisation de solutions de sécurité augmente les calculs effectués ce qui augmente la consommation d'énergie, réduisant ainsi la durée de vie du réseau. Ces défis étant ainsi opposés, il est nécessaire de trouver des compromis entre ces deux objectifs.

Une première approche pour réduire l'impact de la sécurité pour l'IdO est d'utiliser des solutions considérant l'énergie limitée des appareils IdO. Cette approche est appelée sécurité économe en énergie et attire de plus en plus d'attention ces dernières années. La première partie de cette thèse présente une étude approfondie des récentes solutions de sécurité IdO économes en énergie. Nous analysons les différents mécanismes qu'elles utilisent pour réduire l'impact de la sécurité sur la consommation d'énergie des appareils, en passant par des techniques de gestion de l'énergie aux approches basées sur l'optimisation ou l'Intelligence Artificielle (IA). Nous discutons ensuite des différentes approches dans le domaine ainsi que de l'utilité des approches basées sur l'IA pour réduire la consommation d'énergie de la sécurité de l'IoT.

L'étude précédente montre que la recharge sans-fil n'a pas été prise en compte parmi les techniques de gestion de l'énergie utilisées pour réduire l'impact des solutions de sécurité IdO. Les solutions de recharge sans-fil utilisent un ou plusieurs dispositifs de recharge qui se déplacent dans le réseau pour recharger les appareils. En outre, nous avons observé que les stratégies de recharge sans-fil existantes n'étaient pas sensibles au contexte ou capables de s'adapter à des changements critiques dans le réseau, tels qu'une attaque du réseau. Par conséquent, nous proposons comme seconde contribution une stratégie de recharge sensible au contexte où celui-ci est modélisé comme un niveau d'importance variable des appareils. L'approche proposée se base sur de la prédiction de contexte afin de déterminer l'importance des appareils. Ensuite, le problème de la recharge sensible au contexte est modélisé par un Processus de Décision Markovien (MDP en anglais) dont nous proposons

la résolution via l'utilisation d'un algorithme d'Apprentissage par Renforcement Profond (DRL en anglais) qu'est le Deep-Q Learning.

La stratégie de recharge sensible au contexte est ensuite modifiée pour résoudre le problème de la recharge sans-fil sensible aux menaces. Cette nouvelle stratégie considère le niveau de menace estimé à un moment donné dans le réseau IdO pour déterminer quel nœud consommera le plus d'énergie pour se défendre contre la menace détectée. Cette stratégie ne nécessite pas d'avoir à disposition des jeux de données ou des informations antérieures pour fonctionner. Cette stratégie, également basée sur le DRL, apprend des observations de l'environnement pour déterminer le nœud à charger à chaque action. Les résultats obtenus montrent que la considération du niveau de menace améliore la durée de vie du réseau comparé à des approches non-intelligentes ainsi qu'à une approche basée sur le Deep-Q learning qui n'a pas connaissance du niveau de menace.

Mots clefs : Internet des Objets (IdO), sécurité, consommation d'énergie, chargeurs mobiles, recharge sans-fil, sensibilité aux menaces.

Acronyms

AES Advanced Encryption Standard.

AI Artificial Intelligence.

BLE Bluetooth Low Energy.

DDoS Distributed Denial of Service.

DL Deep Learning.

DoS Denial of Service.

DRL Deep Reinforcement Learning.

EAP Energy Access Point.

EH Energy Harvesting.

IDS Intrusion Detection System.

IIoT Industrial Internet of Things.

IoT Internet of Things.

MCU Microcontroller Unit.

MDP Markov Decision Process.

ML Machine Learning.

NIST National Institute of Standards and Technology.

QoS Quality of Service.

RFID Radio Frequency Identification.

RL Reinforcement Learning.

SDN Software-Defined Networking.

TEA Tiny Encryption Algorithm.

WMC Wireless Mobile Charger.

WSNs Wireless Sensor Networks.

XTEA eXtended Tiny Encryption Algorithm.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and motivation

Since the advent of computer networks, our society has evolved. In the past years, a new paradigm has emerged and holds the promise of revolutionizing our society: the Internet of Things (IoT). Thanks to the IoT, citizens can get real-time data on traffic jams, available car parks, air pollution, smart waste management, etc. IoT also empowers smart industry in which companies can track in real-time the production of their factories across the globe. Thanks to IoT devices, smart sensors, and actuators, companies will know how many objects are produced, how is the production line performing, or even determine if any products are missing in the warehouse, in a real-time fashion. IoT devices can also be used to monitor and predict environmental disasters. The Internet of Things holds many promises for the future. However, in many applications, multiple challenges impede the deployment of IoT networks on a large scale. Energy and security are among the most challenging issues that slow down the deployment and acceptance by our society. IoT devices are small, cheap, and are powered via batteries or capacitors which limits the number of complex tasks they can handle. Furthermore, they are deployed in large and open environments, making them vulnerable to many attacks. Besides, with all the existing actors in an IoT environment, the protocols used are varied, thus, leading to an increased attack surface.

While the Internet of Things (IoT) is revolutionizing many domains, there are some uncertainties regarding many facets, especially regarding the energy efficiency and the security of IoT networks. IoT networks have to provide during many years services to their users while being able to adapt to environmental or context changes. However, IoT networks produce a lot of data that may be sensitive, and thus, it interests many malicious entities. Consequently, it is mandatory to protect IoT data and its users against numerous cyber-attacks. Communities and governments will accept more easily the deployment of IoT networks if the guarantees of protecting data and the networks are strong enough. Nevertheless, securing IoT devices and networks comes with many costs: an increased latency, a reduced network throughput, and an increased energy consumption. Therefore,

how it is possible to fulfill both objectives which are the maximization of network lifetime and securing IoT networks?

The objective of this thesis is to develop efficient solutions that can fulfill both objectives, i.e. securing IoT networks while maximizing their lifetime. This goal is not easy to achieve because the two constraints, energy and security, are opposed. In a given application, a naive approach to secure more a network is to use stronger security solutions, from encryption, authentication, or access control, to intrusion or anomaly detection. However, this naive approach increases the energy consumption of all IoT devices and thus, reduces network lifetime.

1.2 Contributions and outilne

The aim of this thesis is to develop efficient solutions for the joint problem of energy and security. After introducing the Internet of Things, its application domains and relevant challenges, we present background on energy management and IoT security. These preliminary chapters are the building blocks of our research work on energy provisioning using context-awareness and then, threat-awareness.

Our contributions are manifold:

1. First, we proposed, alongside the review of energy-efficient security, how could Artificial Intelligence (AI) improve the energy efficiency of security at device and network scales. We reviewed extensively existing research works that can answer the problem of energy-efficient security solutions. The energy consumption of IoT security is not negligible and directly reduces network lifetime.

2. Then, we proposed a context-aware wireless charging strategy for IoT networks. The use of context-awareness and environmental changes to design a wireless charging strategy was not explored in the existing literature. Existing works solely rely on neighboring devices, remaining energy, but they did not consider that the energy consumption of devices may vary over time due to context changes. Thus, we proposed a first model to introduce context-awareness into the planning of a charging path for heterogeneous IoT networks.

3. Finally, based on the principles of context-aware wireless charging, we proposed a threat-aware wireless charging strategy for IoT networks. In this strategy, we consider that IoT networks can estimate the threat level by itself and if an intrusion was detected. Then, given the current threat level, the energy consumption induced by the use of necessary security solutions can be derived. Finally, a wireless charger will choose the device that is the most relevant for the charging task. The research

work can effectively answer the problem of energy provisioning for IoT security.

This manuscript is organized into six chapters. First, we introduce the general context and our contributions in Chapter 1. Then, we present in Chapter 2 the background of our research work: the Internet of Things, energy-efficiency, and security methods for IoT. These are the building bricks of the following chapters. In Chapter 3, we review how IoT security solutions impact the energy consumption of IoT devices and how we can quantify it. Then, we review existing IoT security solutions that are energy-efficient, yet secure enough. We classify these works into different categories, extending existing taxonomies. Furthermore, we present the advantages of using Artificial Intelligence (AI) and Software-Defined Networking (SDN) approaches to reduce the energy consumption of IoT security solutions. In Chapter 4, we present the problem of context-aware wireless charging which opens many research directions. We propose a system model enabling context modeling and wireless charging in IoT networks. This chapter also introduces fundamental notions about Markov Decision Processes (MDP) and Reinforcement Learning (RL). We introduce variables named *importance level* and *modified importance level* that enable context-awareness. These variables are computed in a context modeling module and sent to a wireless mobile charger that will decide which device to charge according to the current context and device information. The proposed model and charging strategy use Deep Reinforcement Learning (DRL), especially Deep Q-learning as a solution to the context-aware problem. In Chapter 5, based on the work presented in Chapter 4, we study the problem of threat-aware wireless charging. With varying threat levels in an IoT network, the defense mechanisms activated for each threat are different, and thus, have different energy consumption levels. Through threat modeling and deep reinforcement learning, we design a charging strategy that considers the current threat level to determine which device should be recharged next. This approach increases network and device lifetime compared to not threat-aware approaches. Finally, in Chapter 6, we conclude this manuscript, summarize the different contributions, and we present different research directions for future works.

Chapter 2

Internet of Things: context and background

The Internet of Things (IoT) is the evolution of traditional computer networks, where everything is connected to the Internet and other devices. This new networking paradigm transforms our lifestyle and our relationship with real-time applications.

The main key feature of IoT is that the smart devices are usually deployed in an open environment, produce or collect data, and send the collected data without the need for constant human interactions. This particularity leverages new challenges and also new threats.

In this chapter, we introduce general background and notions related to the IoT. We also quickly remind the architecture of an IoT network. Then, we present two of the existing challenges that impair the deployment of IoT: energy and security. For each challenge, we present existing solutions for energy management and the security of IoT networks.

2.1 The Internet of Things: a new networking paradigm

The Internet of Things is a new technology that takes advantage of small smart objects, such as smart sensors or actuators, to provide society and companies with new applications such as smart cities, smart agriculture, or smart factories [1–4]. These smart objects can be connected to the Internet, and provide to users across the world data that will be central to decision-making processes. Internet of Things networks are considered as the evolution of Wireless Sensor Networks (WSNs) in which only sensors were connected. It is no more the computers that are connected to the Internet, but all the world around us.

2.1.1 The IoT applications

Thanks to the IoT, many applications have been created and changed our society. For instance, smart industry, smart cities, smart agriculture, smart health, and smart environment greatly benefit from the IoT.

Smart industry: In a smart factory, there are a lot of machines, i.e. production lines, actuators, sensors, etc. The IoT will be helpful to connect these machines and smart devices to optimize productivity and detect if there are any problems with the whole system [4].

Smart cities: Smart cities are the promise of improving our cities to their maximum. Thanks to smart sensors, cities will be able to get real-time data regarding traffic jams, air quality, parking availability, etc. These pieces of data will be useful to help citizens in their daily life.

Smart agriculture: In a smart farm, IoT devices will essentially consist of smart sensors. Crop, soil, temperature, and humidity will be monitored in real-time. Then, farmers will be able to have a real-time data view of their fields and take appropriate decisions to manage their lands [3].

Smart health: IoT devices will be gamebreakers for the smart health domain [2]. Practitioners will be able to get health data in real-time, provide patients with tailored health services, and even automatize treatments.

Smart environment: IoT devices can be useful to monitor the environment, detect anomalies (e.g. fires, target tracking), and allow decision-makers to make the best decision(s).

2.1.2 The IoT environment: interconnecting entities

The IoT connects multiple environments, from smart cities to smart industries. In Figure 1, we present the general infrastructure of an IoT network. In the different environments, multiple smart objects are deployed and connected to gateways, sinks, or base stations. The smart devices that can be deployed may be sensors, actuators, smart robots, smart bins, etc. They may use different technologies to communicate with gateways, edge or fog devices, and the Internet such as Sigfox, LoRA, or NB-IoT (for low power IoT networks) [5], Wi-fi, cellular technology (3G, 4G, 5G, or 6G in the future) [5, 6], etc. Devices can be interconnected between themselves, only to the gateway, or in an ad-hoc

fashion (with mobile IoT devices). The networking elements (edge, fog, base stations) are linked to the Internet which connects cloud services and data centers [7]. The cloud services and data centers are used to provide administrators, users, and developers data storage, real-time analytics, and use third-party software [7].

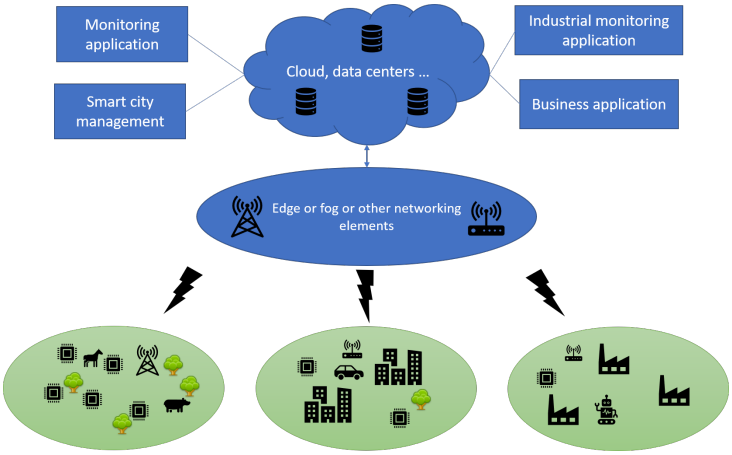


Figure 1: The IoT infrastructure connecting different environments (nature, smart city, smart industry).

2.1.3 The IoT architecture(s): The interoperability challenge

IoT has known a fast development in the past years, both in academia and industry. Thus, it led to multiple architectures, different from the classical seven-layered architecture introduced by OSI for computer networks, presented in Figure 2.

Layer 7 Application
Layer 6 Presentation
Layer 5 Session
Layer 4 Transport
Layer 3 Network
Layer 2 Link
Layer 1 Physical

Figure 2: OSI model.

One of the most considered architectures in research is the three-layered architecture [7–9] as depicted in Figure 3. In this architecture, the first layer is the sensing (or perception

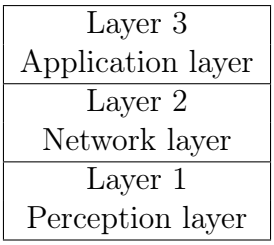


Figure 3: Three-layered IoT model [7–9].

layer) made of the smart things. Then, the second layer is the network layer and the third layer is the application layer. This architecture is basic and has limits since the network layer cannot represent all existing protocols and technologies for the networking part between IoT devices and applications [7].

Another existing architecture is a 4-layered one, where a middleware layer is between the network and the application layer [10]. The International Telecommunication Union (ITU), a specialized agency of the United Nations, also proposed a four-layer architecture in its recommendations Y.2060 and Y.4455 in order to standardize the existing architectures [11, 12]. This model is similar to the three-layer model with a fourth layer called 'Service support and application support layer' between the network and the application layer. Two transversal capabilities to the four layers are also specified: management capability and security capability. As specified in recommendation Y.2060 [11], management capabilities are used to manage device, network, or Quality of Service (QoS). Similarly, security capabilities are the different security services required at each layer of the proposed architecture.

Some research works tend to consider a five-layered IoT architecture based on the three-layer architecture [7, 8, 13]. In this five-layer architecture, the network layer is broken into a transport and a processing layer, and a business layer is added on top of the application layer as presented in Figure 4. Furthermore, there are also other five-layer architecture models, presented by Al Fuqaha et al. in their research work [7].

The diversity of architecture models for the IoT leads to a first challenge: architectural models vary and no model is being enforced by the different normative institutes. There is not a unique standard, which leads to the interoperability problem [14]. Interoperability between different entities and their applications is difficult if they have different architectural approaches and protocols. With all the existing communication technologies, networking protocols, middleware, data specifications, etc., the direct communication between devices or existing environments is difficult; adaptation is required. Researchers found that industries propose solutions to overcome the interoperability problem, but each solution proposed is not perfect. The integration of existing protocols to gateways or to adapters

Layer 5 Business layer
Layer 4 Application layer
Layer 3 Processing layer
Layer 2 Transport layer
Layer 1 Perception layer

Figure 4: Five-layered IoT model [7, 8, 13].

is not an easy task and requires skill and knowledge to overcome this interoperability problem [14].

2.2 Fundamentals on energy and network lifetime

The Internet of Things transforms our world and enables the connectivity of millions of devices. IoT devices may be deployed in hard to reach places or even some hostile environments [15, 16]. Furthermore, they are powered via batteries in most cases. Thus, changing the batteries of IoT devices in unreachable places has a high economic cost and is very difficult [17]. Thus, a challenge arises: how is it possible to maximize the lifetime of the devices, and thus, maximize network lifetime? In this section, we provide fundamentals on the different energy storages for IoT devices and definitions of device and network lifetime.

2.2.1 Energy storage for IoT devices

The majority of the IoT devices are powered by one or multiple batteries. These batteries can be rechargeable or not. However, the production of batteries is harmful to the environment (pollution of the environment or the resources needed to their production) [18]. An alternative to batteries is the capacitor which is a small energy storage. A stronger version of a capacitor is the supercapacitor which holds more capacitance than capacitors. It also holds capacitance much longer than a traditional capacitor, but less than a classical battery. Each energy storage has its advantages and disadvantages which are summarized in Table 1.

Applications that need devices to have long lifetime rely more on batteries to provide energy, with energy harvesting as a possibility for battery recharge. However, there is still the need to promote the use of rechargeable batteries and harvesting technologies.

Energy storage	Advantages	Disadvantages	References
Rechargeable Li-On battery	Good energy storage, rechargeable	Production is polluting	[19]
Capacitor	Small, cheap	Discharges quickly	[20, 21]
Supercapacitor	Lasts longer than a capacitor, very high number of charge-discharge cycles	Much more expensive than rechargeable batteries, takes more place than a battery for the same amount of energy, high discharge rate	[19, 22]

Table 1: Comparison of different categories of energy storage used for IoT devices.

Rechargeable batteries extend the operating lifetime of objects while energy harvesting refills the battery of the devices.

2.2.2 What are energy and energy consumption?

Regardless of the type of battery used, the amount of energy a battery can store is not infinite. However, what is energy? How is it characterized in IoT networks?

Definition 2.2.1. Energy is a quantitative attribute that characterizes the amount of work the system can provide. The unit of energy is the Joule (J). Another unit commonly used is the Watt second (W.s). One Watt second is equal to 3600 Joules. A common metric for battery capacity is the Ampere hour (Ah).

The amount of energy (in Joule) a battery can hold is given in the following equation, given the voltage U and the amount of Ampere hour Ah:

$$E_b = 3600.Ah.U \ [J] \tag{2.1}$$

This quantity of energy is dependent on the voltage and current required to properly power the device. The energy consumption metric is linked to the notion of power consumption P which is defined as:

$$P(t) = U(t).I(t) \ [W] \tag{2.2}$$

Then, energy consumption is the amount of power P consumed over a time period $[t_1, t_2]$ [23]:

$$E_c = \int_{t_1}^{t_2} P(t)dt = \int_{t_1}^{t_2} U(t).I(t)dt \tag{2.3}$$

Many research works are interested in determining the different causes that have an impact on energy consumption and if it is possible to reduce the consumption of the most costly blocks without degrading the QoS or network availability.

2.2.3 IoT device and network lifetime

In this thesis, device and network lifetime have a central place. We are interested in maximizing network lifetime while efficiently securing IoT networks. The problem of maximizing network lifetime is central in IoT networks because IoT devices are, most of the time, powered by a limited energy supply. Thus, their lifetime, i.e. the time that they can operate or provide a service, is limited. In the literature, multiple researchers defined what is, and how network lifetime can be characterized [16, 24]. Yetgin et al. surveyed in a recent work [16] the existing definitions and classified them into four categories:

- Node-lifetime based network lifetime,
- Coverage and connectivity-based network lifetime,
- Transmission-based network lifetime,
- Parameterized network lifetime which considers the three previous points.

Node-lifetime-based network lifetime is related to the lifetime of IoT devices [16]. Coverage and connectivity-based network lifetime is related to the ability to monitor a target or an area [16]. Transmission-based network lifetime is related to transmissions, data delivery, and other transmission characteristics [16]. Network lifetime can also be parameterized, i.e. it can consider node lifetime, connectivity-based lifetime, etc. [16].

During this thesis, the definition of network lifetime we considered falls into the category of node lifetime-based network lifetime. First of all, the definition of device lifetime we consider is an energy-based definition.

Definition 2.2.2. For a given smart device using a battery, its lifetime is the time duration between the beginning of the use of the device until the battery of the device is empty. [16]

Definition 2.2.3. For an IoT network, the network lifetime is defined as the time duration between the beginning of network use and the time when the number of dead nodes exceeds a given threshold [16].

This definition of network lifetime is representative of some applications. For instance, in an IoT-based smart city, the IoT network can still function if some smart light sensors

or smart bins are down. However, the performance of the smart city will decrease. On the contrary, in a smart industry, if some smart actuators fail, the consequences can be disastrous for the network and the application. The chosen definition of network lifetime is application-dependent, but it can either consider the network dead if one device is dead or if all devices are dead.

Since we consider node and network lifetime definitions based on the remaining energy and the number of dead devices, the definition of energy consumption given in the previous subsection is important.

There are many reasons behind the need to maximize device and network lifetime:

- Providing the users (devices, people, institutions, etc.) different services for long periods of time.
- Reducing monetary costs. If devices with non-rechargeable batteries are less changed thanks to better energy management, then maintenance costs will be less important.
- The application requires the devices and the network to run for a long period of time (e.g. nuclear powerplant).

Practitioners and network designers need to tackle the energy problem in IoT networks, otherwise, the impacts on monetary costs, user satisfaction, or even their safety will be huge.

There are two possible categories of solutions to maximize, and even extend device and network lifetime [16, 19, 25]:

1. Energy management approaches,
2. Energy replenishment approaches.

The goal of energy management approaches is to better manage the energy the devices have. They will then, have an increased lifetime. Energy replenishment approaches provide an energy income for IoT devices, thus, extending their lifetime.

In the following sections, we quickly introduce background on energy management and energy harvesting techniques.

2.3 Energy management mechanisms

Energy management mechanisms are useful to maximize network lifetime [16, 25]. Indeed, by managing well the remaining energy, devices can work for longer periods of time, and thus, have an increased lifetime. Research is very active in the area and led to many

theses and research works. For instance, Rault et al. [25] classified energy management and energy-efficient mechanisms into five categories:

1. Radio optimization techniques,
2. Data reduction techniques,
3. Sleep/wake-up techniques,
4. Energy-efficient routing and clustering techniques,
5. Battery repletion techniques, that is treated as an energy-efficient mechanism and not as a separate category.

Energy management mechanisms are more focused on the algorithmic side of devices than the hardware part [17, 25, 26].

We present below existing research works in the field of sleep/wake-up techniques, clustering techniques, and optimized deployment approaches.

2.3.1 Sleep/wake-up techniques

One of the most efficient approaches to save energy is to use sleep/wake-up mechanisms, also known as duty-cycling. Indeed, a device without activities or events to manage is in a sleep state. When an event occurs, the device wakes up (active state) to process the event. Then, it switches back to sleep mode when it no longer has any activity or event to handle.

Abedin et al. studied duty-cycling for smart devices in a smart home environment [27]. They proposed an algorithm implementing duty-cycling for IoT nodes which enables on-duty, pre-off duty (activated when the device is idle for too long), and off-duty states. To demonstrate the validity of their approach, they implemented their algorithm on a small bed test made with an Arduino board, LEDs, and two sensors. Their results show that if the device cycles between on-duty, pre-off, and off-duty, the power consumption is lower than an approach without duty-cycling (the device would be in the on-duty state indefinitely). Thus, for the user, less energy is spent and paid.

Jaber et al. [28] investigated Content-Centric Networking (CCN) for WSNs and provide an algorithm to reduce the energy consumption for content forwarding: ADDC-CCWSN (Adaptive and fully distributed duty-cycle algorithm for content-centric wireless sensor networks) based on duty-cycling. Nodes having a high activity rate for forwarding content have a high duty cycle and it is reduced if these nodes do not forward a lot of content. This duty-cycling is adaptive and can be increased or decreased according to the interest of the users. Authors show that reducing the activity of nodes does not impact the

functionalities of the protocol and improves energy consumption. The presented concepts can be applied to IoT as duty-cycling can be used to reduce the activity of IoT nodes.

2.3.2 Clustering approaches

Clustering devices into groups is one type of energy management method that is still being addressed by academia. In this approach, devices are clustered and communicate with a cluster head (CH): the chief of the cluster. This CH has more processing power and energy compared to other devices. The CH will be in charge of transferring data to other CHs, edge, fog, or cloud services. The energy burden is on the CH instead of being on all the devices.

A recent research work made by Rashid et al. [29] considers an adaptive clustering technique based on LEACH [30] for WSNs with harvesting devices. The proposed approach has two phases: a setup phase in which clusters are made and an operational phase. The selected devices can harvest energy and play the role of CHs. Clusters are made according to a CSMA-MAC based protocol. After the cluster creation, the CHs send to their cluster members a schedule based on TDMA. During the operating phase, devices send their data to their CH according to the TDMA schedule. Then, the CHs aggregate and send their data to a base station according to a CSMA schedule. Compared to LEACH, their solution consumes less energy after a certain time (between 100 and 120 seconds), has a higher throughput, and has a higher number of remaining alive devices over time.

Wang et al. studied the problem of clustering, re-clustering, and wireless charging trajectory planning for rechargeable WSNs [31]. In their model, the cluster heads are solar-powered devices that have sufficient energy supply and solar panels. However, under rainy conditions, solar-powered devices cannot harvest energy from the sun. Thus, the authors provided an algorithm to select a cluster head among devices that have been recently charged by a mobile charger.

2.3.3 Optimizing the deployment of IoT devices

An interesting approach IoT energy saving is to optimize the deployment of devices before their use. If devices are well dispatched, then, the energy required to transmit and receive data is lower than if the placement is not optimized.

Along with the re-selection problem, Wang et al. explored the problem of the deployment of solar-powered devices [31] with the goal of minimizing the deployment cost. They studied the cases of discrete and continuous environments. For the discrete deployment problem, they proposed an algorithm with an approximation ratio of $1.61(1+\varepsilon)^2$. The authors relaxed the problem formulated for the discrete problem to solve the continuous

problem.

Optimizing the deployment of devices can lead to economic savings. Indeed, Huang et al. presented a deployment scheme to minimize the energy consumption and deployment cost of an IoT network [32]. Devices are hierarchically deployed into three layers: the lowest layer has the smart devices, the middle layer has the relay devices, and the upper layer with the base stations. The main problem they studied was to determine the number of relay nodes to deploy in order to minimize deployment costs and energy consumption. They identified that their problem is analogous to a Steiner Tree problem and proposed an algorithm called Minimal Energy Consumption Algorithm (MECA) to solve it. Their simulations demonstrated that their solution reduced the energy consumption of IoT devices. However, the computational cost is high since the complexity of their algorithm is dependent on both the clustering algorithm and the Steiner tree problem (which is NP-hard).

Wang et al. presented an energy-efficient architecture for the Industrial Internet of Things (IIoT) [33]. They considered a hierarchical deployment of IIoT devices among three layers: the sensing layer with the smart devices, the gateway layer with more powerful devices, and the control layer with devices able to manage the network and communicate with the cloud. The deployment they considered has to satisfy multiple constraints: energy consumption of IIoT devices and traffic constraints. The authors also provided a scheduling mechanism to improve device lifetime. In their experiments, their approach reduced the energy consumption of IIoT devices, improved the resource utilization rate, and enabled duty-cycling for the deployed devices.

2.4 Energy harvesting schemes

A second possibility to maximize network lifetime, and even extend it, is to use energy harvesting. Energy harvesting for IoT networks requires the devices to have a harvester, a mechanism that can collect energy from an environmental source and convert it into an electrical current [34]. Energy harvesting exists for a long time. Watermills are an example of energy harvesting and conversion to another force (mechanical). Multiple energies can be harvested from the surrounding environment using dedicated harvesters. In the literature, IoT devices may harvest energy from many sources. The most studied energies for energy-harvesting WSNs and IoT networks are solar energy, wind energy, mechanical energy, radio-frequency energy, etc. [17, 34, 35].

Taxonomies exist and separate energies into multiple categories: controllable or uncontrollable, predictable or unpredictable [17, 19]. Intensive research has been conducted in this domain and many results, regarding harvester technology or energy prediction, are

available.

Energy harvesting elements are either complementary to energy storage elements, or they may replace them, especially for batteryless devices [36]. In the latter case, special care is needed when designing applications for batteryless devices that rely on energy-harvesting to properly function [19, 36, 37].

In recent years, research on RF energy harvesting has attracted a lot of attention because RF energy can be harvested from many sources, either ambient or dedicated sources [17, 21, 37, 38].

2.4.1 Predicting the harvested energy from renewable sources

The environment around us is a wonder: a lot of processes take place and produce energy. Then, it is possible to harvest and convert them into electricity (using a transducer [39]) to power IoT devices. In the literature, renewable energy sources are well-researched. Solar energy, wind energy, or even mechanical energy provide a lot of energy. While these energy sources are efficient, they are a product of nature, and thus, we cannot control them (except for some mechanical sources). Furthermore, some of them are far away (e.g. the Sun) or are chaotic (e.g. winds), or can even be disturbed by extreme weather conditions [17]. Henceforth, to improve the control of IoT devices under such uncontrollable environments, it is necessary to predict the amount of future harvested energy. Multiple models for harvested energy prediction have been proposed for WSNs, that can be adapted for IoT nodes.

One of the first proposed models to predict the amount of harvested solar energy is the Exponentially Weighted Moving-Average (EWMA) method [39]. The main assumption of the model is that the amount of harvested energy during the day d at the time slot t is similar to the energy harvested during the previous day $d-1$ at the same time slot. Although it can adapt to seasonal changes, EMWA is not suited to environments subject to chaotic weather conditions.

Another work of interest is the model called PROfile Energy prediction (Pro-Energy) was proposed by Cammarano et al. [40] for solar and wind energy. Compared to EWMA, Pro-Energy is able to consider multiple previous observations for the prediction of future harvested energy, and thus, account for sunny, rainy, or cloudy days.

In a more recent model proposed by Kosunalp et al. [41], Q-learning is applied to predict the amount of harvested solar energy. It uses EWMA [39] as the building block while Q-learning is used to compute a *daily ratio* variable that modifies the value computed by the EWMA method. Their prediction model has a lower prediction error than EWMA

or Pro-Energy for the majority of the cases. However, their model has high prediction errors in the mornings and during winter.

2.4.2 RF energy harvesting

Radio-frequency (RF) energy harvesting has attracted research due to the abundance of RF energy in the environment. RF energy comes from all the signals produced by electronic devices. Thus, it is possible to harvest energy from either produced signals, or even a dedicated RF energy source [37, 42, 43].

To be able to harvest energy from RF signals, special hardware is needed. Kamalinejad et al. presented in [42] an architecture that is able to harvest RF energy and manage it. The main contribution of their work is the introduction of a Power Management Unit (PMU) with a Wake-Up Radio (WUR). The PMU manages the available energy for the different modules while the WUR enables duty-cycling. They assessed the efficiency of this new architecture through experiments in ring and ad-hoc topologies. Compared to older approaches that did not use PMU or WUR, the combination of PMU and WUR improves by 110% device lifetime in a ring topology, and roughly by 510% in the ad-hoc topology.

Although there is plenty of RF energy in the environment, harvesting it is still difficult since there are many energy losses. Mishra et al. explored different scenarios to improve the efficiency of RH energy harvesting [21]. They outlined that the sensitivities for data reception and RF energy harvesting are different: $-60dBm$ for data reception and $-20dBm$ for energy harvesting in their scenario (the lower the sensitivity is, the better the process is). An approach to improve RF energy harvesting is to consider a Multipath Energy Routing (MPER). MPER reduces the charging time of the devices by roughly 35% if the devices are powered by a capacitor [44].

Technologies and research on RF energy harvesting are the building blocks of wireless charging [45, 46]. Wireless charging is the process of remotely charging a device, thanks to the transfer of energy between a source and the device. In Chapter 4, we give more details and related works on wireless charging strategies and trajectory planning as they are central to our contributions.

2.5 Internet of Things security

Since ancient times, securing communications has always been a major concern. In ancient Rome, military communications were encrypted with the Caesar Cipher. Nowadays, securing networks is still of major concern as data is the new gold of this era. Many threats target IoT and some attacks were successful in the past years.

2.5.1 Fundamentals requirements for IoT security

IoT networks are networks, thus, they have security requirements similar to traditional networks [9, 47]. Three requirements should always be fulfilled, which are: **C**onfidentiality, **I**ntegrity, and **A**vailability. Together, they form the **CIA** triad [48, 49].

Confidentiality: Communications and data should only be accessible only to those who are authorized.

Integrity: Data and communications should not be modified by unauthorized entities. For critical applications such as eHealth, data integrity should always be fulfilled, otherwise, the life of the patients may be endangered.

Availability: Access to data, resources, or information to the user has to be guaranteed.

Fulfilling these requirements is a good beginning for the security of IoT networks. However, there are supplementary requirements for the security of IoT networks. For instance, non-repudiation and privacy are also important requirements for critical IoT applications.

Non-repudiation: In the future, a device cannot deny that it has produced a piece of data or taken part to communications if logs exist. Furthermore, the receiver cannot deny in the future that it received this piece of data.

Privacy: It should not be possible to infer the identity of a person through their actions or the data they produced [50].

Fulfilling these requirements is done thanks to the use of adequate security solutions. In traditional computer and cellular networks, a lot of security solutions exist. For IoT networks, it is harder to fulfill all of these requirements because IoT devices are constrained on many planes: energy, computation, data storage, etc. [18]

2.5.2 The IoT, a highly threatened world

While the IoT holds the promise of connecting everything in the world, it is not secure for the moment. There are multiple constraints that lead to unsecure IoT devices such as limited computation power or available energy. Furthermore, a lot of threats target the devices, leading to an insecure world. There are two categories of threats against IoT networks:

Passive attacks: these attacks do not aim to actively impede the network but are more focused on spying devices and eavesdropping communications.

Active attacks: these attacks impact the functionalities of the network. An attacker may modify data, forge or drop packets, etc. Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks are active attacks that are dreadful for classical and IoT networks. In these attacks, a malicious entity wants to impede users or devices from accessing some services. These attacks target the availability of IoT devices and networks [48].

IoT can also be a vector for attacks against other networks and entities. The botnet called Mirai [49] took advantage of cameras over IP that were not secured: they were using default or easy-to-guess passwords.

Furthermore, with all the technologies empowering the Internet of Things [7, 51], a lot of vulnerabilities exist. For instance, the Bluetooth Low Energy (BLE) communication protocol suffers from many security breaches [52].

As security in IoT networks, and previously in Wireless Sensor Networks (WSNs) is a hot research topic, there are numerous research works aiming to unveil new attacks and provide adequate countermeasures.

2.5.3 Countering attacks against IoT networks

Even though the landscape of IoT attacks is gloomy, there are security solutions able to mitigate some of them. For each security requirement, there are security solutions that can fulfill them.

Protecting confidentiality

First of all, attacks against confidentiality, such as eavesdropping, can be mitigated using encryption or access control methods [53]. Encryption and access control methods exist for a long time. The encryption process transforms a base message into another message, that is unintelligible. To do so, special strings called *keys* are needed. With operations using the base message and the key, it is possible to encrypt a message. There are two classes of encryption algorithms: symmetric encryption algorithms and asymmetric encryption algorithms.

Advanced Encryption Standard (AES) is a symmetric encryption algorithm that is well known and used in the domain of cryptography [54]. In wireless or IoT networks that use the IEEE 802.15.4 technical standard, the security at the link layer is based on AES [55–57].

Protecting integrity

Data integrity is vital in critical applications such as Industrial Internet of Things (IIoT) applications which use IoT devices as the backbone of factories or powerplants. Then, attacks against integrity can be mitigated using data authentication or false data detection for instance. Data authentication is a well-known and well-research approach in networks.

A recent technology that guarantees data integrity is the blockchain [58]. A blockchain is a chain of blocks in which each block stores the different transactions done by different participating entities. The chain is done by computing a hash of a block and this hash is stored into the next block. Since the blockchain is stored within all the participating devices, it is near impossible to tamper the stored transactions or data of the blockchain [47, 58]. Thus, the blockchain is an excellent candidate for the protection of data integrity. However, since it is a distributed database, it requires the devices to be able to store the blockchain. Furthermore, blockchain is not suited for energy-constrained devices because the operations required to compute the hash are energy-consuming [47, 59].

Protecting availability

Defending availability in IoT networks is harder but can be achieved thanks to redundancy and recovery schemes. To maximize the availability of an IoT network, few or no attacks have to happen. Thus, intrusion, anomaly, or threat detection are important tools to detect any attacks that target the availability of the network [60, 61].

2.6 New approaches and technologies against threats in IoT networks

While encryption-based solutions are efficient against confidentiality and integrity-based attacks, they are still inefficient against availability attacks. Furthermore, they heavily rely on cryptographic keys, which incur supplementary operations for management, revocation, computations, etc. If cryptographic protocols are broken, there is still need to provide other security barriers in IoT networks.

2.6.1 Trust-based approaches

Trust-based security solutions are an interesting alternative to cryptography for securing IoT network communications. Trust-based security solutions are based on the same social concept: trust. A device A trusts a device B if device A thinks that B behaves the way it should [62]. In a trust-based approach, *confidence* between entities (devices,

gateways, etc.) is of uttermost importance. This confidence is evaluated through different methods. If an entity has a high confidence towards another entity, it means that this entity has the right behavior and is not harming communications [62–64].

Hellaoui et al. proposed a simple, yet energy-efficient trust management approach for IoT networks [63]. If a device is trusted enough, then authentication is not applied to its messages. On the contrary, if the device is not trusted, according to an adaptive function f, authentication is applied to the messages received. If the authentication fails, then the messages are dropped. Thus, trust management can be useful to reduce the energy consumption and is a good candidate for adaptive security, which is discussed in Chapter 3.

Boudagdigue et al. designed a trust management system for the Industrial Internet of Things (IIoT), inspired by the social Internet of Things (sIoT) [64]. Their trust management system requires a new architecture: the IIoT network is made of clusters called *industrial communities*, which is heavily inspired by social IoT. The industrial communities are formed according to industrial relationships and distance between devices. IoT devices are called *community members* and belong to an industrial community which is lead by a community leader. Then, these community leaders evaluate the trust of each community member according to three metrics: cooperation, direct honesty and indirect honesty. Finally, the community leaders communicate with a central IIoT server. In their experiments, Boudagdigue et al. demonstrated the effectiveness of their architecture for trust management as their solution has a lower packet loss rate, shows a lower energy consumption, and is resilient to coalition and bad-mouthing attacks compared to a centralized trust management scheme. Their work further demonstrates the usefulness of trust management for energy-efficient security.

Trust-based security is a good approach toward autonomous security management of IoT networks. Devices will collect trust data, determine if there are any malicious devices, and stop communicating with them if their trust value is too low. Thus, trust management approaches for IoT security can either replace or complement well cryptography-based approaches [50, 62–64].

2.6.2 Artificial intelligence-based approaches

Artificial Intelligence (AI) techniques have been widely applied in many applications for IoT networks, especially in the field of IoT security. Recent AI techniques considered for IoT security belong to the fields of Machine Learning (ML), Deep Learning (DL), and Reinforcement Learning (RL).

1. Machine Learning (ML) is a learning paradigm also called *statistical learning*. It is a

field of artificial intelligence that aims to build models from statistical data.

2. Deep Learning (DL) is a sub-field of ML that is built upon neural networks. DL models are efficient for large scale data analysis.

3. Reinforcement Learning (RL) is a sub-field of ML in which an agent aims to learn the best actions to take in an environment.

ML and DL algorithms are used for regression and classification tasks. ML and DL algorithms can be classified into three categories, namely supervised, semi-supervised, and unsupervised algorithms. Reinforcement Learning (RL) is a special sub-field of ML that we will present more deeply in Chapter 4. RL is efficient when decision-making is required. Hence, all these learning paradigms are useful for IoT security.

ML and DL algorithms applied to IoT security achieve good results for anomaly and intrusion detection. Indeed, since ML and DL algorithms can uncover hidden patterns in data, they can detect anomalous behaviors. RL applied to IoT security is useful to fine-tune the control of a security policy [65].

Roopak et al. studied different deep learning models to detect DDoS attacks in IoT networks [60]. They compared four models: the classical Multilayer Perceptron (MLP), a Convolutional Neural Network (CNN), a Long Short Term Memory (LSTM) network, and a hybrid network with a CNN and a LSTM. In their experiments, the hybrid network is the most accurate and has the highest recall value among deep learning models, with an accuracy of 97.16% and a recall of 99.1%. However, compared to classical machine learning models, the CNN combined to an LSTM is less precise than a classical Support Vector Machine (SVM) (97.41% vs 97.72% for the SVM).

Furthermore, recent advances such as Federated Learning (FL) [66] improve the privacy of the users and companies. In a Federated Learning setup, multiple devices communicate to train models. The particularity of FL is that all the devices do not need to share their dataset(s), which means that their data remains where they are produced. Thus, it reduces the risk that an eavesdropper intercepts data being sent to servers or cloud. Huang et al. applied federated learning to intrusion detection in IoT networks when data is not independently and identically distributed [67]. Their solution creates clusters of trusted devices that are connected to local servers. One cluster is considered as a client that will train a model on the local data. After training the local model, the local gradient is sent to an aggregation server that aggregates all the gradients of the local servers. Then, the aggregation server sends the result of the new gradient to all the local servers. They validated their approach using the IoT-23 dataset (representing a network of 23 IoT devices). For bigger cluster sizes, the accuracy of their solution increases, but is lower than a centralized approach. However, the advantage of using federated learning lies

in the privacy of data, i.e. it stays locally and does not go to the cloud or other entities [67].

2.6.3 Software-defined networking

Software-defined networks transform the way networks are designed. While in traditional networks, routing and analysis logic are embedded into networking elements, in an SDN approach, network control and data logic are separated [68, 69]. It means that the routing logic is not done by the network element themselves (routing table, discovery, etc.), but it is rather done by a central entity called SDN controller. This SDN controller creates the routing tables and flows, and then, dispatches them to the different devices. SDN is paired with Network Function Virtualization (NFV) which enables the implementation of software network functions called Virtual Network Functions (VNFs) [69]. These VNFs are softwarized network functionalities, that were hardcoded in the traditional networking hardware. SDN paired with NFV eases the deployment of security functionalities such as abnormal traffic detection, intrusion detection, etc. [69–71]. It even allows a better resource management, implementation of QoS features, etc [69]. This approach reduces the complexity of the deployment of such functionalities, which were deployed in specific devices (gateways, routers, etc.). Thus, the maintenance costs decreases, while increasing the efficiency of the network.

However, since the SDN approach needs a central component, it is the Single Point Of Failure (SPOF) of the system. Thus, if an attacker is able to shutdown the SDN controller, due to DoS or DDoS attacks, the network will be greatly impacted [72]. A solution to mitigate the SPOF of an SDN-based IoT network is to use multiple backup SDN controllers for redundancy [69, 72].

SDN and NFVs are great enablers of IoT security, but they require that the devices are compatible with SDN and NFVs functionalities. Otherwise, traditional networking hardware is required to manage these devices. However, these pieces of hardware may be compatible with SDN and NFVs, or connected to a virtual switch compatible with SDN and NFVs functionalities [68, 72].

2.7 Conclusion

In this chapter, we presented the general context and background of this thesis. Firstly, we introduced the IoT, its applications, and its main features. Then, we outlined the challenges that are energy and security. Both challenges are the main limitations for the adoption of IoT by our society. IoT devices are energy-constrained. Thus, the available energy should not be spent into useless or very consuming tasks. Then, we presented

the basis of IoT security, what are the existing threats, and the existing mechanisms to mitigate them.

In a perfect world, for each existing threat, a defense mechanism should be used. However, each security solution used impacts the IoT devices on multiple criteria: energy consumption, latency, throughput, etc. Even recent advances and concepts applied to IoT security consume energy. Thus, a question arises: 'Is it possible to protect IoT networks against multiple threats while not consuming too much energy?' While it is true that heavyweight security solutions can be used in edge or IoT gateways, they nevertheless induce supplementary latency and energy consumption if the devices are energy constrained. Thus, it is of uttermost importance to determine if it is possible to limit the impact of IoT security on the energy consumption, and thus, network lifetime. In the next chapter, we will tackle the problem of energy-efficient security for IoT networks through an extensive survey of existing solutions.

Chapter 3

Energy-saving security solutions for IoT networks

Ensuring the security of IoT networks is crucial, especially considering their use in critical-mission applications such as e-health or smart power plants. A successful attack on a critical network could have disastrous consequences, ranging from the failure of an industrial process to the death of patients, for instance. Nevertheless, it is important to acknowledge that security solutions tend to be energy-consuming and can significantly reduce the overall lifespan of IoT devices. It is essential to ensure that attempts to reduce the energy consumption of security solutions do not compromise the level of security services they provide.

In this chapter, we present existing security solutions for IoT networks that consider the limited energy of the devices. As previously highlighted in Chapter 2, for applications in which there is no access to the grid (or it is costly), the deployed devices are primarily powered via batteries or capacitors. Given that security solutions induce supplementary computations and communications, they greatly impact the energy consumption of the devices, resulting in a reduced lifetime. Furthermore, the deployed devices consume energy for each running process, application, data transmission or reception, and security solution used. As the deployment of security solutions such as encryption, anomaly detection, or trust approaches increases, the energy consumption of the devices increases which further reduces their lifetime. This observation leverages a crucial question: Is it possible to secure IoT networks against various threats without significantly increasing the energy consumption of the devices? Additionally, we want to determine whether there exist approaches or solutions to mitigate the energy impact of security measures on IoT devices.

To this end, we begin this chapter by discussing the impact of security solutions on the energy consumption of an IoT device. Subsequently, we present a critical review of existing approaches used to reduce the energy consumption of IoT security solutions while efficiently securing IoT networks. Furthermore, we discuss the contributions of Artificial Intelligence (AI) and Software-Defined Networking (SDN) in developing energy-efficient security solutions for IoT networks. These new approaches may prove useful to attain

an optimal balance between security and energy consumption. Finally, we conclude this chapter by motivating our selected research direction for the next chapters.

3.1 Related works

Although research in the field of energy-efficient IoT security solutions is gaining momentum, it receives less attention compared to research focusing on enhancing the overall strength of IoT security (e.g. the use of blockchain technology to provide data integrity). Previous studies have addressed the issue of energy consumption in security for Wireless Sensor Networks (WSNs) and, subsequently, for IoT. However, research works dedicated to reduce the energy consumption of IoT security are rarer compared to research works dedicated to increasing the strength of the provided security services.

A valuable work done by Mauro et al. [73] tackled the impact of energy harvesting on the security of WSNs. One of their major contributions is the design of a method that adaptively secures communications in WSNs. Their approach considers that, for each communication link in the network, a device with lower and maximum security requirements can choose the most suited encryption or authentication algorithm to fulfill these requirements. Each data packet has a security requirement and is transmitted if the encryption or authentication method used also fulfills this requirement. This approach establishes a safe route for data packets. Therefore, their strategy is adaptive toward both available energy and the security requirements of the devices. However, it is not adaptive to the threats they may face.

Alharby et al. proposed a solution to integrate adaptive security in resource-constrained IoT devices[56]. Their contributions are manifold and focus on the IEEE protocol 802.15.4. They first studied the trade-offs between the security levels of IEEE 802.15.4 and latency, energy consumption, and throughput. Then, they designed an adaptive security mechanism named PASER for resource-constrained IoT devices and provided use cases. PASER is used to provide trade-offs between device lifetime and security. Instead of using the eight available security levels of IEEE 802.15.4, they considered four levels: level 0 (no security), level 1 (authentication), level 4 (encryption only), and level 7 (authentication and encryption). PASER also considers the importance of packets and gives more priority to packets with important data when the remaining energy runs low. Finally, they conducted many experiments to demonstrate that adaptive security for resource-constrained IoT devices extends their lifetime while securing them efficiently.

Few surveys tackled the problem of energy-efficiency for IoT security. Hellaoui et al. [74] surveyed energy-efficient approaches for IoT security. They outlined that the impact of security on the energy consumption of IoT devices is not negligible. Thus, it is important

Reference	Year	Scope	Comments
Di Mauro et al. [73]	2015	EH and security in WSNs.	Thesis studying the impact of EH on the security of WSNs.
Hellaoui et al. [74]	2017	Energy-efficient mechanisms for IoT security	Survey presenting energy-efficient mechanisms for security solutions to alleviate computations and decrease the energy consumption of security solutions.
Alharby et al. [56]	2020	Adaptive security for IoT	Thesis studying trade-offs of adaptive security and energy consumption of energy-constrained IoT devices.
Tedeschi et al. [37]	2020	Security mechanisms for energy-harvesting enabled IoT	Survey presenting security solutions for energy harvesting networks, with a major focus on PHY-layer.
Yousefpoor et al. [75]	2021	Security of data aggregation	Survey presenting data aggregation and methods to secure this process. Secure data aggregation is less energy-consuming than securing a network without data aggregation.

Table 2: Table summarizing the scope and remarks of related works.

to design security solutions that consider the energy constraints of IoT devices. They focused their study on energy mechanisms for security primitives, key establishment, and access control. They proposed a new taxonomy classifying energy-efficient mechanisms into six categories: online versus offline security, outsourcing, adaptive security, low-power security protocols, data compression, and hybridization. However, it mainly focuses on authentication methods, signature methods, and key management systems, as they may consume a lot of energy.

A recent review done by Tedeschi et al. [37] studies the problem of security in Energy Harvesting (EH) WSNs in which the devices may not have long-lasting batteries. Furthermore, they focused their study on devices powered thanks to RF energy. These devices are, however, targeted by specific threats such as beamforming vector poisoning attacks, leeching, greedy, or cheating attacks. Against all these threats, Tedeschi et al. determined that three categories of security solutions can efficiently protect these WSNs where devices harvest RF energy: cryptography-based methods, data-secrecy methods,

and PHY-layer countermeasures. First, cryptography methods for EH networks use energy-efficient mechanisms to reduce the impact of cryptography on energy consumption. Thus, pre-computation techniques, computation offloading, or implementation optimization are used to reduce the energy consumption of these solutions. Data secrecy methods for the PHY-layer are an alternative to cryptography methods if the devices cannot afford the energy consumption linked to encryption-based methods. Finally, PHY-layer countermeasures are used to defend devices against threats such as jamming or DoS attacks. Their review tackled well the security of WSNs in which devices are powered via EH.

A study on secure data aggregation methods was presented by Yousefpoor et al. [75]. Data aggregation is a useful approach to reduce the energy consumption of IoT devices. Devices called data aggregators receive from other devices their data, perform a data aggregation algorithm (statistical, time series, etc.), and send the result to a sink or gateway. While data aggregation is useful for reducing energy consumption in the network, the operation is prone to attacks such as false data injection. Thus, it is necessary to secure this energy-efficient approach for WSNs and IoT networks. Their survey shows that securing the data aggregation process is less consuming than securing a network that does not implement data aggregation.

All of these related works tackle a particular problem of energy-efficiency of security in IoT networks. Di Mauro et al. provided protocols to secure communications in EH-WSNs, although their works did not consider threat-awareness [73]. Hellaoui et al. focused their study on energy-efficient mechanisms for security primitives (encryption, authentication, signature methods) [74]. On the opposite, Tedeschi et al. heavily studied the security of EH WSNs with an extensive survey of PHY-layer-based techniques; cryptography-based methods are not the main focus of their study, although they provided the most recent advances in the field for EH devices [37]. Alharby et al. did extensive experiments to study the trade-offs between the different security levels of the IEEE 802.15.4 norm and considered threat-awareness for their contribution: PASER [56]. However, their approach is binary: if there is a threat, then the maximum security level is used; otherwise, the chosen security level is mapped to the sensor values. Yousefpoor et al. only studied secure data aggregation in IoT networks [75] but provided a large overview of the domain. Data aggregation reduces the energy consumption of an IoT network, but the process needs to be secured.

The sparsity of works on the subject of energy-efficient security mechanisms and solutions for IoT networks motivated us to do a new study in this research field. Moreover, emerging approaches such as artificial intelligence or SDN may improve the energy efficiency of IoT security solutions while guaranteeing an appropriate security level. In the next section, we present the impact of security solutions on the energy consumption of IoT

devices.

3.2 Impacts of security on energy consumption

As presented in the previous chapter, the use of security solutions in IoT networks is mandatory to protect data, users, and devices. However, using a security solution may incur supplementary computations and, therefore, devices consume more energy. Thus, the first step is the following research question: How can we quantify the impact of a security solution on the energy consumption of an IoT device?

One of the most studied fields in IoT security is encryption which fulfills confidentiality and keeps eavesdroppers at bay. These methods can be combined with authentication and access control mechanisms to further improve user and data protection. However, the main drawbacks of these approaches are their computational and communication overheads, thus, leading to increased energy consumption for the devices. Consequently, determining their energy consumption is a first step to knowing if it is possible to reduce the strength of the underlying algorithms (to reduce the overheads) or if it is necessary to implement energy management methods to improve device lifetime. There are also other categories of security solutions, such as intrusion detection methods, false data detection methods, etc. Their energy consumption relies on the underlying algorithms, rules, etc.

Overall, the energy consumption of an IoT security solution can either be measured, estimated, or computed. It is possible to differentiate the energy consumption of the computation phase and the communication phase [20, 76]. Within the computation phase, it is possible to determine if it is the encryption phase, the decryption phase, or the set key phase [53]. Different models have been proposed by researchers to determine how can a security solution be evaluated in terms of energy consumption. In this section, we first present the approach of measuring the energy consumption of security solutions. Then, we will present energy models that consider the impact of security solutions on the energy consumption of IoT devices.

3.2.1 Measuring the energy consumption of a security solution

A first approach to quantify the energy consumption of IoT security solutions is to determine their energy consumption during runtime. Thanks to shunt resistors (with a low impedance) and an oscilloscope, it is possible to measure the voltage, the current, or the power of the system when the security solution is running [53, 77–79]. Then, the energy consumption of the security solution can be measured or computed from the power consumption, according to Equation 2.3 (in Chapter 2).

Over the years, many research works studied the energy consumption of encryption-based security solutions. Indeed, encryption-based solutions are the basis of guaranteeing the confidentiality of communications. The study of the energy consumption of encryption-based solutions is not new, as it was already a research problem in WSNs.

Alharby et al. quantified the impact of the IEEE 802.15.4 protocol and its different security levels [80] that are based on Advanced Encryption Standard (AES). For higher security levels, i.e. that use longer keys and more services (encryption and authentication for levels 5, 6, and 7), the impacts on latency, energy consumption, and throughput are important. For a payload of 24 bytes, when using the lowest security level (level 1, only authentication), the overall energy consumption is increased by 31.54 % compared to an unsecured packet. If only encryption is used (level 4), the energy consumption for a payload of 24 bytes is increased by 33 %. On the contrary, if the highest security level is used (level 7, encryption and authentication, with a MIC of 16 bytes), the energy consumption is increased by a factor of 60.46%. When considering a payload of 80 bytes, the relative increase is less important. However, the energy consumed in Joules is bigger than a payload of 24 bytes. Another contribution of this research work is that the energy consumption of the Microcontroller Unit (MCU) is smaller than the consumption of the radio, but it cannot be neglected. Indeed, for security level 7, the energy consumption of the Microcontroller Unit (MCU) accounts for 22%.

The previous observation for the energy consumption of the radio versus the MCU is confirmed by Maitra et al. [78]. In their research work, they studied the impact of AES and eXtended Tiny Encryption Algorithm (XTEA) for encryption when deploying IoT-based applications such as monitoring or fall detection of elderly people. The encryption mechanism consumes roughly 10 % of the energy consumption of the applications, while the radio part consumes the majority of the energy.

From the two previous articles, one point is interesting: the energy consumption of an encryption-based solution is platform-dependent. It means that the energy consumption of a particular encryption algorithm relies on the hardware platform considered (the MCU). Furthermore, while the energy consumption of the radio module is the most important, the energy consumption of IoT security is not negligible, even if its part is lower than the consumption of the radio module.

Schaumont et al. outlined in [81] that the use of signature algorithms for capacitor-powered IoT devices is near impossible. Indeed, if the device is powered with a piezo-electric harvester and relies on the ECDSA signature algorithm, it can only perform three authentications per hour. On the contrary, a device powered by an AAA battery can ensure 250.10^3 authentications with the same algorithm thanks to the higher available

energy. They advocate the need to introduce energy-awareness for the design of security solutions, especially if the device can harvest energy. Heavy computations can be carried during periods when harvested energy is high, thus allowing the device to authenticate more times per hour.

Schaumont et al. further studied the impact of 18 authentication protocols on the energy consumption of a solar-powered MSP430 microcontroller [20]. During this research, they identified that the factors impacting the energy consumption of an authentication protocol were the algorithm type (MAC-based or signature-based), the security level (in bits), the number of passes during authentication (one or two), the voltage multiplier, and eventually the use of a hardware multiplier. They considered SHA1, SHA2, and Keccak for the MAC-based algorithms, while ECDSA, Winternitz, and Lamport signatures were considered. Their findings indicated that signature-based authentication protocols consumed more energy compared to MAC-based solutions. Furthermore, the use of a hardware multiplier demonstrated a reduction in energy consumption for eligible algorithms (in this study, ECDSA). These results contribute to the design of energy-efficient authentication protocols for IoT devices.

Vračar et al. focused their study on three encryption algorithms and their energy consumption on a PIC18F45K22- microcontroller [77]. They considered Tiny Encryption Algorithm (TEA) [82], eXtended TEA (XTEA) [83] and SKIPJACK [84]. Furthermore, they studied two asymmetric signature algorithms, RSA and ElGamal, however, with low-size keys (16-bit). Their experiments outlined the increased energy consumption of TEA, XTEA, and SKIPJACK during the communication phase, whereas RSA and ElGamal consumed more during computation and signature phases. They also observed that XTEA is the least energy-consuming encryption algorithm, while SKIPJACK is the most energy-consuming but the fastest one.

Kane et al. confirmed that AES consumes more energy than Chacha and Acorn, regardless of the experimental platform [53], whether it is the ATmega328, the STL32F103C8T6, or the ESP8266. However, encryption and decryption times are platform-dependent, e.g. Atmega328 is the slowest platform while the ESP8266 is the fastest one. According to the researchers, the STM32F103C8T6 microcontroller is a good choice for developing IoT applications that need data confidentiality since it has a good balance between energy consumption and cipher performance.

Through extensive experiments on nRF51822 and Atmega328 platforms, Aerabi et al. confirmed that AES and its variants are not among the least energy-consuming block ciphers [79]. Their experiments confirmed that RC6, TEA, and Simeck have the best performance, the lowest energy consumption, and few cycles spent to process a

bit. Furthermore, their results show that many stream ciphers consume less than block ciphers, with the most energy-consuming stream cipher at $30nJ/bit$ while the most energy-consuming block cipher consumes nearly $98nJ/bit$. They also did a case study on a batteryless implantable medical device that gains energy from solar cells. In this case, the throughput is directly impacted by the harvested energy and the key size of the cipher.

In another category of encryption solutions, Attribute-Based Encryption (ABE), Girgenti et al. [85] studied the energy consumption with the encryption and decryption times of three Attribute-Based Encryption (ABE) schemes: Goyal-Pandey-Sahai-Waters's scheme (KP-ABE), Bethencourt-Sahai-Waters scheme (CP-ABE), and Yao-Chen-Tian scheme (KP-ABE). Through extensive simulations, they observed that the number of attributes has a direct impact on energy consumption, encryption, and decryption times. KP-ABE schemes are more energy-efficient than the CP-ABE scheme, but CP-ABE schemes are easier to implement.

There are many research works detailing the energy consumption of encryption, authentication, or signature algorithms, as they are the backbone of IoT security. Although these solutions are necessary to guarantee confidentiality and integrity, various measurements and experiments have shown that these security solutions consume a lot of energy. Consequently, a first approach to reducing the energy consumption of an IoT network is to choose the right encryption, authentication, and signature algorithms or mechanisms. This choice should consider the following:

- The energy constraints of the devices [53, 79]
- The computation constraints of the devices [53, 79]
- The criticality of the application (a more sensitive application will require stronger encryption, authentication, or signature methods) [57, 73].

If less energy is spent on encryption, authentication, or signature tasks, then device and network lifetime will increase.

3.2.2 Modeling the energy consumption of a security solution

In the previous subsection, we outlined that encryption, authentication, and signature methods consume a lot of energy. Many experiments validated the non-negligible impact of these solutions on energy consumption and therefore, on the lifetime of devices.

Besides measurements, a second approach to quantify the energy consumption of a security solution is to model this security solution and its different composing blocks (methods used, phases of the solution, etc.). The use of models to evaluate the energy

consumption of security solutions may pinpoint the most costly blocks of a particular solution, not only regarding computation costs but also communication costs.

A recent model proposed by Conceição et al. evaluates the different phases of an IoT security solution to determine its energy consumption [76]. This energy model evaluates all the phases of a security solution, from the establishment of secure communication to its end. The contribution of the authors is the consideration of the networking cost induced by a security service, as opposed to previous works which only considered the energy consumption of the computation phase. At a given time t, the energy consumed by a node d, given it has n connections, is given by:

$$E_d = \sum_{i=1}^{n} E_c(i) + E_{OS} \tag{3.1}$$

where $E_c(i)$ is the cost of the i^{th} connection of the node d and E_{OS} is the energy consumption of standard tasks. The authors established that a connection between two nodes could be broken down into three phases (E_c): the creation of the security context, the data exchange phase, and the key update or revocation (if the connection has ended) phase. All these phases consume energy and rely on the security services implemented in the devices. Furthermore, a symmetric cryptography service consumes less energy than an asymmetric cryptography service (as presented in the previous subsection).

The model presented by Conceição et al. [76] tackles more operational phases of a security solution; however, it can only be used on security solutions based on encryption or authentication services. Furthermore, given the variety of IoT security solutions and the huge number of existing threats, determining a general model for the energy consumption of a security solution is a tedious task. More research is needed to determine a model covering more categories of security solutions since IoT security is not only encryption, authentication, or access control mechanisms but also covers intrusion detection, anomaly detection, PHY-layer security, etc.

3.2.3 Discussion

The number of available studies on the energy consumption of encryption, authentication, or signature methods is still growing and shows that these solutions drastically impact the energy consumption of IoT devices. The common point among these papers is that the energy consumption of AES is not negligible [53, 78, 79]. Thakor et al. presented many lightweight encryption algorithms (with a software or hardware implementation) which may provide a sufficient security level along with a lower energy consumption [86]. Encryption, authentication, and signature algorithms are the primary brick to ensure

Parameter	Impact on energy	References
Hardware platform	***	[53, 78, 79, 86]
Implementation type (hardware vs software if applicable)	* to ***	[79, 86]
Encryption category (symmetric vs asymmetric)	***	[79]
Key size	Algorithm-dependent	[20, 78, 79]
Data size/digest size	Algorithm-dependent	[20, 79, 80]

Legend: *** strong impact, ** medium impact, * low impact.

Table 3: Impacting parameters of encryption-based methods on energy consumption.

confidentiality and integrity. If manufacturers and developers carefully choose appropriate encryption-based methods, their products may have a longer lifetime than those using only AES or ECDSA, for instance.

Multiple studies showed that the energy consumption of security primitives is device-dependent. Moreover, the different implementations (hardware, software, or both) for a given cipher (if they exist) have different energy consumption. Thus, the main parameters that impact the energy consumption of encryption-based methods are:

- the hardware platform;
- the considered implementation (hardware or software);
- the category of the encryption method, whether it is symmetric or asymmetric;
- the key size;
- the size of the digest or data to encrypt.

For encryption-based methods, the previous parameters have varying impacts on the energy consumption of the devices. According to the different results, we give in Table 3 the impacts of these parameters on energy consumption. For more details, we refer the reader to corresponding works [20, 77, 79, 80, 85, 86]. The energy consumption of complex security solutions, merging different encryption-based methods plus learning techniques, is far more difficult to evaluate. In the next section, we present IoT security solutions aiming to balance the energy consumption of the security solutions in the devices while ensuring a high security level.

3.3 Categories of energy-efficient security solutions for IoT networks

In the previous section, we highlighted the significant impact of security solutions based on encryption, authentication, or signature methods on the energy consumption of IoT devices which cannot be overlooked. Existing security solutions for IoT networks, and even computer networks (even if these networks are out of the scope of this thesis), are more designed with performance in mind and less energy efficiency and sustainability. In energy-constrained networks and applications, using a security solution that provides a high security level (such as strong encryption, local anomaly, or threat detection for instance) reduces network lifetime since it requires many computations.

Furthermore, many applications consider a fixed security level, usually the highest one (longest keys, strongest algorithm supported by the MCU, etc.). This approach is inefficient as it draws useless resources from the devices, which could be either used for other tasks or to increase device lifetime [55]. For instance, if the system considers the IEEE 802.15.4 specification, which is built on AES, the common approach would be to use the maximum security level, the 7th level (which guarantees authentication, confidentiality, and integrity). This approach is the most energy-consuming [55, 80]. Hence, these solutions cannot be directly implemented into energy-constrained devices unless they are lightweight or take advantage of energy-awareness. Therefore, how it is possible to reduce the impact of IoT security on energy consumption and network lifetime?

In this section, we explore the literature to determine the different categories of security solutions that could efficiently secure IoT networks while not increasing a lot the energy consumption of IoT devices and therefore, increasing network lifetime. We also introduce a new taxonomy presented in Fig. 5 for these efficient security solutions. This taxonomy extends the existing ones with five categories: lightweight protocols, energy-efficient solutions, adaptive security solutions, context-aware security solutions, and energy harvesting concepts for security.

3.3.1 Lightweight cryptography approaches

Lightweight cryptography approaches are cryptographic algorithms designed for constrained platforms. Hence, they can extend the device lifetime compared to non-lightweight cryptographic protocols. Recently, the National Institute of Standards and Technology (NIST) has finalized the Lightweight Cryptography Standardization Process and has chosen the ASCON family to be the brick of lightweight cryptography [87]. The ASCON family is a set of cryptographic algorithms that enable authenticated encryption

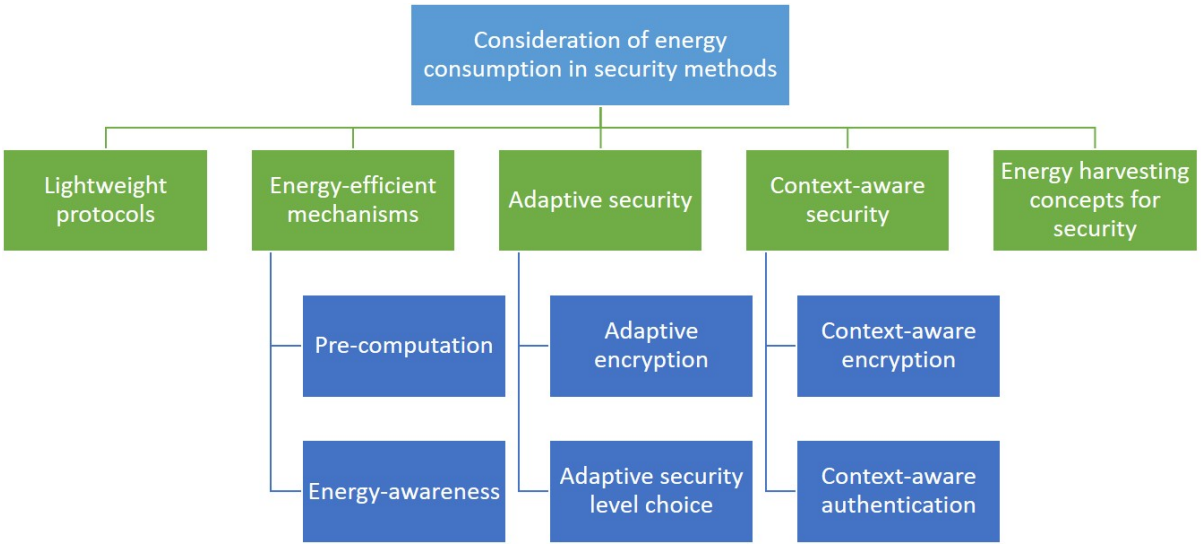


Figure 5: Categories of security solutions that are strong and energy-efficient.

and hashing.

Until now, the Advanced Encryption Standard (AES) algorithm, a well-established standard, was considered in both IoT and for the security of computer networks [54]. However, according to many benchmarks and as outlined in the previous section, the energy consumption of AES is not negligible. Thus, NIST considered that the choice of the ASCON family as the standard for lightweight cryptography is relevant to reduce the energy consumption of authenticated encryption and authentication.

Lightweight protocols such as lightweight encryption and authentication are designed to cope with the constrained nature of IoT nodes. Research in this domain is still active and led to numerous protocols and algorithms. For instance, Thakor et al. referenced existing lightweight cryptography algorithms for the IoT, classified them regarding their structure, and provided a comparative study for their hardware, software, and security performances [86].

Lee et al. provided two schemes for lightweight and mutual authentication and key agreement for IoT networks [88]. The first scheme is designed for resource-constrained devices and considers the use of the Elliptic Curve Qu-Vanstone (ECQV) which is an implicit certificate scheme. The second scheme is based on certificateless authentication and key agreement (CL-AKA) and provides a slower but higher security level. In their simulations, their schemes are faster and have a lower overhead than the majority of compared works, but the second scheme is slower than the second scheme. However, they did not study the energy consumption of their solution and they did not simulate their work in a heterogeneous network.

Seok et al. provided a secure Device to Device (D2D) communication system for

5G-based IoT networks [89]. They used lightweight cryptography based on ECC and lightweight Authenticated Encryption with Associated Data (AEAD) ciphers. A token system based on ECDSA is used between IoT nodes and general Node-B (gNB, 5G base stations). During their experiments, they observed that AES had the highest delays compared to lightweight AEAD ciphers. Their system performs basic authentication using 5G-AKA and provides confidentiality and integrity of the exchanged data. It also provides anonymity and protection against impersonation attacks, eavesdropping, privacy sniffing, free-riding attacks, and location spoofing.

3.3.2 Energy-efficient mechanisms for IoT security

Energy-efficient security methods exploit different mechanisms for energy savings while providing an adequate security level. Energy-efficient mechanisms for IoT security have one common ground: the concept of energy-awareness.

Definition 3.3.1. A system is deemed as *energy-aware* if, for one or more decisions it takes, energy is a decision variable.

For instance, if a device has to choose between security solution A and security solution B, it has the following information, presented in Table 4.

Security solution	Strength	Energy consumption
Solution A	low	low
Solution B	mid	high

Table 4: Example of security solutions with their strength and relative energy consumption.

If the security solution choice is not energy-aware, then the security solution chosen will always be in function of the security requirements of the system. In a classical setting and previous works, the main approach was to always use the highest security level or the lowest one if the budget was insufficient. Otherwise, the choice will be based on two functions: energy availability and security requirements. In this case, if energy is low, but communications need to be secured, the solution A will be used. If energy is sufficient and the threat is low, solution A is sufficient. If energy is sufficient and the threat is high, solution B should be used. If energy is insufficient, but the threat is high, either solution A should be used, with the risk of data theft or other attacks, or stop the communications.

Energy-awareness enables a better management of security solutions and, thus, improves device lifetime.

As stated previously, Hellaoui et al. studied energy-efficient mechanisms for IoT security solutions [74]. In what follows, we present recent solutions using energy-efficient mechanisms.

Ateniese et al. considered the offloading of costly security operations when energy harvesting is possible [90]. Costly security operations are, for instance, the computation of cryptographic values used in random generators or for key generation. A *delegating device* will offload computations to *delegated* devices that will compute the costly operations. Their offloading solution, called HELIOS, has three variants: tHELIOS for trusted environments and dHELIOS alongside iHELIOS for untrusted environments. dHELIOS is used to detect if there are any malicious devices in the area of the delegating device. iHELIOS is used to determine which devices are malicious (if there are any). In their experiments, tHELIOS and dHELIOS decrease the energy consumption of the delegating node, regardless of the chosen security level. On the contrary, iHELIOS increases the energy consumption of the delegating node for an increasing number of nodes and higher security levels.

Kommuru et al. provided a scheme to reduce energy consumption while ensuring an adequate security level in WSNs [91]. They used XOR encryption and asymmetric cryptography to secure the network while using PSO and LEACH to cluster nodes. They validated their solution in simulations and improved network lifetime compared to an approach only based on LEACH or PSO. The energy-efficient approach here was to cluster the devices, which improved their lifetime.

Suslowicz et al. investigated in [92] the use of pre-computed values called *coupons* for security methods in IoT networks. Their proposition is for cryptographic operations and algorithms that have two phases: offline and online phases. These coupons must not rely on the data that should be encrypted. Hence, they are computed during the offline phase of the algorithm and used during the online phase when data has to be processed. They demonstrated the validity of their approach by using it on AES-CTR for key expansion and counter increments and observed that energy consumption and latency were reduced.

An energy-efficient approach for security is to determine if data should be sent in a unique block or within multiple blocks, hence, requiring multiple security headers. Fang et al. studied this problem and proposed two algorithms to determine if data should be sent in a unique data block or within multiple data blocks [93]. Security headers induce a supplementary but fixed energy consumption (for a given security level). Each algorithm corresponds to a specific case: a case when nodes have harvested enough energy and a case when energy harvesting is not sufficient to supply the capacitor. Their simulations exposed that their algorithms achieved near-optimal results and were able to consider the available energy.

As explained earlier, energy-awareness is useful to improve the energy-efficiency of IoT security. De Rango et al. applied this principle and proposed a security solution based on ECC and MQTT for IoT networks [94]. ECC became energy-aware by assigning a key

frequency exchange to each elliptic curve length considered (193, 239, and 409-bit length). When the device has less available energy, the strength of ECC decreases by choosing a lower security level. While during their experiments, network lifetime was improved, the energy spent for requests for key re-generation increased. Thus, the energy for the communication phase increased.

Mohd et al. also showed that power-awareness is useful for adaptive encryption [95]. For each power level determined, an encryption method is used. In their experiments, compared to static security levels (use of a single encryption method with a fixed number of rounds), their method consumed 39 % less energy than a method using 2 rounds and 32 iterations. 35 % of energy is saved compared to the encryption made with one round and 32 iterations.

Yazdinejad et al. proposed an efficient SDN controller architecture to secure IoT networks while reducing the energy consumption of all devices [96]. They used two categories of blockchain since devices and SDN controllers have different capabilities. IoT devices and their associated SDN controller share a private blockchain, while all the existing SDN controllers are registered to a public blockchain. Furthermore, the authors explained that they eliminated the Proof of Work (PoW) by using the two categories of blockchain. The PoW is the main problem of the blockchain that makes it unusable by energy-constrained devices. The blockchain is used to manage malicious devices, and their ID is registered into the public blockchain if they are indeed malicious. They validated the effectiveness of their method in simulations and observed a reduced latency along with reduced energy consumption for IoT devices.

Farooq et al. presented a security framework for IoT networks with a focus on heterogeneous and constrained devices [97]. Their approach selects the security level to use according to resources and throughput constraints. They modeled the problem using a multi-objective optimization approach and solved it using the Hungarian algorithm. Compared to a greedy approach which maximizes the throughput, their approach improves throughput and resource utilization.

Recent works use energy-awareness as the central piece of energy-efficient security, along with optimizations such as the *coupons* proposed by Suslowicz et al. [92]. Energy-efficient mechanisms are a second step toward energy-efficient and strong IoT security.

3.3.3 Adaptive security solutions

Energy-awareness is important to improve the energy-efficiency of IoT security solutions. However, another approach to consider that may complement or use energy-awareness is the domain of *adaptive security*.

In IoT, adaptive security solutions are used to adapt the security service to either the different types of data or to the different threats [98, 99]. By choosing a lower security level, if there are few or no ongoing threats, devices may consume less energy. This approach is a good step toward energy-efficient security in the IoT. However, it requires a complete architecture to properly function: threat detection or data differentiation (what pieces of data must be protected). Learning techniques and game theory approaches leverage adaptive security. Furthermore, the ability of trust-based approaches to determine trusted and untrusted nodes can further improve adaptive security approaches [63, 64]. These techniques were presented in Chapter 2.

In [100], the focus is on adaptive security using Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL). The goal is to determine the optimal security policy to choose in an IoT network using 5G and User Equipments (UEs). The choice of a security solution regarding multiple parameters such as available energy, harvested energy, or consumed energy can be modeled as an Infinite Horizon Markov Decision Process (MDP). Thus, the choice of a security context (4 available levels) is energy-aware by using RL and DRL techniques. Each packet type (user plane, control plane, and network discovery messages) has a set of allowed security levels. Nodes can also harvest energy from their environment, which is considered in the environment model used in RL and DRL models.

Hellaoui et al. proposed an adaptive security framework based on coalitional games [98] to choose the optimal security level (encryption method and key length) for IoT devices during the establishment phase. During the use phase, the network uses a trust system to monitor, detect threats, and make appropriate and adaptive security decisions. They validated their framework through extensive simulations and observed a reduced energy consumption compared to a static approach where only the highest security level is used in the network.

Wang et al. provided a machine-learning based scheme for anomaly detection in Wireless SDNs [101]. They designed an adaptive anomaly detection, i.e. the strength of the anomaly detector depends on the suspected threat. It merges a lightweight anomaly pre-detector and a heavyweight anomaly detector. To determine if the threat is important and suspicious, the authors leveraged game theory techniques. The heavyweight anomaly detector uses machine learning and likelihood-based techniques to detect if suspicious flows are signs of DoS and DDoS attacks or not. Their module consumed less energy, had a better detection rate, and an overall lower false positive rate than other machine-learning based detection schemes.

Mohammed et al. presented UbiPriSEQ, a deep reinforcement learning scheme

to guarantee privacy, security, QoS, and reduce energy consumption in 5G-based IoT networks [102]. UbiPriSEQ provides security against rogue nodes and jamming attacks while ensuring privacy through Laplace mechanism. Nodes use less energy by offloading tasks to other nodes. UbiPriSEQ is evaluated in simulations and compared to an approach based on Constrained Markov Decision Process (CMDP); their approach provides better privacy, a lower latency, and a better average utility. However, the authors did not provide details on how much energy was saved and, on average and how many tasks were offloaded.

Adaptive security can also be implemented on the link layer of IoT networks. This is the research done by Mao et al. to secure IoT networks based on energy harvesting and SDNs [57]. If it is possible to predict the harvested energy for the future m time slots (as presented in Chapter 2), then it is possible to determine which security level to use at the device level for the future m time slots. Furthermore, this security level choice is threat-aware, i.e. the chosen security level cannot be inferior to the impact of the detected threat. Their simulations validated their method and improved network lifetime and throughput. Moreover, IoT nodes needing privacy protection have a higher security level than other IoT nodes with non-sensitive data.

Adaptive security leverages energy-efficiency and good security against either varying threats or different categories of data in the network. However, it requires methods to detect threats (or the different categories of data) [57, 98].

3.3.4 Context-aware security

Context-awareness for IoT security solutions allows a node to consider the context in which it operates. Context-awareness can provide a form of intelligence [103] in IoT networks. These solutions may have a reduced energy consumption compared to classical approaches. The concept of context-awareness is old and has been introduced by Abowd et al. [104]. We do not aim to detail what are the different works regarding context-awareness and security in the IoT. For instance, context-aware authentication [105], context-aware anomaly detection [106], or context-aware trust systems [107] are context-aware security methods. We are interested in the application of context-awareness to the choice of a security level or security decisions and thus, reducing the energy consumption of the security solution.

Zhou et al. provided a scheme named PRCOES to preserve the privacy of the users based on their context in a smart home environment [108]. PRCOES is also designed to reduce the energy consumption of smart home devices. Their scheme chooses, using an online RL model, the best Energy Offer (EO). PRCOES protects the privacy of the users by using Laplace mechanism on EOs and Exponential mechanism on user data. The

authors simulated a smart home environment and fulfilled user satisfaction while saving energy and preserving user privacy.

Roy et al. provided a method based on dynamic programming to provide a context-adaptive and energy-aware security for mobile devices [109]. The underlying problem is to allocate a security level to each place the user goes to, subject to energy and security constraints. The authors opted for an offline approach where places do not have preferences regarding security levels. Therefore, the problem is an allocation problem which is similar to the knapsack problem. The authors provided a greedy heuristic to solve this optimization problem and observed lower computation times compared to a brute-force approach.

Asaithambi et al. [110] continued the work done by Roy et al. [109] and provided an online algorithm for security allocation for mobile users under energy constraints. Compared to the previous work, locations require a minimum security level. They provided two algorithms to tackle this problem: a greedy algorithm and an efficient algorithm. They observed during simulations that the benefits of the efficient algorithm are higher than those of the greedy algorithm. However, the greedy algorithm always allocates a security level, as opposed to the efficient algorithm, which is a clear limit.

Massad et al. provided a scheme called MQTTSec (Secure MQTT) enhancing MQTT v5 [111]. MQTTSec consists of a selection algorithm, CASA, to choose an encryption algorithm given the context and available energy. MQTTSec also enhances CONNECT and CONNACK messages by adding new fields to those messages. They created a small test bed and considered AES, DES, RSA, and Blowfish for the set of available encryption methods. Authors stated that MQTTSec provides security against multiple attacks such as broker impersonation attacks, eavesdropping, chosen plaintext attacks, chosen ciphertext attacks, man-in-the-middle attacks, and cryptanalysis.

Thus, context-awareness is a valid approach to protect IoT networks and their users while consuming less energy. If the context is not deemed important (no users or important processes), context-aware security methods reduce the security level. On the contrary, if the context is critical, the security level is increased. *Context-aware* security is complementary to *adaptive security*. The former focuses on what security solution or level to apply according to the current context, while the latter determines which security level to use according to the current threat level or classes of data.

3.3.5 Energy harvesting, wireless charging, and energy transfer for IoT security

The concepts of energy harvesting, energy transfer, and wireless charging may be used to improve the energy availability in IoT networks and, thus, have enough energy

to deploy strong IoT security solutions (which may be energy-aware or even adaptive). Indeed, Schaumont et al. presented pre-computed values called *coupons* for cryptographic algorithms [81]. Devices may calculate these *coupons* during time windows where the harvested energy is high or when energy may be wasted. Indeed, computing *coupons* during harvesting periods may reduce the impact of cryptographic algorithms on the battery of the device. For costly security operations, Ateniese et al. provided an offloading scheme for devices that have an excess of energy [90].

Mao et al. considered that to choose an optimal security level for a future time slot, the corresponding energy harvesting prediction is available [57]. Knowing how much energy may be harvested in current and future time slots helps in the final decision regarding the choice of a security level (among predefined levels). In another research work for 6G-IoT networks, the same authors applied an extended Kalman filter to predict the amount of harvested energy for the future time slots [112]. Their goal was to balance the network lifetime and the security of the network. The selection of a security level remains the same as their previous work [57]. The prediction of harvested energy has a central place in these security solutions; thus, it is necessary to be able to predict well the amount of harvested energy. Many research works focus on the prediction of harvested energy [39–41, 113, 114]. On one hand, having a precise prediction for the future time slots of the harvesting process improves the energy management of the network. On the other hand, if the prediction is available and if there are multiple available security levels against one threat, the device may choose the security level that consumes the least energy and is relevant to the threat. This approach efficiently limits the impacts of security on available energy, thus, increasing network lifetime.

Energy harvesting can also be used to protect devices against attacks on energy. Indeed, Cheng et al. provided a way to mitigate Denial of Energy (DoE) attacks by using wireless charging signals to build a new communication channel [115]. This method enables Power-Positive Networking (PPN) on the receiver side. Moreover, for network protocols based on one exchange (request-reply) between a requester and a receiver, the energy consumption of these protocols is the burden of the requester. It means that each time a device A wants to communicate (a requester) with another device (a receiver) when PPN is enabled, the requester sends energy to the receiver. When a DoE attack happens (which is a special case of DoS attacks), in the PPN framework, the DoE attack is completely mitigated because each time an attacker sends requests to a victim, the requests charge the battery of the victim, which is the characteristic of power-positive networking.

Thus, the principle of wireless energy transfer (moreover, from an attacker) is useful to nullify attacks against energy. Furthermore, energy harvesting can provide the devices with the energy intake needed for the use of strong security solutions.

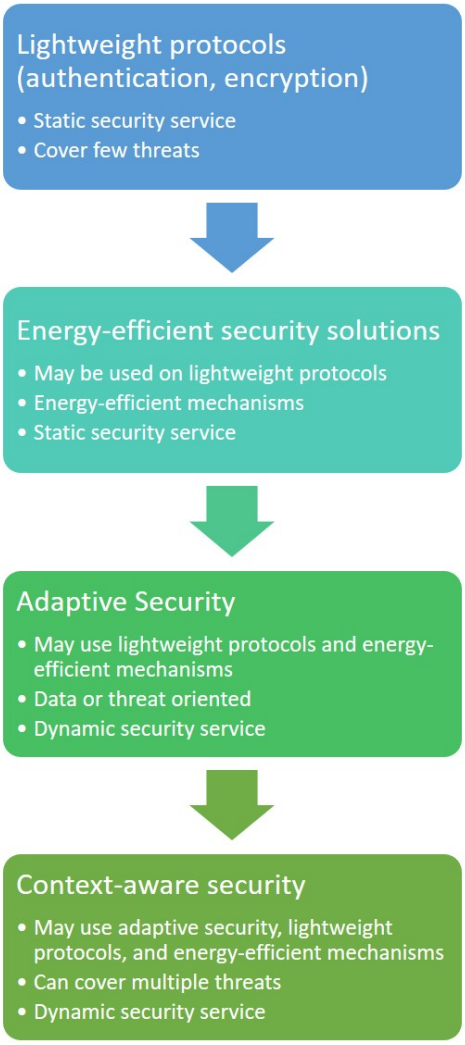


Figure 6: Characteristics of surveyed IoT security solutions that may save energy while providing an adequate security service. Complexity, flexibility, and potential saved energy increase from top to bottom.

3.4 Summary and discussion

As IoT is used in many domains, some solutions are more appropriate due to the consideration of domain-related parameters and environment constraints. In Fig. 6, we remind the different categories of solutions we surveyed in Section 3.3. Both security and device lifetime are QoS criteria for IoT networks. On one hand, there is a need for different security levels to tackle different threats. On the other hand, device lifetime is also an important QoS criterion and high energy consumption reduces device and network lifetime. From the previous section, we observed that the stronger the security level is, the higher the energy consumption is. Thus, one cannot ask for both a high device lifetime and high security level at the same time: trade-offs have to be made.

3.4.1 Summary of studied solutions

This classification helped us to identify the main blocks and concepts to develop energy-efficient and strong security solutions for IoT networks.

Firstly, lightweight encryption and authentication methods should be used as the first building blocks. Lightweights protocols [86, 88, 89] (authentication, encryption) are useful for resource-constrained nodes since computational power and energy are limited. However, these protocols are static and offer only a fixed security level. They need to be combined with other methods to have a better consideration of energy and threats.

Secondly, energy-efficient mechanisms (and energy-awareness) are the second block to consider to reduce the energy consumption of this security solution. Energy-efficient security solutions can consider the use of lightweight protocols and use energy-efficiency mechanisms (described in [74]) to lighten the energy consumption of such protocols. They may also adapt the security service to the remaining energy but not necessarily to threats, data, or users. Thus, they provide fixed security levels against varying threats.

Thirdly, adaptive security concepts may prove useful to continuously adapt the security level to a plethora of threats [57, 63, 98, 102]. In our study, these solutions are energy-aware and may use various lightweight protocols to provide a suited security level with a decreased energy consumption. These solutions are dynamic with regard to the provided security service. The choice of an adapted security level instead of a static security level saves energy in the long run.

Then, context-awareness provides additional information from the environment and the users to the security solution in order to fine-tune the choice of a security level. Due to the use of multiple data sources (historical, environmental observations, network traces, trust sources, etc.), implementing a context-aware security solution is far more complex than using a lightweight protocol. The dynamism behind context-aware security solutions makes them useful and appropriate for mobile IoT nodes [109, 110]. It may appear natural to merge context-aware security and adaptive security to exploit possible synergies between them.

Last but not least, energy harvesting concepts may enhance security against attacks on energy and provide sufficient energy to power adaptive security solutions.

Combining the concepts of adaptive security and context-aware security may improve security while reducing the energy consumption of security tasks. On one hand, if the environment is safe and fully trusted, a low-security level might be applied to save more energy. On the other hand, if the environment becomes insecure, the highest security level may be applied. Furthermore, for sensitive events and applications, context-aware

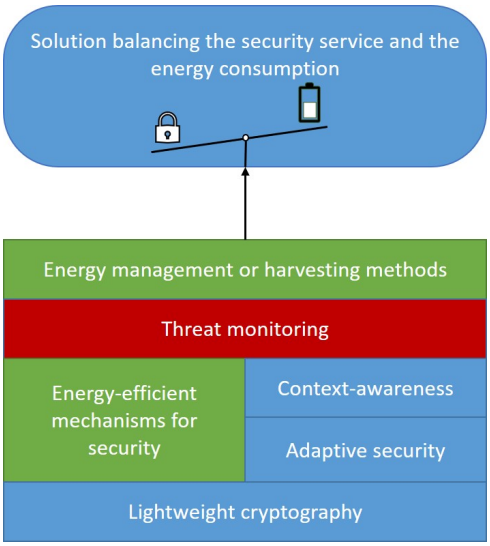


Figure 7: Elements needed to provide a security solution balancing the provided security level and energy consumption.

security can further improve the security of the system. Moreover, network administrators and developers may use threat monitoring systems to improve the choice of a security level, along with context-aware and adaptive security modules. Threat monitoring process and intrusion detection can also be adaptive according to a recent research work done by Wang et al. [101]. Furthermore, energy harvesting approaches can be used to either power the devices or strengthen their security. Energy harvesting concepts may also be used to provide energy to the system or secure it [115]. Figure. 7 presents an overview of the key elements to consider when designing an IoT security solution that can minimize energy consumption in both trusted and untrusted environments while maintaining an adequate security level.

3.4.2 Remarks on surveyed works

The surveyed works in this chapter showed that it is possible to secure well IoT networks while reducing their energy consumption. Compared to existing surveys on IoT security [10, 37, 47, 59, 116–119], the number of surveys on energy-efficient security is low [37, 74, 75]. Hellaoui et al. focused their study on energy-efficient mechanisms for IoT security solutions [74], but they only considered encryption-based, authentication-based, or signature methods. On the contrary, Tedeschi et al. surveyed many solutions for energy-constrained or batteryless devices, with a focus on PHY-layer security solutions [37]. Yousefpoor et al. tackled the problem of secure data aggregation, which is central in current and future IoT networks [75]. Data aggregation reduces the energy consumption of the devices, but it is important to secure the aggregation and sending processes. While these surveys offer a deep understanding of a particular security facet, they do not consider a

general approach to IoT security that encompasses the IoT network as a whole. Nowadays, there are research works that consider encryption-based methods as a block of the global solution, as opposed to being the proposed solution. Although encryption increases the energy consumption of IoT devices, it is mandatory as the number of threats and eavesdroppers increases.

The encryption-based methods used as the building blocks of energy-efficient security solutions are not always lightweight. Indeed, lightweight encryption methods (such as SPECK or SIMON) are not the primary choice. Practitioners favor strong security primitives such as AES or ECDSA [53, 79]. It is also possible to offload heavy computations of some security primitives, according to Ateniese et al. [90]. It may increase device lifetime while guaranteeing a good security level unless the environment is made of a majority of malicious devices.

Energy-awareness is one of the keys to energy-efficient and strong security solutions. For instance, Mohd et al. provided power-aware encryption [95]. However, their approach is static with regard to threats because, with decreasing energy levels, the security provided by the cipher decreases. Other methods using learning approaches [100, 102] or game-based approaches [98] for 5G-based IoT networks provide sufficient security against adaptive threats while considering energy constraints.

Context-aware security and privacy solutions use multiple data sources (historical data and contextual data, neighbor nodes, and servers) to secure an IoT network (or node). The solutions we have surveyed are energy-aware and context-aware. Both [109] and [110] considered context and the user's energy budget to choose a security level when they arrive in a new place. Context-awareness combined with energy-awareness may provide improved security and better energy management. These solutions may consume less energy compared to a static approach. However, in [109] and [110], authors did not provide comparisons with static approaches for the energy consumption. What is the result if only the highest (or lowest) security level is used in each place?

Many research works on IoT security we surveyed considered that energy harvesting or energy transfer are elements of the system model, but few research works used the core concepts of energy harvesting, wireless charging, or energy transfer to improve the security of IoT networks. To the best of our knowledge, only Chang et al. studied the case where the defense mechanism is the energy transfer mechanism, which is based on inductive charging channels [115]. Nevertheless, designing harvesting-aware security solutions is necessary since future connected devices will rely more on energy harvesting to operate. Mao et al. [57, 112] considered this approach: they provided an adaptive security level choice according to observed threat and predicted harvested energy.

3.4.3 Issues and challenges

There is an urgent need to design security solutions covering multiple threats and suited to heterogeneous IoT networks. However, as they are resource-constrained, it is impossible to efficiently cover each existing threat when devices are energy-constrained. Depending on the application domain, some threats are more present and should be the focus of the security system deployed. Many issues and challenges arise when implementing security solutions, even if they are energy-efficient or adaptive. The next points illustrate the issues with energy-efficient and adaptive security.

Security of highly energy-constrained or batteryless IoT devices

IoT networks deploy small devices that may operate with a very small battery or even be batteryless. Batteryless devices rely on energy harvesting to function, and harvested energy is stored in a capacitor [120]. For instance, passive RFID devices are batteryless [121], thus, relying on the signals of the reader to be powered. In this case, it is near impossible to use cryptography-based or learning-based security solutions [37]. The main approach is to consider data-secrecy methods at the PHY-layer along with supplementary countermeasures [122, 123]. Protecting such devices is a challenging task, and many researchers explore the aforementioned categories of solutions to secure these devices. Research also focuses on the energy-efficiency of such devices to improve the harvesting efficiency, thus, having more energy for their security or application needs. Given that the trend is to miniaturize devices to deploy them everywhere at a low cost, their energy storage decreases. However, at the same time, more and more threats appear in the IoT ecosystem [48]. Thus, it is challenging to secure IoT networks made of heavily energy-constrained devices, even if the security solution is threat-aware or context-aware.

The use of non-suited security primitives

For battery-powered devices, it is possible to use security primitives and other solutions to secure them. However, as presented in Section 3.2.1, the energy consumption of security primitives reduces device and network lifetime. With stronger security primitives, the energy consumption of devices increases, and thus, their lifetime decreases. Thus, it is necessary to favor the use of lightweight security primitives within IoT security solutions. To this end, NIST launched a standardization process for lightweight cryptography methods [124] in 2015, with a first draft in 2016, and finalized it in 2023 [87]. Lightweight security primitives exist, but AES is still a favored choice among practitioners in the field because it secures well communications and has been challenged many times to determine its vulnerabilities.

The impact of learning methods on the battery lifetime

Although security solutions based on learning approaches deliver better results thanks to the detection of new or even unknown attacks, their use has some drawbacks. First, for machine-learning and deep-learning based approaches, there is a need to train the models beforehand. The training process can be long and consume a lot of energy, depending on the complexity of the model [125]. If these solutions are deployed in devices with an unlimited amount of energy (fog or cloud), such as an IDS, it will not impact their lifetime. However, if heavy security solutions based on learning algorithms are deployed in energy-constrained devices, the different training phases will consume a lot of energy and time, depending on the complexity of the algorithm. Thus, the use of lightweight learning algorithms is an appealing approach. It is the approach of Wang et al. who merged heavyweight and lightweight anomaly detectors [101]. For both detectors, they consider machine learning-based techniques which are lightweight compared to deep learning-based approaches.

If an intrusion (or anomaly) is detected, then adequate security decisions should be taken. Using threat detection to adapt the security level of IoT devices is promising and leads to energy savings [57] while efficiently securing the network.

However, as explained before, learning mechanisms tend to be energy-consuming. Yet, having an excellent estimate or the true consumption of a learning model is a hard task [125]. Also, the process of anomaly and intrusion detection is energy-consuming [126]. This intelligence can only run on devices if and only if the energy storage is big enough to handle the process. Otherwise, it will greatly reduce the lifetime of the device. In Table 5, we sum up the characteristics of each security solution using a learning method and if the solution has reduced energy consumption.

Guaranteeing energy availability for energy-efficient, adaptive, and smart security

If IoT security is energy and threat-aware, then security decisions can be adapted to each dangerous situation while saving energy. There are research works that are dependent on the future harvested energy. The considered networks need to either have harvesting devices or charging strategies available. These requirements are on the architecture side of an IoT network, and the security of such networks has to be planned from the beginning.

Moreover, mobile chargers [135–137] may also be considered to extend network lifetime and reduce maintenance costs. However, the use of such chargers has an important monetary cost, and recharge time relies on antenna efficiency and distances. If there are unreachable nodes or the mobile charger cannot move in the environment, other methods

Reference	Application domain(s)	Algorithm(s)	Real-time	Scalability	Reduced energy consumption of security?
Wan et al. [127]	Intrusion detection in smart home	Best results: Random forests + PCA	Yes	No	Not studied
Kulkarni et al. [128]	Mitigation of DoS attacks in generic WSNs	Multi-layer Perceptron	Yes	Yes	Not studied
Yang et al. [129]	Intrusion detection in generic IoT	Variational autoencoders	Not studied	Not studied	Not studied
Shahid et al. [130]	Intrusion detection in smart home	Sparse autoencoders	Yes	Left for future work	Not studied
Vu et al. [131]	Attack detection in IoT networks	Autoencoders and transfer learning	Near real-time	Yes	Not studied
Fan et al. [132]	Intrusion detection in 5G-based IoT	Federated learning, transfer learning, convolutional neural networks	Not studied	Yes	Not studied
Tan et al. [133]	Intrusion detection in UAV networks	Deep Belief Networks	Not studied	Not studied	Not studied
Tu et al. [134]	Impersonation detection in Fog-based IoT	Q-learning	Yes	Yes	Not studied
Wang et al. [101]	Anomaly detection in Wireless SDNs	Statistical learning (contrastive pessimistic likelihood estimation)	Yes	Yes	Yes
Conceição et al. [100]	Dynamic security in 5G-based IoT	SARSA, Q-learning, Double Q-learning, Actor-critic (Deep-RL)	Yes	Yes	Yes
Zhou et al. [108]	Privacy in smart home	Contextual bandits (RL)	Yes	No	Yes

Table 5: Security solutions based on learning methods presented in this chapter.

to charge batteries and operators may be required. These mobile chargers may also be mobile nodes dedicated to heavy computations. Indeed, computation offloading in Mobile Edge Computing (MEC) nodes is a topic of interest in research [138]. Offloading and outsourcing security operations in mobile robots could be an interesting way to manage heavy security operations, and thus, energy-constrained devices may save more energy. However, to the best of our knowledge, no work considered the use of mobile chargers to help secure IoT networks as shown in Table 6 and as discussed in Section 3.2.

Security approaches / Energy approaches	Energy-efficient mechanisms for security	Adaptive security	Context-aware security
Energy management methods	[91, 96]	X	X
Energy harvesting	[90, 92, 93, 115]	[57, 73, 100]	X
Wireless charging	X	X	X
No particular mechanism used	[94, 95, 97]	[98, 101, 102]	[108–111]

Table 6: Classification of studied works with regard to energy management or harvesting methods they use and the security classes they belong to.

In fact, energy harvesting and energy saving mechanisms can lead to energy savings when used in security solutions [81, 92]. Having a prediction of the future harvested energy also helps in the choice of a security level [57, 112], and reduces energy consumption. If hardware constructors design harvesting units with dedicated MCUs for cryptographic operations [139], other MCUs or chips can have more energy dedicated to other tasks and balance security with energy consumption. However, according to the authors in [140], asynchronous duty-cycling may negatively impact the energy consumption induced by security solutions. This point requires further research for different duty-cycling based protocols. Chang et al. provided an interesting approach for mitigating denial of energy attacks by using power-positive networking [115], but their method only works for short distances, considers only small devices, may not scale well, and their scenario only considered devices under an ongoing attack.

In Table 6, we classify the IoT security solutions that may use (or be built upon) energy management or harvesting methods to further reduce energy consumption (and gain energy).

In future IoT networks, mobile chargers and energy transfer architectures may be used due to the promising performances in network lifetime improvement and applications [37, 43]. A good research direction would be to study wireless mobile chargers and their impacts on the security of IoT networks. Furthermore, they may have good results in

mitigating energy-based attacks.

Software-defined networking and energy-efficient security

SDN along with Network Function Virtualization (NFV), are enablers of energy-saving security solutions. Indeed, Rawat et al. surveyed both energy-efficient mechanisms and possible security solutions for SDNs [141]. Yazdinejad et al. proposed an energy-efficient SDN controller [96] along with the use of blockchain technology (public and private blockchains) to secure and reduce the energy consumption of all devices in the network. SDN may be used along with 5G networks [101]. Furthermore, the energy consumption of anomaly detection can be reduced if well designed in an SDN architecture [101]. However, this approach is an architectural approach that requires the devices to be enabled with SDN capabilities (or using gateways that enable the compatibility between SDN and legacy communications).

Then, the deployment of virtual network functions such as threat, anomaly, or false data detection is easier in an SDN environment, as explained in Chapter 2. However, there is still the challenge of integrating legacy and traditional IoT and WSNs networks with SDN-enabled IoT networks.

3.4.4 Towards energy-efficient and strong security

The design of security solutions has to consider the remaining energy since the impact of security on energy consumption is non-negligible thanks to energy-awareness [20]

Then, learning-based methods can leverage trade-offs between ensuring a good security level and the energy consumption of IoT devices. Indeed, Mohammed et al. considered this approach thanks to deep reinforcement learning to optimize QoS, security, privacy, and energy consumption for 5G IoT networks [102]. Conceição also used reinforcement learning to dynamically attribute security levels along with the use of energy harvesting in 5G IoT networks [100]. On the contrary, authors in [57] favored an approach based on optimization to find the best security suite for a given time cycle. However, using learning methods such as reinforcement learning or deep learning incurs additional complexity, and sometimes, these solutions may not scale well. Some solutions considered real-time and constrained environments (such as UAV networks [133]), but no practical information on the feasibility is given. If the training phase occurs in an energy-constrained device or during a period where available energy is low (in the battery or in the environment), the device may run out of energy. Moreover, during the lifespan of the device, multiple model training may occur to adapt the model to the dynamic environment, thus, increasing the energy consumption of the device. Transfer learning [131, 142] may alleviate the

devices from this energy-consuming process. Thus, learning-based security solutions for IoT networks should provide the energy consumption of the training and use phases, and if training occurs multiple times, what would be the corresponding energy consumption?

To have a good balance between ensuring a good security level while reducing energy consumption, a general approach would be to:

1. Determine the current threat level thanks to anomaly and intrusion detectors. Learning-based solutions can be used.
2. Check past security records and decisions taken. If there is a similar security situation in the database, take the same security decision.
3. Otherwise, choose the best category of encryption-based methods to protect confidentiality and integrity while considering energy constraints. This choice can benefit from learning methods.
4. If other methods are needed (secure routing, friendly jamming, etc.), use them.
5. Append to the current security record the taken security decisions and store it in the database.

In a complex IoT environment, the available security solutions are not always linked to cryptography or link-layer security such as authentication but may be linked to trust-based approaches or anomaly detection (detect false data or abnormal energy consumption or CPU usage) [143]. As explained before in Section 3.3.3, trust-based IoT security solutions can detect malicious devices while reducing the energy consumption of the devices [64]. Indeed, the security level can be reduced with trusted devices while it will be increased for communication with less trusted devices.

Research works done by Mao et al. [57, 112] took a similar approach to determine the best security level according to the remaining energy and predicted threats. The security at the link layer is made dynamic and threat-aware thanks to the use of an intrusion detection system.

We believe that researchers should pursue further research to achieve a good balance between security level and energy consumption to improve network lifetime while efficiently securing IoT devices. Moreover, combining adaptive security and context-aware security may improve network protection against advanced threats. Learning-based solutions can quickly adapt against threats and fine-tune the security decision, reducing energy consumption compared to a fixed security level. In addition, such solutions may consider energy constraints, user needs, security requirements, and other attributes to continuously adapt the security services with regard to the available resources. Software-Defined

Networking is also a good enabler for energy-efficient and strong security solutions since it makes the deployment and modification of security services easier. There is research in the field of green IoT, energy-efficient IoT, security for IoT, and energy-efficient security, but research tackling both green IoT and energy-efficient security in IoT is scarce. Authors in [144] advocate the need for research in the field of sustainable security for IoT. We also think that more research needs to be done in this field. The energy consumption of security solutions cannot be ignored anymore when devices are getting smaller and smaller, with less available energy.

3.5 Conclusion

With an ever-increasing number of threats and numerous deployments in critical applications, it is necessary to protect IoT networks that require the deployment of security services. However, the deployment of such services induces additional energy consumption which may reduce the network lifetime. Hence, it is important to consider energy constraints while designing IoT security solutions.

In this chapter, we first presented related theses and surveys in the field of energy-efficient security and outlined the limits of these works. Second, we presented the energy consumption of encryption-based security methods and the impacting parameters of such methods on energy consumption. Then, we proposed a new taxonomy extending the existing ones by presenting, discussing, and comparing recent security solutions aiming for a good security level while decreasing energy consumption. Afterwards, we used this classification to propose a model for a general IoT security solution that is both energy-efficient and adaptive against threats. Finally, we discussed recent advances such as Software-Defined Networking (with Network Function Virtualization) and learning techniques for the design of energy-efficient and robust security solutions. Learning techniques leverage intelligence while Software-Defined Networking improves the deployment of security modules thanks to Network Function Virtualization.

However, existing approaches for energy-efficient IoT security lean more toward energy-efficient mechanisms and the adaptation of the strength of the security mechanisms. No works tackled adaptive energy management and adaptive energy provisioning approaches for IoT security. This approach may increase network lifetime by considering the energy requirements of IoT security and efficiently managing the available resources to ensure the continuous operation of IoT security. Thus, this literature study on energy-efficient security for IoT networks gave us an interesting research direction: considering wireless charging alongside context awareness and then threat awareness to maximize network lifetime. In Chapter 4, we will present the problem of context-aware wireless charging,

while in Chapter 5, we will study the problem of threat-aware charging.

Chapter 4

An efficient context-aware approach for IoT wireless charging

IoT networks deploy numerous energy-constrained devices, making the efficient management of their energy resources mandatory to ensure high network lifetime. To this end, various approaches such as energy management techniques, energy harvesting, or wireless charging approaches have been explored to increase device and network lifetime.

In the previous chapter (Chapter 3), we observed that security solutions increase the energy consumption of IoT devices. Despite the use of energy-efficient, adaptive, or context-aware security solutions, energy is still consumed, thus, reducing network lifetime. An innovative approach to address this issue would be to provide energy for their security needs based on the current threat status. However, there is a general approach that has yet to be tackled in the literature: adapting the charging path to the events in the environment. The problem of context-aware wireless charging is a building step to study the problem of threat-aware wireless charging, which is further presented in Chapter 5. In dynamic environments, some devices may go to sleep because they do not have to manage an event whereas other devices may have to wake up to manage this event. This leads to variable energy consumption among devices. Therefore, one research question appears: 'How can the knowledge of ongoing events and the prediction of future events improve the lifetime of an IoT network?' This knowledge of ongoing events and predicting future events is tackled with context-awareness approaches. Context awareness gives information on the ongoing event(s) in the environment, their criticality, and their impact on the devices. Thus, it is possible to determine the device that will require energy to effectively process upcoming events. Notably, existing wireless charging strategies lack context awareness in their trajectory planning, whether they are offline or online, which motivates us to propose a context-aware wireless charging strategy for IoT networks.

We start this chapter by introducing the fundamentals about mobile wireless charging. Then, we provide background on Markov Decision Processes (MDPs), Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL), leading to the presentation of wireless charging strategies using these techniques. Furthermore, we show that these

research works lack context awareness. Thus, we present our system model and the different hypotheses considered to overcome the limits of the related works. Furthermore, we discuss and justify the use of context awareness and present a general approach to how it could be used by a Mobile Charger (MC). Then, we formulate the problem of context-aware charging as a Markov Decision Process (MDP) and justify the use of deep reinforcement learning to solve it. Finally, we present an algorithm describing the different steps of the context-aware charging strategy.

4.1 Fundamentals of wireless charging

As introduced in chapter 2, wireless charging approaches can increase device and network lifetime. We first introduce background on wireless charging, and then we present related works in the domain.

4.1.1 Background on wireless charging

Wireless charging is a special case of energy harvesting since a dedicated energy source (fixed or mobile) sends energy signals to devices. These energy signals carry Radio-Frequency (RF) energy. There are two main approaches for energy transfer and wireless charging [145]:

- Far-field wireless charging
- Near-field wireless charging

For both approaches, a source emits a signal with one or multiple antennas (in our case, a wireless charger), and a receiver receives this signal (the IoT device). Then, the IoT device can harvest with one or multiple antennas the energy contained in the signal to charge its battery and power itself.

In the far-field wireless charging approach, the charging signals are emitted by far-away stations. These signals may either be propagated or not by intermediate devices. Then, the devices harvest the energy contained in the signals. In the near-field wireless charging approach, the energy signal is near the device, which may either come from a fixed station or a mobile charging device (e.g. a wireless mobile charger).

To be able to harvest the energy from a signal, an IoT device needs to have a particular architecture. Mishra et al. [21] presented the aforementioned architecture, which is made of:

- One or multiple antennas that capture the signal,
- A matching circuit to maximize the power transfer process [146],

- A voltage multiplier,
- An energy storage to store the converted energy,
- A power management unit to manage the power.

RF energy is abundant in the environment [19, 21, 147]. However, its main problem is that its power density is very low compared to other energies. Furthermore, if the model is based on Friis's free space equations, the energy received by the charged device is inversely proportional to the square of the distance [46, 135]. Then, to maximize the amount of received RF energy, the source has to be the closest as possible to the receiver. A solution to this problem is to consider a mobile node with energy transfer technology to charge the devices. In the literature, this special mobile node is called a Wireless Mobile Charger (WMC) or sometimes a Mobile Charger (MC, the word wireless may be omitted). Thus, in the following sections and chapters, we may use WMC and MC interchangeably.

4.1.2 Wireless charging and network lifetime maximization: related works

In the past years, research on wireless charging strategies has attracted a lot of interest. According to Yang et al., wireless charging strategies can be categorized into different classes [148]:

Periodical wireless charging strategies: These strategies aim to establish a charging path for each charging period, which begins and ends in a safe zone.

On-demand wireless charging strategies: These strategies consider the recharge requests issued by the devices to determine the charging order. These strategies are more adaptive than periodical wireless charging strategies.

Proactive or dynamic wireless charging strategies: These strategies may consider both recharge requests, remaining energy, or energy needs of devices that did not issue recharge requests. They are more flexible than on-demand wireless charging strategies.

There are three main approaches in the literature to plan a charging path:

- Through linear or nonlinear programming,
- Through genetic algorithms,
- Through Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL) approaches.

The domain of wireless charging in Wireless Sensor Networks (WSNs) has attracted many researchers in the past years due to the promising results for network lifetime extension. Thanks to the results of Wireless Energy Transfer (WET) technologies [46], Mobile Chargers (MC) are investigated to extend device and network lifetime. There exists charging strategies using heuristics and charging strategies using learning methods. We focus our study on terrestrial MCs, even if unmanned aerial vehicles [45, 149] may be considered for charging devices. Many works use heuristics, linear, or non-linear programming to determine a charging path. Furthermore, some of them consider that the charging path planning problem is a TSP. Thus, operational research tools can be used to solve these problems.

Wang et al. [31] provided a non-learning scheme using energy harvesting devices and mobile wireless charging to maximize device lifetime. They introduced the use of partial recharges to charge the devices. The underlying problem is similar to the Traveling Salesman Problem with Neighborhoods (TSPN), which cannot be solved efficiently. The designed algorithm for optimizing recharge times has a time complexity of $O(n^3)$ in the worst case (n is the number of wireless-powered nodes in a tour). Their experiments demonstrated that these partial recharges improve network lifetime. Indeed, partial recharges imply that the WMC stays less time at the device position to charge them, allowing more devices to be charged. However, the approach is offline.

Abid et al. [136] provided three on-demand mobile charging strategies for an architecture based on solar energy harvesting and wireless charging in WSNs: DDP, PDP, and PDPP. These strategies take advantage of the multiple Energy Harvesting Base Stations (EHBS) deployed in the network and Mobile Chargers (MC) to maximize network lifetime. Their experiments outlined that the PDP strategy outperforms the other strategies. Furthermore, the deployment cost of the network is lower when the number of EHBS is low. However, their on-demand strategies are not context-aware: they only consider network characteristics and remaining energy.

Na et al. tackled the problem of charging multiple IoT devices at once and planning the charging path [135]. They provided two non-learning algorithms, namely Best Charging Efficiency (BCE) and Branching Second Best Efficiency Algorithm (BSBE) to solve the problem. In their experiments, BCE is computationally faster than BSBE. Nevertheless, BSBE has a better charging cost and performs better than BCE in large networks. However, their approach is an offline approach, which is inefficient when unexpected events occur during the tour of the charger.

Gharaei et al. [137] provided two non-learning algorithms for the optimization of charging tours and charging time. Their algorithms are designed for the use of two MCs.

One MC balances the energy in the network, and a second MC charges devices to keep them above a predefined threshold. The main contribution of their research is the consideration of the variance of the remaining energy of the devices. Their approach achieved valuable results and increased network lifetime compared to other works. However, their approach is offline, and it is only activated when the variance of the remaining energy of the devices goes above a threshold.

Although the presented research works increase network lifetime, they use offline approaches to determine the charging path which are inefficient when considering dynamic environments. Furthermore, there is a major downside of using linear or non-linear programming to determine a charging path: these approaches give the whole charging path given the state of the network at a given time, but they do not give an adaptive charging path given potential changes in the network [31, 137, 149]. These approaches are suited for offline solutions but are unsuitable for online path planning in dynamic environments. Online path planning and other online decision-making problems can be solved using Reinforcement Learning (RL) approaches. In the next section, we introduce Reinforcement Learning (RL) and the associated modeling tool: Markov Decision Processes (MDPs).

4.2 Reinforcement Learning: a novel approach for wireless charging

As explained before, the main problem of non-learning-based approaches is their difficulty in adapting to varying environments and changing situations. Reinforcement Learning (RL) and Deep Reinforcement Learning (DRL) are two learning frameworks able to adapt to varying environments and, thus, are suited for wireless charging strategies. Reinforcement Learning (RL) draws its origins from multiple sciences: psychology, control, mathematics, etc. It has many application domains, ranging from autonomous driving to IoT networks [65, 150–152]. In recent years, Deep Reinforcement Learning (DRL) has emerged to solve more complex problems with improved performance compared to reinforcement learning. In this section, we present fundamentals regarding reinforcement learning and deep reinforcement learning. We also present applications of RL and DRL in IoT networks and for wireless charging. We finally justify our approach for a context-aware wireless charging strategy.

4.2.1 Fundamentals of Markov decision processes

As presented before, RL and DRL are based on an agent that takes an action at each time step. Both frameworks build on a common theoretical ground: the Markov Decision Process (MDP).

Formally, an MDP is defined as a 4-tuple (S, A, R, P) in which:

1. S is the state space of the MDP. The state space represents how is the environment and what are the variables the agent is looking at.
2. A is the set of actions of the MDP. The action space represents what are the actions an agent can take.
3. R is the reward function of the MDP. It represents how the agent is performing (well or badly).
4. P is the probability transition state function of the MDP. It is the probability of being in the new state s_{t+1} knowing that action a_t was taken in state s_t.

The probability transition state function P can be known or not. It is defined as $P(S_{t+1} = s_{t+1}|S_t = s_t, A_t = a_t) = P(s_{t+1}|s_t, a_t)$, where S_{t+1}, S_t, and A_t are random variables. This probability transition state function represents the dynamics of the environment [151]. It leads to an important property called the *Markov property*, which states that the future state of the environment is only based on the current state and the chosen action, i.e. there is no memory of the past states and taken actions (the process is *memoryless*). More formally, the future state is conditionally independent from the past given the present state and action [151, 153]. It can be written as:

$$P(S_{t+1} = s_{t+1}|S_t = s_t, A_t = a_t, S_{t-1} = s_{t-1}, A_{t-1} = a_{t-1}, \dots) = P(S_{t+1} = s_{t+1}|S_t = s_t, A_t = a_t) \tag{4.1}$$

The previous equation implies that the time is discretized and the agent interacts with the environment at each time step t. It is important to note that the duration between two timesteps i and j may vary.

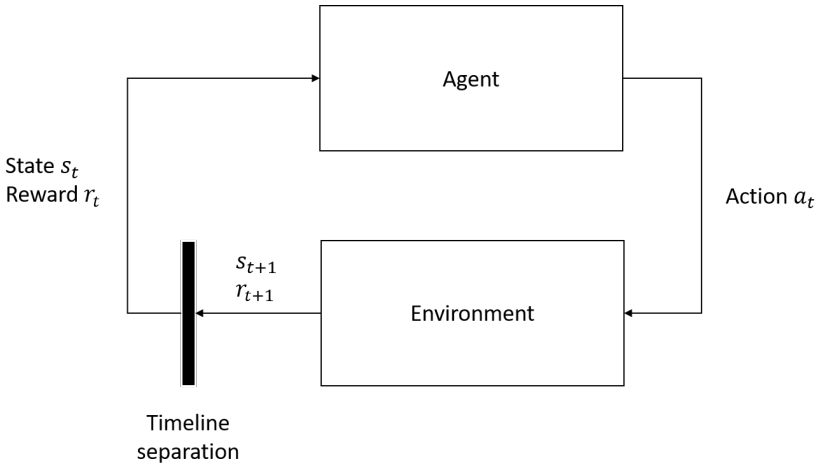


Figure 8: Interaction model between an agent and the environment in an MDP [65, 151]

When the MDP is ongoing, there is a succession of random variables tied to state observations, actions taken, and received rewards: $S_0, A_0, R_0, S_1, A_1, R_1, S_2, A_2, \ldots$ [151]. This succession is the core of the agent-environment interaction loop depicted in Fig. 8: an agent gets from the environment the current state s_t and the reward from the previous action taken r_t, determines what action a_t it should take, apply it, and then, it modifies the environment to the next state s_{t+1} and the agent receives the reward r_{t+1}. The time horizon can be finite, i.e. the agent will only interact with the environment a given number of times, infinite, or indefinite.

Definition 4.2.1. A finite-horizon MDP is an MDP in which the agent interacts with the environment for a **fixed** number of interactions [151].

Alongside finite-horizon MDP, there are also indefinite-horizon MDPs and infinite-horizon MDP.

Definition 4.2.2. An infinite-horizon MDP is an MDP in which the agent interacts with the environment for an **infinite** amount of interactions [151].

Definition 4.2.3. An indefinite-horizon MDP is an MDP in which the agent interacts with the environment for an **arbitrarily long number** of interactions, but the **interaction can terminate** due to a terminating state [151].

In the domain of WMC, the MDP is an indefinite-horizon MDP (the MDP can end early if the charger dies or later if the charger can keep the network alive for a long period).

4.2.2 Classification of Reinforcement Learning algorithms

Reinforcement Learning (RL) has emerged in the 80s [151]. A smart agent has to accomplish some tasks in an environment. The agent will learn what actions to take at

each step to maximize a reward signal. This is similar to human children when they learn a new skill: when they succeed, they earn a new skill and rewards. If they fail, they need to pursue their training and get no rewards or even a penalty.

After modeling the problem with an MDP, it is possible to apply reinforcement learning algorithms. Reinforcement learning algorithms are classified into two categories: model-free *versus* model-based algorithms. The difference between model-free and model-based approaches lies in the transition probability function P. In model-free RL, the agent does not know the transition probability function P, while in model-based RL, the agent knows P [151, 154]. To summarize, the agent learns the MDP and the environment model in model-based RL, while in model-free RL, the agent interacts with the environment and learns what to do thanks to the accumulated experience [151, 155]. Thus, if along the MDP, there is a model available with the transition probability function P, model-based RL algorithms are well suited. Otherwise, if no model is available or if the model is far too complex to design, model-free RL algorithms are more suited. Due to the dynamics of an IoT network, especially in the field of wireless mobile charging, we will focus on a model-free approach.

Then, model-free RL can be decomposed into two categories: value-based RL algorithms and policy-based algorithms. Policy-based RL algorithms aim to learn the policy function directly [151, 153]. Value-based RL algorithms aim to learn a value function that defines the *quality of being in a given state* s. Being in a good state is tied to the actions that are taken by an agent; how it behaved such that it reached this state. This behavior is the *policy* of the agent. Formally, a policy π maps the states s of the MDP to the actions a the agent can take, written as $\pi(a|s)$. Thus, it is a probability of taking action a if the state is equal to s. Thus, if the agent follows the policy π, the value function is defined as *the expected discounted reward when starting in the state* s *at time* t [151, 156]:

$$V^{\pi}(s) = \mathbb{E}_{\pi}[G_t|S_t = s] \tag{4.2}$$

where $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$ is the discounted reward starting from time step t, $0 \leq \gamma \leq 1$ is a discount factor, and $\mathbb{E}(.)$ is the mathematical expectation. In practice, $\gamma = 0$ or $\gamma = 1$ cases are not considered (a null discount factor implies that the agent will always maximize the immediate reward, a unit discount does not discount all future rewards).

This value function is useful to introduce the analogous action-value function: the Q-function Q. The Q-function gives the expected reward when the agent, for a policy π, takes action a in state s. It is formally defined as:

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}[G_t|S_t = s, A_t = a] \tag{4.3}$$

This Q-function is a central piece in many reinforcement learning algorithms. A smart agent (in our case, the wireless mobile charger) wants to learn the best policy π^* that gives the best discounted rewards [156]. π^* is the policy that maximizes the Q-function, i.e. that gives the optimal action-value function $Q^*(s,a)$.

$$Q^*(s,a) = Q^{\pi^*}(s,a) = \max_{\pi} Q^{\pi}(s,a) \tag{4.4}$$

This function is central to the eponymous algorithm presented in the next subsection.

4.2.3 Q-learning

Q-learning is one of the most famous model-free reinforcement learning algorithms designed by Watkins et al. in 1989 [157]. Q-learning is an *off-policy* RL algorithm. It is said to be off-policy because the agent acts according to a policy different from the one it is learning [151, 155]. In *on-policy* algorithms, the same policy is used during learning and decision-making.

Q-learning is based on a table of state-action pairs with a Q-value associated to each state-action pair. At each time step, an agent that uses Q-learning updates each Q-value according to the update rule described below:

$$\mathcal{Q}(s_t,a_t) \leftarrow \mathcal{Q}(s_t,a_t) + \alpha[r(s_t,a_t) + \gamma \max_{a} \mathcal{Q}(s_{t+1},a) - \mathcal{Q}(s_t,a_t)] \tag{4.5}$$

When using Q-learning, the agent continuously learns a policy π that gives a good future discounted reward R_t. Indeed, the agent wants to maximize the reward it receives over time.

Although Q-learning has been a breakthrough in the field of reinforcement learning, its main drawback is that the more the state space increases, the more costly Q-learning becomes. Furthermore, it only works when the state and action spaces are discrete. Other reinforcement learning algorithms are required if the state and action spaces are continuous.

4.2.4 Deep Q-learning

Deep Q-learning is also one of the most famous deep reinforcement learning algorithms. This approach was proposed by Mnih et al. in 2015 [150]. In Deep Q-learning, a neural network is used to approximate the Q-table. Deep Q-learning is an answer to the curse of dimensionality induced by the Q-table. Indeed, the Q-table is a matrix that can take a lot of memory space if the state space increases. Since the Q-table maps the states and actions to a Q-value $Q(s,a)$, the size of this table is equal to $|S| \times |A|$. If the state space S

is discrete and does not hold too many states, then the $Q-table$ may be used in memory. However, if the state space is continuous (or some variables are continuous), then the state space S can take an infinite amount of state values, and thus, the Q-table would have an infinite size. Unless some special machines can store in their memory infinite-sized tables, it is not the case for the computers of this era. Thus, one of the solutions to tackle this curse of dimensionality is to approximate the Q-Table.

That is the approach considered by Mnih et al. in 2015 [150] in which they used a Convolutional Neural Network (CNN) to approximate the Q-table. They combined the CNN to an experience replay, a memory structure, to remove correlations between multiple successive observations. Combined with this memory structure and the use of a target network, their agent achieved superior results compared to human levels on a broad range of Atari games.

One major advantage of neural networks is that they are good non-linear function approximators [158]. It has been observed that, with just one hidden layer of neurons, it is possible to approximate a non-linear function. Furthermore, thanks to the neural network, it is possible to have continuous state spaces, but the action space is still discrete.

The main interest of Deep Q-learning and other deep reinforcement learning algorithms is to be efficient when the state space increases. Existing deep learning approaches can be combined to RL to tackle different problems. Deep learning algorithms build on neural networks that combine one or multiple hidden layers of neurons. Figure 9 presents a multi-layer perceptron which is one of the most simple neural networks Deep Q-learning can use.

If the reader is interested in reinforcement learning and deep reinforcement learning, the book written by Sutton and Barto is one of the major references in the field [151]. For one of the major applications of deep reinforcement learning, the reader can read the article written by Mnih et al. about Deep Q-networks and Atari games [150] or the more recent success of AlphaGo [159] which is based on Monte-Carlo tree search and policy iteration using a deep neural network.

4.2.5 Applications of RL and DRL in IoT networks

Thanks to the promises of automated and intelligent decision-making, reinforcement learning has got a place of interest in IoT research. Indeed, RL and DRL have many applications, from the physical layer to the application layer [65, 70, 154, 160].

For instance, reinforcement learning is useful for controlling IoT devices. Murad et al. [160] showed that deep reinforcement learning approaches such as policy-gradient

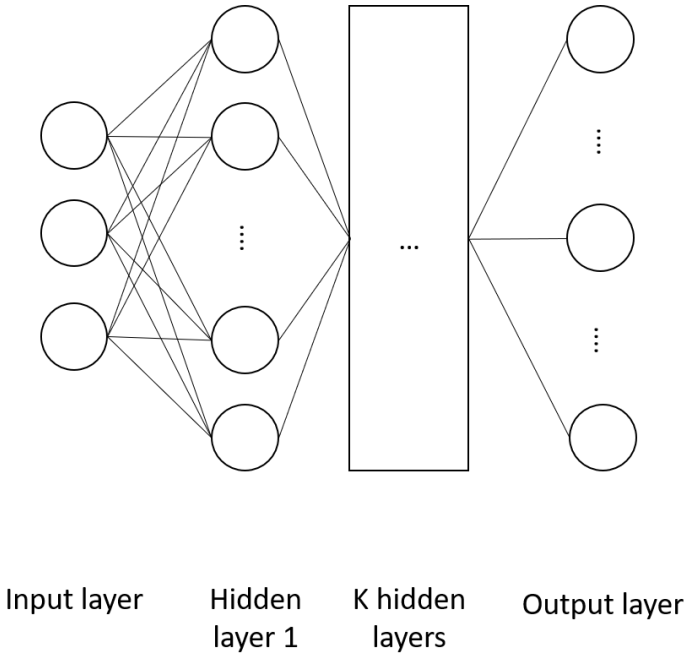


Figure 9: Example of an artificial neural network that is a multi-layer perceptron.

perform well for the control of energy-harvesting IoT devices. Thus, it is a viable choice for long-term control of IoT devices.

DRL is also viable for data collection in IoT networks. Benhamaid et al. proposed two DRL-based path planning strategies [161]. The first one is an off-policy approach based on Deep Q-learning, and the second is an on-policy approach based on Deep SARSA. In their experiments, the Deep Q-learning approach can collect more data but consumes more energy than the Deep SARSA approach. They showed that Deep Sarsa is safer in its decisions than the Deep Q-learning approach since it consumes less energy. Therefore, there are fewer risks that the mobile data collector runs out of energy during the data collection process.

DRL can be applied to computation offloading to improve IoT device lifetime. Min et al. proposed such an approach [138] for devices that can harvest energy from their environment. An IoT device has to offload computational tasks to Mobile Edge Computing (MEC) devices that have more computational power and energy. After proposing an offloading scheme based on Q-learning named RLO, they provided a DRL-based offloading approach called DRLO that uses a CNN as the function approximator. Their DRL-based approach consumes less than the RL-based approach, has a lower communication latency, and drops fewer tasks. However, the DRL-based approach takes more memory space (345 MB) than the RL-based approach (168 MB) and is slower to make decisions (8.3 ms vs. 0.4 ms).

DRL can also be used for the mitigation of threats in SDNs (applied to IoT or traditional networks). Akbari et al. [70] proposed this solution for security management in SDN environments. Their framework, called ATMoS, gives guidelines for implementing RL-based security agents for SDN environments. Observations from the different hosts are used to build the state space, and actions are host migration to virtual networks with stronger or weaker security levels. They studied the performances of ATMOS when the threat is an Advanced Persistent Threat and observed good mitigation results.

The main advantage of RL and DRL is their capability to learn during the whole system lifetime. Furthermore, RL and DRL approaches iteratively give the action for the current time slot in a short time, while optimization-based or constraint programming approaches can take more time to give results.

However, an important drawback of RL and DRL methods is their memory size. As already mentioned, the contributions of Min et al. [138] have a big memory footprint: the DRL-based offloading solution used 345 MB of memory, while the RL-based offloading solution used 168 MB of memory. This is a huge amount of memory, unbearable for IoT devices that are either too small or heavily energy-constrained.

4.2.6 Applications of RL and DRL to wireless charging

Although reinforcement learning and deep reinforcement learning have been considered in robotics and other industrial domains for many years, their application in IoT networks is quite recent. Furthermore, the application of RL and DRL for the design of wireless charging strategies is even more recent.

The first proposed work that considered reinforcement learning for wireless charging path planning was carried out by Wei et al. for Wireless Rechargeable Sensor Networks (WRSNs) [162]. Their strategy, called Charging Strategy based on RL (CSRL), based on the simulated annealing algorithm, considers the remaining energy, the position of devices, their power consumption, and if they were already visited in a charging round. Compared to a greedy algorithm (that always selects the node with the lowest energy level), with a mean lifetime of 5.5×10^4, CSRL achieved a mean lifetime of 9.74×10^4, with a lower mean driving distance.

A second work of interest in this field was led by Van Quan et al. [163]. They studied the problem of on-demand charging in which devices issue recharge requests. The goal is to find the best charging path according to the different demands and device status. They provided a Q-learning-based on-demand wireless charging scheme for rechargeable WSNs with a multi-node charging scheme. Their Q-learning algorithm considers an estimated charging time, and it aims to maximize the network coverage after the recharge at each

charging location.

Yang et al. [148] investigated the use of Actor-Critic Reinforcement Learning (ACRL) for rechargeable WSNs for dynamic charging. They considered that the energy consumption rate of devices may vary due to uncertainties in network operations. Devices have energy demands which are the amount of energy needed to be over a certain threshold. The actor-critic method is implemented with Gated Recurrent Units (GRUs), which are an evolution of recurrent neural networks. In their simulations, their scheme achieved better results than prior works, especially in networks with a high number of sensors.

An interesting research work was carried out by Cao et al. [164]. They proposed an on-demand charging scheme based on Deep Reinforcement Learning (DRL) for rechargeable WSNs. During their research, they modeled the charging demand of each device as time windows and used this model to plan the trajectory of MC to minimize the number of dead nodes and the distance traveled by the MC. Compared to a random approach and a heuristic (named NJNP), their approach achieved a lower number of dead devices and higher rewards.

Besides, Bui et al. provided an adaptive charging strategy for rechargeable WSNs [165]. Their strategy is built on DRL to determine what device to charge next according to a set of static and dynamic parameters. Static parameters are parameters that do not vary, while dynamic parameters change over time (such as the remaining energy, the energy consumption rate, etc.). Their strategy is designed to answer the problem of on-charging strategies that rely on charging requests of field devices. Indeed, charging requests are sent when the remaining energy of the devices is below a threshold. Thus, the main difficulty is to choose a good threshold to not overwhelm the MC with charging requests. The model merges attention and pointing mechanisms alongside a multi-layer perceptron to determine the action to do: charge a device or go back to a charging station. Their experiments outlined that their agent can increase network lifetime compared to non-learning approaches.

Although recent works based on RL or DRL improve greatly network lifetime compared to non-learning-based approaches, in Table 7, we outline the main elements of the state space of each related work and show that they are not context-aware. Van Quan et al. [163] did not consider context information, and future energy consumption rate is not considered. Moreover, for large networks, using a Q-table is counter-effective because the size taken by the table may explode. Yang et al. [148] explored an alternative based on actor-critic RL for WSNs. It exposed good results, but they did not consider the context and probable future energy of devices to charge them in advance, but only their current energy needs. Cao et al. [164] used a model to consider the charging demands

of devices, but it is linked to a threshold. Nevertheless, they considered that the mobile charger has a limited battery. Moreover, the energy consumption rate is considered as known, while in reality, the consumption may fluctuate according to ongoing events.

Reference	RL or DRL	Remarks on the state space
Wei et al. [162]	RL	State space only considers remaining energy of the devices, distance to the charger and if the device was already visited. It is not context-aware.
Van Quan et al. [163]	RL	The state space only considers charging locations determined from charging requests issued by the devices. It is not context-aware.
Yang et al. [148]	DRL	The state space only considers the coordinates of each device and their energy demand to reach a threshold. It is not context-aware.
Cao et al. [164]	DRL	The state space only considers the remaining energy of the charger and the devices, the distances to the charger, and if the time window of the device to charge it is open. Context awareness is limited (can the device be recharged or not at this moment).
Bui et al. [165]	DRL	The state space considers for the device-related information its remaining energy and its consumption rate, but it is estimated from previous remaining energy levels and not the current context. It is not context-aware.

Table 7: Description of the state space of existing works and their limitations regarding context-awareness.

Recent related works in wireless mobile charging are not context-aware, except the research work conducted by Cao et al. to a limited extent [164]. All these reasons motivated us to propose a new approach considering the contexts, past and future, for the design of an intelligent wireless mobile charging scheme. Context modeling will be carried out by a Context-reasoning module (CRM) at the base station (BS). This CRM will give the charger information related to future context to determine the device to charge next thanks to an variable called *modified importance*.

4.3 Our solution: context-aware wireless charging

In the previous sections, we introduced different useful concepts to tackle and model the problem of context-aware charging. We start this section by explaining a general idea of our solution. Then, we present the considered system model. Besides, we detail how

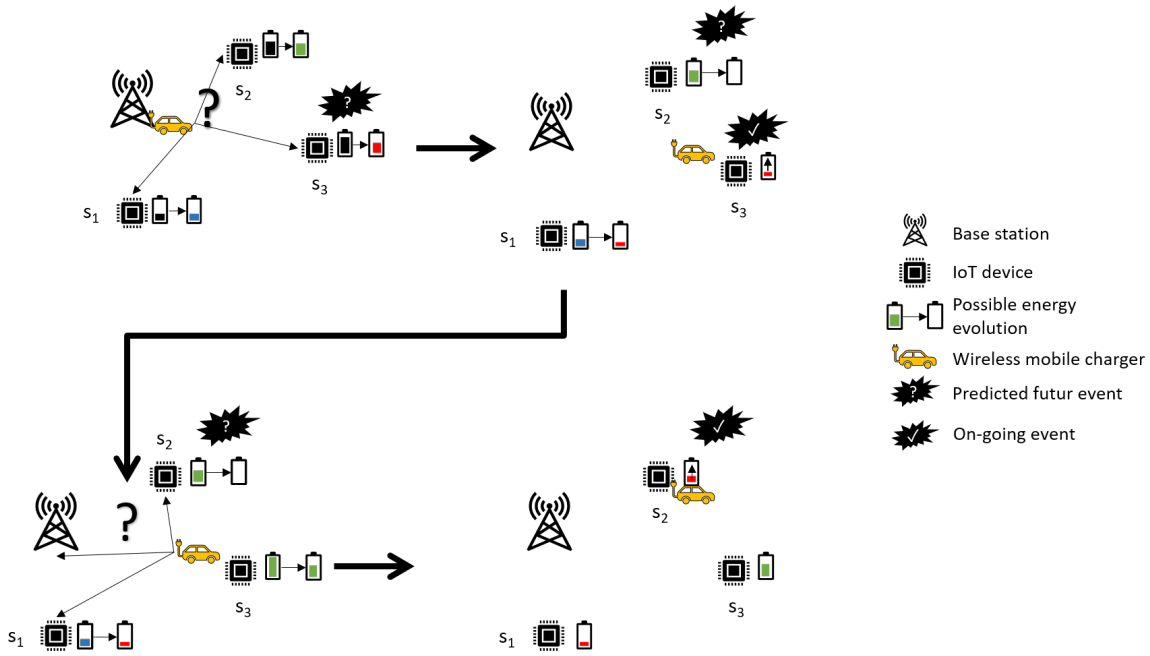


Figure 10: Model of the network and the wireless mobile charger.

context-awareness can be useful for wireless charging trajectory planning. Finally, we present the context-aware charging strategy for IoT networks.

4.3.1 General overview of the solution

To give the reader a general idea of the solution, we present in Fig. 10 the proposed solution with a sole base station, three smart devices s_1, s_2, s_3, and a Mobile Charger (MC).

Classical charging schemes choose devices with the lowest remaining energy and/or with the highest energy consumption rate, and build the charging path at once. However, the main limitation of only considering energy consumption rate and remaining energy is that these values do not reflect the possible future energy consumption of the devices, which may have a sudden increase in their energy consumption rate. Devices with a high energy consumption rate may, in the next time slot, have a low energy consumption rate, while for other devices with a low energy consumption rate, this rate may explode due to a future event that may happen.

In our example, at time t, the MC has to choose which node should be recharged. A classical charging scheme would choose s_1 since its remaining energy is the lowest one. However, in this time slot, an event at node s_3 is likely to happen, with a huge energy consumption induced by the event, while node s_1 will not consume a lot of energy. With our scheme, the MC will charge s_3 to cover the energy expenditure during the time slot t. At time $t+1$, some context information alerts the network that an event is likely to happen

at s_2, with a possible node death. Thus, the MC should drive towards s_2 to preventively charge it. Node s_1 will have a low energy consumption, as in the previous time slot.

Our scheme aims to cover these cases, as classical schemes, and even intelligent mobile charging schemes, lack to consider. Related RL-based or DRL-based such as [148, 164] does not consider future impacts of events in the environment.

Knowing possible events in the environment and their impact on the energy consumption of IoT devices will improve the efficiency of wireless mobile charging schemes. In the next subsection, we present the system model.

4.3.2 System model

In this subsection, we detail the general system model of our scheme, which is also depicted in Fig. 10. The considered IoT network is made of different devices d_i that belong to different classes. A device d_i belongs to a class c_k, which defines its battery size B_{c_k}. The set of all classes is written as $\mathcal{C}$. Hence, this network is a heterogeneous IoT network. These devices are placed into a field of size $L * L$ meters. There is a Base Station (BS) at coordinates $(0, 0)$ and multiple Energy Access Points (EAPs) located at fixed coordinates in the network. We consider a unique Wireless Mobile Charger (WMC) that travels in the field to charge the devices. The WMC has a battery capacity of B_{MC}, coming from the BS or an EAP. EAPs will act as safe points for the WMC. We consider a battery-limited WMC, similar to some previous works [163, 164]. We also consider that the WMC may charge only one device at a given time. It is important to note that, in the literature, there are related works that consider the WMC can charge multiple devices at once [135, 163, 166].

Each $d_i \in D$ has an importance level imp_i, denoting its importance in the network. A high value of imp_i indicates that d_i has critical tasks in the network. In this case, if d_i dies, the network (or running application) may be critically impacted (e.g. if an actuator dies during a critical process, the process will fail and it may have disastrous consequences). A low value of imp_i indicates that the tasks of d_i are not important with regard to the application/ongoing process. This importance value may be fixed (i.e. defined by operators during network deployment) or may change over time (e.g. a function of different network parameters). We consider in this chapter that imp_i may vary over time due to varying contexts. Thus, we write imp_i as imp_i^t. This importance imp_i^t does not have a general formula since it is application-dependent. For the sake of clarity, we summarize the main notations used in this chapter in Table 8.

In the next subsection, we detail how we can enable context-awareness for wireless mobile charging.

Notation	Definition
D	Set of smart devices
$\mathcal{A}$	Set of Energy Access Points
E_i^t	Remaining energy of device d_i at time t
imp_i^t	Importance level of device d_i at time t
$\widetilde{imp_i}^{t+1}$	Modified importance level of device d_i at time $t+1$
$\mathcal{C}$	Set of device classes
B_{c_k}	Battery capacity of devices in class c_k
$\mathcal{E}$	Set of events
$\mathcal{H}$	History of events
e^t	Event happening at time t
$crit_{e_i}$	Criticality of event e_i
$\tilde{e}^{t+1}$	Predicted event at time $t+1$
$\tilde{E}_{e^{t+1}}$	Estimated energy cost of the predicted event at time $t+1$
$E^t_{activities}$	Energy spent in activities outside of event handling at time t
N^t_{req}	Number of recharge requests at time t
N^t_{alive}	Number of alive nodes at time t
N^t_{dead}	Number of dead nodes at time t
B_{MC}	Battery capacity of the MC
E^t_{MC}	Remaining energy of the MC at time t
E_{move}	Energy consumption of the movement of the MC
η_{c_k}	Charging rate of MC towards class of devices c_k

Table 8: List of the main notations used in the chapter.

4.3.3 Enabling context-awareness for wireless mobile charging

Context-awareness is the promise of adapting any service to the needs of a user [167] thanks to information regarding the environment, the moment, the place, etc. One of the first definitions on what is context has been proposed by Abowd et al. [104]: '*any information that can be used to characterize the situation of an entity, where an entity can be a person, place, or physical or computational object*'. They also proposed a definition for context-awareness in which the context is used to take a relevant decision [104, 168].

Context and context-awareness are concepts that have plenty of applications, from networks to user's well-being. For instance, context-awareness can be useful in smart museums to provide visitors with an enhanced experience and real-time information on art pieces [169]. As presented in Chapter 3, context-awareness can be used to design efficient security solutions for IoT networks. Context-awareness enables energy-efficiency while protecting the IoT network when the current context needs strong security levels. Thus, context-awareness is very useful to make decisions. Below, we present related works on context-awareness for decision making, and then, how we can model varying contexts for wireless charging.

Context awareness and decision-making: related works

In the context of wireless charging, knowing the context may improve the decision-making process for the choice of the device to charge. Indeed, modeling the context and predicting it is of uttermost importance to model the interactions between the environment and the IoT network since it improves the decision-making process in which context-awareness is embedded [104, 170, 171]. Thus, context-awareness can improve decision-making in a wireless mobile charging scenario.

Many works of interest describe the usefulness of considering context-awareness for decision-making. An interesting work on context-awareness was done by Rodrigues et al. [172]. They provided an integration of contextual security information for their architecture called HAMSTER for unmanned aerial vehicle networks. A security context is separated into three components: external security context, mission context, and internal security context. HAMSTER uses the three components to compute a perceived security index (PSI) and uses it to activate security mechanisms. This PSI can change during the mission, enabling the UAV to improve its security level or determining if the mission should stop. Thus, contextual information is thus useful to determine the security solutions to use before, during, and after a UAV mission.

A work of interest we found for context prediction was done by Ding et al. for industrial environments such as mining [170]. They applied context-prediction for the prediction of power consumption since power is expensive. Thus, if it is possible to predict the context and reduce the number of active machinery, then energy savings are made. To enable context-prediction, the proposed system merges data acquisition (noted as sensing), context identification (noted as computing), and prediction along with adaptation (noted as adaptation).

A second work of interest for context-aware decision-making was done by Meurer et al. [171]. They studied the problem of context-aware control for smart home and company environments. Indeed, if there are no users in a room, no appliances or lights should be active. They presented a context-aware decision engine built on neural networks and fuzzy controllers. The neural network is used to predict the future control to apply to the devices, given the current context (that comes from sensors, actuators, etc.). Then, the fuzzy controllers may modify the control command using the user's preferences and pre-defined rules. The user can give feedback, such as undoing the control done by the neural network and the fuzzy controller. They demonstrated during experiments that context-aware control can reduce the power consumption of the environment.

These research works demonstrated that the use of context-awareness improves the control or the decision-making process. The problem of context-aware wireless charging

was not tackled in the literature before. Thus, it is important to determine how should context-awareness be embedded into the wireless charging decision process. One of the main challenges is to design the interaction between a context management module and the wireless charging module, that we present next.

Context modeling for wireless charging

To have a context-aware wireless mobile charging scheme, the first building block to consider is context management. One may manage the different contexts by collecting information about the contexts, modeling them, reasoning about them, and taking action. IoT environments generate a lot of data. Thus, there are records of the residual energy of each device and what kinds of events happened, and how much it lasted.

We do not detail which kind of context we consider here, as it may be contextual information on security [172], on the environment, of an ongoing process, etc. The system proposed in [171] considered the context to have an adaptive control for smart home environments. Ding et al. considered power consumption as the context to predict the future context [170]. If context-awareness may improve the control of smart homes or mining environments, then having an adaptive WMC scheme is possible. Context modeling and reasoning are managed by the Context Reasoning Module (CRM) presented in Fig. 11. Thanks to the CRM, the base station may estimate the future events that are likely to happen and what devices will manage these events. It will then send to the WMC the information related to future events.

Formally, we define an event as: $e_i =< crit_{e_i}, \mathbf{C} >$ where $crit_{e_i}$ is the criticality of event e_i and $\mathbf{C}$ represents the classes of devices it targets ($\mathbf{C} \subseteq \mathcal{C}$). The set of events is written as $\mathcal{E}$. With multiple events happening in the network over time, an entry $\mathcal{H}_i$ in the history of events is defined as:

$$\mathcal{H}_i \ = \ \{t_{start_i}, t_{dur_i}, E_{evol_i}, e_{e_i}\} \tag{4.6}$$

where t_{start} is the beginning time of the event, t_{dur} its duration, E_{evol} is the matrix of the remaining energy at t_{start} and $t_{start} + t_{dur}$ (the end of the event) for each device, and e_e is the event that happened. The aggregate of all entries $\mathcal{H}_i$ is the history of the events $\mathcal{H}$. It is important to note that multiple events may happen at the same time, increasing the need for energy. However, for the sake of simplicity, we consider that only one event may happen at a given time t.

The CRM takes as input the current context (e.g., sensor values) and finds similarities with past events to output the probable future events and the concerned device classes.

If one sees $e^1, e^2, \ldots, e^i, e^{i+1}, \ldots, e^t$ as a sequence of events up to time t, then, the

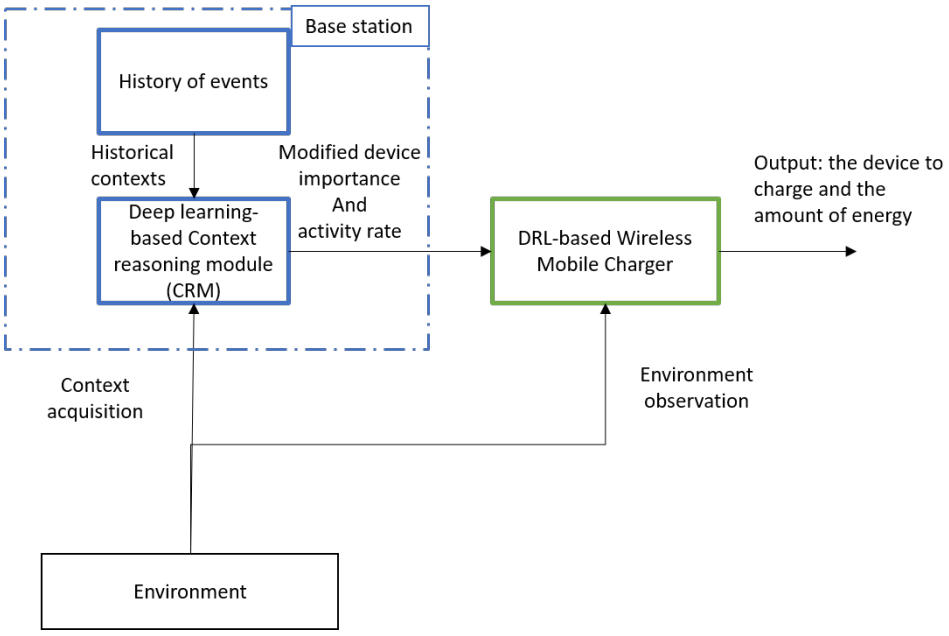


Figure 11: General model of the scheme. Two modules are considered: a deep learning-based context reasoning module (blue) and a DRL-based Wireless Mobile Charger (green).

next events $e^{t+1}, \ldots, e^{t+\tau}$ may be predicted by the CRM, with τ being the time horizon used for the prediction. The predicted events of $e^{t+1}, \ldots, e^{t+\tau}$ are denoted as $\tilde{e}^{t+1}, \ldots, \tilde{e}^{t+\tau}$.

There exist different deep learning methods that take a sequence of k things (words, values) in input to give outputs for the $k+1, \ldots, k+\tau$ future outputs such as time series models, Recurrent Neural Networks and derivatives (which are deep neural networks: GRU, LSTM-RNN, etc.). Since we aim to provide a general approach of the CRM, we do not detail what algorithm we should choose, and we present the general logic behind the process of the CRM.

For each event $e \in \mathcal{E}$, there is an estimated energy cost (energy needed to process the event). Thus, when the CRM outputs the sequence $\tilde{e}^{t+1}, \ldots, \tilde{e}^{t+\tau}$, the energy cost of each predicted event is $\tilde{E}_{e^{t+1}}, \ldots, \tilde{E}_{e^{t+\tau}}$.

The CRM is also able to output a matrix $\mathbf{P}$ of size $N * \tau$ reporting the possible activity rate ρ_i^{t+k}, $k \in 1 \ldots \tau$ of each device d_i at time $t+1, \ldots, t+\tau$. This activity rate is a percentage of time the device will be active for the next time step. When its value is near zero, it means that the device will be in a sleep state most of the time. On the contrary, when it is near one, it means that the device will be in an active state most of the time. This activity rate is application-dependent.

This predicted activity rate might impact the future importance level imp_i of each

device d_i. Thus, for each device d_i, we define a modified importance level $\widetilde{imp}_i^{t+1}$ as:

$$\widetilde{imp}_i^{t+1} = imp_i^t + \rho_i^{t+1} crit_{\tilde{e}^{t+1}} + \frac{1}{\max(0, E_i^t - \rho_i^{t+1} \tilde{E}_{e^{t+1}})} \tag{4.7}$$

This modified importance level will provide the MC with information on the importance of each device in the future time slot. It also considers the potential impact of the energy cost of the predicted event. If the event is likely to happen near the device d_i and the remaining energy of d_i is near zero, then $\widetilde{imp}_i^{t+1}$ tends to the infinity, indicating that device d_i should be prioritized for charging. If no event is likely to happen, then $\widetilde{imp}_i^{t+1}$ may be set to the initial value $imp_i^{t_0}$. This modified importance $\widetilde{imp}_i^{t+1}$ may also be generalized to consider the values up to the time $t + \tau$.

Estimating such importance level and what may be the future events are difficult tasks. We considered that the CRM was located at the BS. However, a better place for this module would be either a Fog device or the Cloud for the computational power of these devices. We present the general guidelines of the context-reasoning module in Algorithm 1. The CRM takes as an input the observed context, historical events, and the time horizon τ. The first step is to determine the most probable events to happen in the future (Line 1). Then, using this prediction of the future events, the activity rate is predicted (Line 2). Finally, for each device and for each future time slot, the CRM computes the modified importance (Line 4) and sends it to the WMC (Line 6). The time horizon τ is fixed by the administrators of the CRM or application-dependent.

Algorithm 1 Pseudo-code of the Context-reasoning module

Input: Observed context (e.g. sensor values, user input, etc.), historical events, time horizon τ

Output: The modified importance of each device d_i $\widetilde{imp}_i$ from $t+1$ to $t+\tau$

1: From the past events and the observed context, determine the most probable events in the future $\tilde{e}^{t+1}, \ldots, \tilde{e}^{t+\tau}$
2: Compute the matrix $\mathbf{P}$ of the different activity rates of each device ρ_i^{t+k}, $k \in 1 \ldots \tau$
3: **for** $k = 1, .., \tau$ **do**
4: Compute the modified importance $\widetilde{imp}_i^t$ of each device d_i according to Eq. 4.7
5: **end for**
6: Send the modified importance values to the WMC. If $\tau \neq 1$, the WMC will store the values on-board and use them for its future decisions

This architecture is somewhat similar to the one considered by [171]. Indeed, Meurer et al. used a neural network to determine the control to apply to the devices according to the current context. Their neural network is trained on past user commands, prior knowledge, room, and user room data. However, the neural network part of Meuer et al.

would predict the future control to apply to the devices, while in our work, we consider that the CRM only predicts the state of the devices in the future, thus, their modified importance $\widetilde{imp_i}^{t+1}$. Then, this modified importance for each device will have an important place in the decision-making process.

4.3.4 Establishing an intelligent charging strategy with deep reinforcement learning

With regard to the existing works and the previous section, we investigate in this section the use of context-awareness for a wireless mobile charging strategy, depicted in Fig. 11 with the Wireless Mobile Charger module.

As explained in the overview of our solution, we aim to preventively charge devices instead of always responding to recharge requests, even if it is a possibility. In this approach, the proposed strategy is not an on-demand strategy nor a periodic charging strategy but an adaptive strategy. Thus, we have to design an intelligent scheme able to consider the modified importance of devices, along with their remaining energy and their distance to the MC, with a minimum number of queued recharge requests N_{req}.

First of all, we need to determine the energy spent by the devices and the WMC. On the device side of d_i, the amount of energy left at any time t is:

$$E_i^t = E_{d_i}^{t-1} - E_{e_e}^t - E_{activities}^t \tag{4.8}$$

$E_{activities}^t$ is the energy spent in normal tasks (sensing, computing), which we assume is known. $E_{e_e}^t$ is the energy cost of handling event e_e at time t, which is known when it happens. For each device, the following constraint always holds:

$$\forall t \in T,\ E_i^t\ >\ 0 \tag{4.9}$$

Furthermore, the estimated energy at the future time slot $\tilde{E}_{d_i}^{t+1}$ for device d_i is:

$$\tilde{E}_{d_i}^{t+1} = E_{d_i}^t - \tilde{E}_{tasks}^{t+1} \tag{4.10}$$

where $\tilde{E}_{tasks}^{t+1} = \tilde{E}_{activities}^{t+1} + \tilde{E}_{e_e}^{t+1}$ may be estimated thanks to the output of the CRM (described in the previous subsection).

In a similar manner, if the time horizon τ is higher than one, then:

$$\tilde{E}_{d_i}^{t+\tau} = E_{d_i}^t - \sum_{j=1}^{\tau} \tilde{E}_{tasks}^{t+j} \tag{4.11}$$

Thus, depending on the time horizon τ chosen, the goal is to find what node should be recharged first to always respect the constraint in Eq. 4.9. However, the amount of energy transferred to a device should not exceed its maximum capacity to avoid energy wastage.

The WMC may send an amount of energy E_{MC,d_i} to d_i, up to a maximum amount E_{MCmax}, with a charging rate η_{c_k} tied to each device class c_k. The MC moves at a constant speed of v m/s and consumes E_{move} for each second it moves, for a total movement duration of t_{travel}. However, the MC must consider its own remaining energy E^t_{MC}, defined by:

$$E^t_{MC} = E^{t-1}_{MC} - E_{MC,d_i} - t_{travel}E_{move} \tag{4.12}$$

If the wireless mobile charger does not have enough remaining energy to charge devices, it should go back to an EAP or the BS before running out of energy. If it runs out of energy before reaching a safe point, i.e. $E^t_{MC} = 0$, then the scheme has to give a penalty $r_{deathMC}$ to the WMC.

If a device d_i does not have enough energy to fulfill the tasks due to the event e^t, i.e. $E^t_i \leq \tilde{E}_{e^t}$, then the task fails. The scheme receives a penalty r_{fail} for this failure.

If the energy of the i^{th} device reaches 0, the device dies, and the IoT network has to re-organize itself to recover from the node death. It will incur high energy and computational costs, and that should be avoided. For each node death, a penalty of $r_{deathDEV}$ is given to the WMC.

Based on the previous details, we formally define our problem as a Markov Decision Process (MDP) with the 4-tuple (S, A, R, P) where:

S is the set of states. It considers the remaining energy E^t_i of each device, the remaining energy E^t_{MC} of the mobile charger, the modified importance level $\widetilde{imp}^t_i$, the Euclidean distance between the devices d_i and the mobile charger $d(d_i, MC)$.

A is the set of all actions where $a_t = [d_i, E_{MC,d_i}],\ i \in 1 \ldots |D| + |\mathcal{A}| + 1$, $a_t \in A$, with the device d_i and the amount of energy sent to d_i. Here, d_i may be a device, an energy access point, or the base station. If the WMC chooses to go to the BS or an Energy Access Point (EAP), the amount of energy sent to the device E_{MC,d_i} becomes zero. Otherwise, E_{MC,d_i} is the amount of energy until its battery is full.

R is the reward function, which maps the states and the actions to numerical values. Especially we have:

- For a successful event handling by node d_i, the MC receives a reward of r_{succ}. For nodes staying alive (and did not have to handle an event), the MC receives a reward

of r_{alive}.

- If d_i could not handle the event e^t, the MC receives a penalty of r_{fail}.
- If the MC dies of energy exhaustion, it receives a penalty of $r_{deathMC}$
- If a device d_i dies of energy exhaustion in the round, the MC receives a penalty of $r_{deathDEV}$.
- Otherwise, it receives no reward.

Penalties are negative values whereas the rewards r_{succ} and r_{alive} are positive. Thus, the immediate reward received $r(s_t, a_t)$ by the MC due to the action a_t is:

$$\begin{aligned} r(s_t, a_t) = &\sum_{i=1}^{n}(h_i r_{succ} + (1 - h_i) r_{fail}) \\ &+ r_{deathMC} + N^t_{alive} r_{alive} \\ &+ \beta(N^t_{dead} r_{deathDEV} - N^t_{req}) \end{aligned} \tag{4.13}$$

With h_i being a binary variable indicating if d_i has successfully handled event e^t or not, β a penalty factor for the number of recharge request messages N^t_{req} and the penalty tied to the dead nodes. For the sake of clarity, we will write $r(s_t, a_t)$ as r_t.

The future discounted reward, starting from time slot t and considering a discount factor γ, is defined as:

$$R_t = \sum_{i=t}^{T} \gamma^{i-t} r(s_i, a_i) \tag{4.14}$$

P is the probability of going from state s_t to state s_{t+1} given that the MC did the action a_t. In this chapter, the transition probability from s_t to s_{t+1} is unknown. Because the transition probability P is unknown, we consider a free-model setting. Thus, using Q-learning is suited to the problem. The MC aims to learn a policy π that gives a good future discounted reward R_t.

The agent, the MC, has to learn what actions it must take to maximize the device lifetime with regard to our constraints. Thus it has to learn a policy $\pi(a|s)$ mapping the states s to actions a. However, the MC seeks to learn the optimal policy π^*, which maximizes the expected return of R_t, i.e. $\pi^* = arg\max_\pi \mathbb{E}(R_t|\pi)$. As presented in

subsection 4.2.3, Q-learning relies on updating the Q-function, recalled in Eq. 4.15.

$$\mathcal{Q}(s_t, a_t) \leftarrow \mathcal{Q}(s_t, a_t) + \alpha[r_t + \gamma \max_a \mathcal{Q}(s_{t+1}, a) - \mathcal{Q}(s_t, a_t)] \tag{4.15}$$

The MC has to find an approximation of the optimal Q-function which follows the optimal policy π^*, denoted as $\mathcal{Q}^{\pi^*}(s_t, a_t)$. Since our state space has many parameters, implementing a Q-table to find the optimal policy π^* would be too memory-consuming. Thus, to solve this problem, we consider the use of Deep Reinforcement Learning (DRL) [150, 151] to approximate the Q-function, as presented in subsection 4.2.4. Deep neural networks are good non-linear function approximators (as explained in Section 4.2.4, thus, useful to approximate the values of the $\mathcal{Q}$-function. We call these networks Deep Q-Networks (DQN). To further improve the efficiency of DQNs, we use the experience replay structure introduced by Mnih et al. [150]. This experience replay is a memory pool $\mathcal{M}$ that stores past experience, i.e. it stores the past state, the corresponding action taken, the received reward, and the new state induced by the action. A typical entry in $\mathcal{M}$ is:

$$m_t = (s_t, a_t, r_t, s_{t+1})$$

Keeping a memory pool will allow the agent to draw entries, the sampling process, and limit the impact of the correlation of successive states. When the sampling process happens, there is a batch of experiences sampled from $\mathcal{M}$ as (s, a, r, s').

A second network $\tilde{Q}$-network, called target network, may be useful to stabilize the learning process [150, 164]. This target network also approximates a $\tilde{Q}$-function, called target function and denoted as $\tilde{Q}$. During the training process, the parameters of the target network will be updated thanks to the parameters of the Q-network. This target network is necessary because it reduces the risk that the policy the agent is training diverges.

During the training process, the DQN trains its weights θ through the minimization of a loss function $L(\theta)$. The error function $f(\theta)$ considered is the Huber loss that is more resilient to outliers compared to the mean squared error [173]. When the prediction error is under a certain margin, Huber loss is similar to the mean squared error (also called the L2 loss). Otherwise, it is similar to the mean absolute error (also called L1 loss).

$$f(\theta_k) = \begin{cases} \frac{1}{2}(Tar_t - Q(s, a, \theta_k))^2 & \text{if } |Tar_t - Q(s, a, \theta_k)| < \delta \\ \delta(|Tar_t - Q(s, a, \theta_k)| - \frac{1}{2}\delta) & \text{otherwise} \end{cases} \tag{4.16}$$

where $Tar_t = r_t + \gamma \max_{a'} Q(s', a', \tilde{\theta}_k)$, $\tilde{\theta}_k$ are the parameters of the target network $\tilde{Q}$ at timestep k. δ is equal to 1. Then, the loss function $L(\theta)$ is defined as:

$$L(\theta) = \mathbb{E}(f(\theta)) \tag{4.17}$$

Then, the minimization of the loss function is done thanks to a gradient descent method. Finally, the new weights θ of the Q-network are used to update the weights $\tilde{\theta}$ of the target network $\tilde{Q}$. This update of the target network is done every λ time steps, reducing the risks of having a divergent policy [150].

Furthermore, for the whole training process, we consider an ε-greedy policy to explore and avoid local extrema. It means that with probability ε, the charger will take a random action with probability ε and the best action with probability $1 - \varepsilon$. This approach is useful to make the WMC explore the different spaces and possible actions [150, 161, 164].

Based on all the previous details, we present in Algorithm 2 the pseudo-code of the context-aware Wireless Mobile Charger based on Deep Reinforcement Learning. Lines 2 to 6 are the state construction step. It is important to note that, to get the modified importance level, the charger sends a request to the CRM, and then receives the different values. Lines 7 and 8 are the phase of determining the action. Then, after the action was chosen thanks to the ε-greedy policy, the WMC applies the action to the environment, the state evolves from s_t to s_{t+1}, and the WMC receives the corresponding reward $r(s_t, a_t)$ (Line 9). The memory pool is then updated with the entry $(s_t, a_t, r(s_t, a_t), s_{t+1})$ (Line 10). Finally, the parameters θ of the Q-network are updated after minimizing the loss function $L(\theta)$ (Lines 11 and 12), while the parameters of the target network $\tilde{Q}-$network are updated every λ steps (Line 13).

Algorithm 2 Pseudo-code of the context-aware energy management scheme on the Wireless Mobile Charger side

for each time step t **do**
  Retrieve the current remaining energy E_i^t of each device d_i
  Retrieve the remaining energy of the Wireless Mobile Charger E_{MC}^t
  Retrieve the modified importance level $\widetilde{imp_i}^t$ of each device d_i from the context-reasoning module.
  Compute the distance $dist_i^t$ between the mobile charger and each device d_i.
  Build the state s_t with E_{MC}^t, E_i^t, $\widetilde{imp_i}^t$, and $d(d_i, MC)$, for $i = 1, \ldots, n$
  Feed the Q-Network of the agent with the state s_t, the parameters θ
  Take action a_t w.r.t. the $\varepsilon-$greedy policy (it is the device to charge or an EAP, E_{MC,d_i} is either 0 or energy to full battery)
  Receive reward $r(s_t, a_t)$ and process to the state s_{t+1}
  Update the memory pool $\mathcal{M}$ with the tuple $(s_t, a_t, r(s_t, a_t), s_{t+1})$
  Use the target Network $\tilde{Q}$ with a sample batch from the memory pool $\mathcal{M}$ to minimize the loss function $\mathcal{L}$
  Update the Q-network thanks to the minimization of $\mathcal{L}$ described in Eq. 4.16 and Eq. 4.17
  Update the target Network $\tilde{Q}$ every λ steps with the trained weights θ of the Q-Network.
end for

4.3.5 Discussion

The proposed scheme considers the context to preventively charge IoT devices. We believe that this approach is suited to IoT environments due to the large amount of data they generate. Context-awareness has been successfully applied for smart home environments [171] or operation in hostile environments [172]. It is also possible to consider these pieces of data to make real-time decisions. This is the main goal of our scheme: considering device data plus residual energy information to build context information to decide on the quantity of energy that the MC should transfer in the future time in a preventive manner. The event prediction is, in our work, a modification of the importance value imp_i plus a set of possible activity rates that are given to the MC, which then decides on what device to recharge. Moreover, we did not tackle the case when multiple events happened at the same time.

However, the efficiency of our scheme may depend on the efficiency of the prediction of future events. If the predictions are always false, then the MC may go towards a device that may not need preventive charging. In that case, devices that simply need energy should be prioritized. Furthermore, depending on the complexity of the CRM, predicting future events may induce some latency between the moment the CRM receives the request for the modified importance of each device and the moment it sends back the values to

the WMC.

We also considered the use of the Euclidean distance. This distance is useful for large environments with negligible obstacles and within a plane. However, for urban or industrial applications, the Manhattan distance may be more useful. For networks deployed in mountains or environments with multiple height levels, the use of the geodesic distance, which is the distance between two points on a surface, may be more precise. Another approach to compute the distance is to determine the shortest path to the device, given the obstacles, thanks to the A^* algorithm for instance, and then, compute the length of this path.

Existing works on wireless charging did not focus on the impact of tasks and events in the environment. Context-awareness may improve the charging process by providing the WMC an indicator of the current importance and future importance level for each device.

4.4 Conclusion

Existing research works did not consider context awareness to design a wireless charging strategy for IoT networks. They considered quantitative variables such as remaining energy or energy consumption rate and the distance, but the impact of varying contexts was not studied. Therefore, we proposed in this chapter a new energy management approach based on wireless mobile charging that uses context awareness. The proposed solution uses a Context-Reasoning Module (CRM) that may leverage deep learning algorithms to predict future events and the most important devices to tackle these future events. This CRM communicates with a wireless mobile charger that is in charge of choosing the next device to charge given their remaining energy, their distance to the charger, their modified importance in the future, and the remaining energy of the charger. We modeled the problem of context-aware charging with a Markov Decision Process (MDP) and used Deep Reinforcement Learning (DRL) to enable intelligent decisions on the wireless charger side.

This chapter creates the ground for the next chapter which considers threat awareness to determine a charging path. The advantage of reinforcement learning and deep reinforcement learning approaches lies in their adaptability to uncertainties, as opposed to offline approaches. Given that a threat may be random, RL and DRL approaches may be able to solve the problem of threat-aware charging.

Chapter 5

Threat awareness for wireless charging in IoT networks

IoT networks are vulnerable to varying threats. During some periods, attackers may not be interested in data produced by the devices whereas, during mission-critical periods, attackers may target critical devices to steal data or shutdown them. Adaptive security solutions are useful for mitigating threats. However, an adaptive security solution induces variable energy consumption among IoT devices: some of them may consume a lot of energy because they have to use the highest security level, whereas some devices will only use the lowest security level, thus, consuming less energy than other devices for some periods of time. Thus, some devices will run out of energy faster than other devices. As a result, some devices will not be able to secure their communications, leading to energy exhaustion and thus, device failure.

IoT security solutions greatly impact the energy consumption of IoT devices, thus, reducing device and network lifetime. In Chapter 4, we studied the problem of context-aware charging in IoT networks. This approach paved the way to threat-aware charging that was not tackled in the literature. If a charger can identify the device that may need energy for its future security needs, the charger should adapt its trajectory to charge this device. This approach is more flexible and adaptive than static approaches considering only the residual energy and the distance. However, the charger should not neglect devices that are low on energy but that will not consume a lot of energy in the near future. In this chapter, we give an answer to the following research question: 'With knowledge of the current energy status of devices, the distance from the wireless charger to the devices, and the current threat level, how may a mobile charger determine the device it should charge next?' This approach has the potential to reduce the impact of security solutions on the lifetime of IoT devices, thereby increasing the overall network lifetime. During our study, we found out that no research work studied the problem of threat-aware charging, and thus, we are the first to propose a solution to this problem. Therefore, we propose an intelligent and dynamic threat-aware charging strategy that considers the threat level in the network to determine the next device to charge.

To this end, we begin this chapter by presenting background on anomaly, threat detection, the usefulness of trust management, and adaptive security. Then, we present the system model and we detail the impact of security solutions on the energy consumption of IoT devices. Thanks to the system model, we formulate our problem as a Markov Decision Process (MDP) problem and justify the use of Deep Reinforcement Learning (DRL) to determine the device to charge according to our criteria. We evaluate and present the performances of the proposed solution compared to non-learning approaches and a DRL-based agent that is not threat-aware. Finally, we study the complexity of the proposed solution and discuss its strengths and weaknesses.

5.1 Abstracting threats and security risks in IoT networks

To design a threat-aware wireless charging scheme, we need to define multiple concepts: What is a threat? How can we measure it? And how can we mitigate or nullify them?

Definition 5.1.1. A passive threat does not directly impact the IoT network. It is hard to detect it.

Definition 5.1.2. An active threat aims to impede the IoT network and its users. It is easier to detect an active threat than a passive threat.

Both threats may either be inside or outside the network [48, 52]. In our research work, we are more interested in threats related to malicious entities.

5.1.1 Detecting and quantifying threats

A threat may be estimated, detected, or even quantified. Before quantifying a threat, it is necessary to detect it, but it is a challenging task. As presented in the definitions of passive and active threats, it is easier to detect active threats than passive threats because the former ones have a direct impact on the IoT network. This impact can either be the loss of packets, node loss, or service interruption for instance [48, 52]. The cornerstone to efficiently detect and mitigate threats is the use of an Intrusion Detection System (IDS): it provides information on what kind of intrusion has been detected, where, and when. Consequently, it is an important tool to guarantee availability and enable adaptive security solutions. Zarpelao et al. [174] surveyed numerous IDS and classified them into four categories: Signature-based, Anomaly-based, Specification-based, and hybrid. Each category has advantages and drawbacks which are listed in Table 9.

IDS category	Advantages	Drawbacks
Signature-based	Performs well against known threats, easy to understand	Unable to determine new threats
Anomaly-based	Performs well to detect new threats	May lead to high false positive rates
Specification-based	Performs well to detect new threats	If specifications are bad, then false positives and false negatives may occur; need of an expert to determine what is an abnormal behavior
Hybrid	Minimizes the drawbacks of the other categories	More complex to implement than the other categories

Table 9: Advantages and drawbacks of each IDS category according to [174].

Trust-based solutions are also valid candidates for threat detection. Indeed, trust-based solutions [62, 64] define what are a good behavior, a bad one, and intermediate states (e.g. compromised) based on multiple criteria. These criteria can be network-related, such as throughput, packet delivery ratio, etc.

Recent trust-based solutions, threat, or intrusion detection systems use artificial intelligence approaches, especially machine learning and deep learning approaches, as described in Chapter 2. The high volume of available data and the need for adaptability can be tackled with these algorithms. However, due to their computational complexity, they need to be run on devices with good computational capabilities and a big (or unlimited) energy supply.

Then, it is possible to improve or decrease the security level of the IoT network according to the detected threat(s). Adaptive and context-aware security solutions are the way to go. Recent research works consider that the threat can be estimated to determine the best security level to use [15, 56, 57, 112]. If there is a low threat level, encryption, authentication, and privacy levels may be low [15]. If the threat level increases, authentication or encryption may be activated or strengthened. If the threat level is critical or maximum, one-time signatures, strong privacy measures (*e.g.* homomorphic encryption), or maximum encryption strength may be considered. Being adaptive to threats reduces energy consumption compared to traditional approaches which consider a fixed security level, as presented in Chapter 3. However, it induces variable energy consumption among IoT devices [57].

In this research, we consider that the IoT network has a hybrid, centralized threat detection module able to estimate the current threat. However, estimating the current threat level is still a difficult task since there exist many attacks and vulnerabilities in IoT

networks. We abstract the kind of threats and available security solutions. The current threat observation the MC receives from the threat detection module is the true current threat level, i.e. there are no false positive and no false negative detection results. The current threat is written as $danger^t$. This threat level may change after x seconds. Then, the MC will choose the device to charge according to this security information, along with information regarding remaining energy and the distance between the devices and the MC. In the next section, we present the system model, the energy consumption model of security, and the Markov Decision Process (MDP).

5.2 Our solution: Threat-aware charging strategy

5.2.1 System model

We consider an IoT network composed by a set $D = \{d_1, d_2, \ldots, d_n\}$ ($|D| = n$) of smart devices and a fixed Base Station (BS) with high computation power and unlimited energy. We also consider that each device d_i belongs to a class, denoted by c_k which defines the consumption rate of devices belonging to this class. The IoT network is deployed in a squared environment of dimensions $L * L$ and all the devices are powered with a rechargeable battery. The base station is located at coordinates $(0, 0)$.

For sake of simplicity, we consider that the Mobile Charger (MC) has an infinite amount of energy and that the devices have the same maximum amount of energy B_{max}. This assumption may hold if the MC can harvest energy from its movement (mechanical energy harvesting) or other sources, and the amount of harvested energy is greater than the energy it spends for moving and charging the IoT devices. We also consider that the MC may charge only one device at a given time. At each time step k, the MC has to determine which device it should charge.

We consider that each device $d_i \in D$ has an importance level imp_i, denoting its importance in the network. A high value of imp_i indicates that device d_i has critical tasks in the network. In this case, if d_i dies due to an attack, the network will be critically impacted. A low value of imp_i indicates that the tasks of d_i are not important with regard to the application/ongoing process. This importance value may be fixed (i.e. defined by operators during network deployment) or may change over time (e.g. a function of different network parameters or the ongoing processes). We consider in this chapter that imp_i is fixed and $imp_i > 0$. The main notations used in the chapter are summarized in Table 10.

Then, we define in the next subsection the impacts of adaptive security and threat-awareness on the energy consumption of the IoT devices.

Notation	Definition
D	Set of smart devices
DR	Set of dead devices during this round
E_i^t	Remaining energy of device d_i at time t
imp_i	Importance level of device d_i
$\mathcal{C}$	Set of device classes
η_{c_k}	Consumption rate of class c_k
$danger^t$	Current threat happening at time t
E_{def_k}	Energy cost of the k-th defense mechanism
N_{dead}^t	Number of dead nodes at time step t
v_{MC}	Velocity of the Mobile Charger (MC)

Table 10: List of the main notations used in this chapter.

5.2.2 Modeling the impact of security on energy consumption

To model the impact of the current threat level on IoT devices, we need to define what is a security need and how it can be fulfilled. Each device d_i has a security requirement denoted as req_i which can be task-dependent (e.g. data has to be encrypted for critical scenarios). For sake of simplicity, let's consider that the security solution provides u security levels. To fulfill this requirement, the device can activate a defense mechanism def_k, with strength pow_k. The chosen security level has to protect the network against the current threat $danger^t$, i.e.:

$$\forall t \in T, pow_k - danger^t \geq 0 \tag{5.1}$$

It means that if the security level chosen by the device at the beginning of the application cannot guarantee a sufficient protection against the detected threat, it has to choose a higher security level (e.g. stronger authentication or encryption, depending on the security requirements of the devices). The higher $danger^t$ is, the bigger consequences will be if an attack succeeds. Thus, $danger^t$ is a risk indicator.

This definition of a threat level is similar, yet different from [57, 112]. This defense mechanism def_k has a known energy consumption rate, written as E_{def_k}. The choice of the security mechanism can either be handled by the device [15] or a central component such as an SDN controller [57] which holds a global view of the network threats.

At time step t, the mobile charger is at coordinates (x_{MC}, y_{MC}), and has to choose a device d_i to charge with coordinates (x_i, y_i). The less time the mobile charger spends traveling, the more energy d_i will receive from the MC. With a velocity of v_{MC}, the MC takes $t_m = \frac{dist_i^t}{v_{MC}}$ seconds to reach d_i, and t_c seconds to fully charge d_i. If the MC chooses

to charge a device that was its last action taken, it is time penalized, i.e. it has to stay to this device during t_p seconds. This time penalty can be seen as a time between two fetch requests for energy levels and current threat estimation. Indeed, since a reinforcement learning approach does not exclude previous choices, it is necessary to consider the case when $a_t = a_{t+1}$, i.e. when the charger decides to do the same action during two consecutive time steps. In a wireless charging scenario, there is no need for the charger to charge the same device in two consecutive time steps.

Thus, the elapsed time between s_t and s_{t+1}, Δt is: $\Delta t = t_m + t_c$ or $\Delta t = t_p$.

$$\Delta t = \begin{cases} t_m + t_c & \textit{if } a_t \neq a_{t+1} \\ t_p & \textit{otherwise} \end{cases} \tag{5.2}$$

Finally, the remaining energy of a device d_i of class c_k at time step $t+1$ is:

$$E_i^{t+1} = \begin{cases} B_{max}, & \text{if } d_i \text{ is charged by the MC} \\ E_i^t - \Delta t(\eta_{c_k} + E_{def_k}), & \text{otherwise} \end{cases} \tag{5.3}$$

However, if the threat level changes during the transition between t and $t+1$, the energy consumption has to consider this change. Until the time where the threat level changes, the defense mechanism used is def_k. Then, the new threat occurs and the defense mechanism $def_{k'}$ is used, with its consumption rate $E_{def_{k'}}$ during the remaining time.

The main objective of the mobile charger is to minimize the number of dead devices while maximizing network lifetime given the network threat level. In this chapter, we consider that the network lifetime is the time duration until the number of dead devices exceeds a certain amount [16]. This choice is relevant since applications based on IoT networks may stop if too many nodes are dead. To solve the problem of threat-aware charging, we model it using a MDP, presented in the next subsection.

5.2.3 Enabling threat-aware wireless charging with DRL

To tackle the problem of proactively charging the devices for their future security needs, we formally model our problem as an MDP with the 4-tuple (S, A, R, P).

At time step t, the mobile charger observes a state s_t, made of the remaining energy of each device E_i^t, the distances $dist_i^t$ between the mobile charger and the current threat observation $danger^t$. A is the set of actions the mobile charger can take. At time t, the action a_t, $a_t \in A$ is $a_t = [d_i], i \in 1 \dots n$. It is the device d_i to recharge (to its maximum battery level).

R is the reward function of the mobile charger. It maps the states and the actions to a numerical value.

The reward $r(s_t, a_t)$ the MC will receive after taking action a_t considers:

- a penalty if a device d_i dies, r_d, multiplied with the importance imp_i of d_i,
- a penalty if the chosen device dies during the transit of the MC r_{dm},
- a penalty if the MC chooses the same device to charge r_s,
- a penalty if the MC chooses a dead device to charge r_{ad}
- a variable reward related to the transferred energy r_e

Penalties are negative values, whereas the variable reward r_e is a positive value.

Therefore, the reward $r(s_t, a_t)$ the MC receives after taking action a_t in state s_t is:

$$r(s_t, a_t) = \sum_{i \in DR} r_d.imp_i + r_{dm} + r_s + r_{ad} + r_e \tag{5.4}$$

where DR is the set of dead devices after taking action a_t. In equation 5.4, $r_{dm},\ r_s,$ and r_{ad} are only added if at least one of the above situations occurs.

Moreover, the future rewards starting from the time step t are discounted to give more importance to near rewards than to the far-future rewards. With a discount factor γ, this future discounted reward is defined as:

$$R_t = \sum_{i=t}^{T} \gamma^{i-t} r(s_i, a_i) \tag{5.5}$$

Finally, P is the transition probability of the environment between states s_t and s_{t+1} given that the mobile charger has taken action a^t. Given that the environment is dynamic and the mobile charger has no knowledge of the whole model, the transition probability is unknown. Hence, we consider model-free approaches such as Q-learning or Deep Q-learning. The MDP cannot be solved with Q-learning because the state space is too large. This would incur a too-large Q-Table, and thus, too much memory would be consumed. In Eq. 5.6, we remind the Q-learning update rule (which is also described in

Chapter 4, subsections 4.2.3 and 4.2.4).

$$\mathcal{Q}(s_t, a_t) \leftarrow \mathcal{Q}(s_t, a_t) + \alpha[r(s_t, a_t) + \gamma \max_a \mathcal{Q}(s_{t+1}, a) - \mathcal{Q}(s_t, a_t)] \tag{5.6}$$

The mobile charger has to learn the actions, i.e. what device to charge to maximize network lifetime given the current threat. Hence, the MC has to learn the policy $\pi(a|s)$ that maps the states (described beforehand) to the actions a. Furthermore, to maximize network lifetime, with a minimum amount of dead devices, the MC has to learn the optimal policy, denoted as π^*. It also needs to approximate the optimal Q-function which follows the optimal policy π^*, denoted as $\mathcal{Q}^{\pi^*}(s_t, a_t)$. π^* maximizes the expected value of R_t, i.e. $\pi^* = arg\max_\pi \mathbb{E}(R_t|\pi)$. To balance the choices between exploration and exploitation, it is useful to use an $\varepsilon-$greedy policy. Thus, the mobile charger will take the best action with a probability of $1-\varepsilon$ and choose a random action with a probability of ε, as explained in Chapter 4.

Since the state space is very large, computing and updating the Q-table to find the optimal policy π^* would be memory and time-consuming, as explained in Chapter 4. Thus, we focus on Deep Reinforcement Learning (DRL) [150] to ease the estimation of the Q-function. Indeed, deep neural networks are widely used for non-linear function approximation, and thus, they can approximate the Q-function. These neural networks are called Deep Q-Networks (DQN). DRL has many uses in IoT networks [155] such as security [65] or wireless charging [164].

For the DQN, we consider the same architecture as described in Chapter 4. Thus, we also use experience replay and a target network $\tilde{Q}-$network to stabilize the training process [150]. The memory pool $\mathcal{M}$ used in experience replay has entries written as $m_t = (s_t, a_t, r(s_t, a_t), s_{t+1})$. As previously explained in Chapter 4, this memory pool is useful for loss minimization during the training of the Q-network. This memory pool will attenuate the impact of the correlation of successive states. To improve and stabilize the learning process, a target network $\tilde{Q}$-network is also useful [164]. This target network also approximates a $\tilde{Q}$-function, called target function and denoted as $\tilde{Q}$. During the training process, the parameters of the target network will be updated thanks to the parameters of the Q-network. The loss function used to update the parameters of the $Q-$network is the Huber loss function [173] (more details are given in Chapter 4).

The underlying algorithm of our threat-aware charging strategy is presented in Algorithm 3. Firstly, the MC builds the state s_t (lines 2 to 5). Then, it uses it as an input to the Q-Network, determines the device that will be charged, and receives the corresponding reward (lines 7 to 9). After that, the MC updates the memory pool $\mathcal{M}$ with

Algorithm 3 Pseudo-code of the threat-aware energy management scheme

1: **for** each time step t **do**
2: Retrieve the current, observed threat level $danger^t$
3: Retrieve the current remaining energy E_i^t of each device d_i
4: Compute the distance $dist_i^t$ between the mobile charger and each device d_i
5: Build the observation s_t with $danger^t$, E_i^t, and $dist_i^t$, for $i = 1, \ldots, n$
6: Feed the Q-Network of the agent with the state s_t, the parameters θ
7: Take action a_t w.r.t. the $\varepsilon-$greedy policy
8: Receive reward $r(s_t, a_t)$ and process to the state s_{t+1}
9: Update the memory pool $\mathcal{M}$ with the tuple $(s_t, a_t, r(s_t, a_t), s_{t+1})$
10: Use the target Network $\tilde{Q}$ with a sample batch from the memory pool $\mathcal{M}$ to minimize a loss function $\mathcal{L}$
11: Update the Q-network thanks to the minimization of $\mathcal{L}$
12: Update the target Network $\tilde{Q}$ every λ steps with the weights of the Q-Network.
13: **end for**

the tuple $(s_t, a_t, r(s_t, a_t), s_{t+1})$. If there are enough samples in $\mathcal{M}$, it takes a random batch of samples and uses it to minimize a loss function $\mathcal{L}$. The results of this minimization are used to update the parameters of the Q-Network (lines 10 to 12). Moreover, for every λ time steps, the parameters of the target network $\tilde{Q}$ are updated with the parameters of the Q-network (line 12).

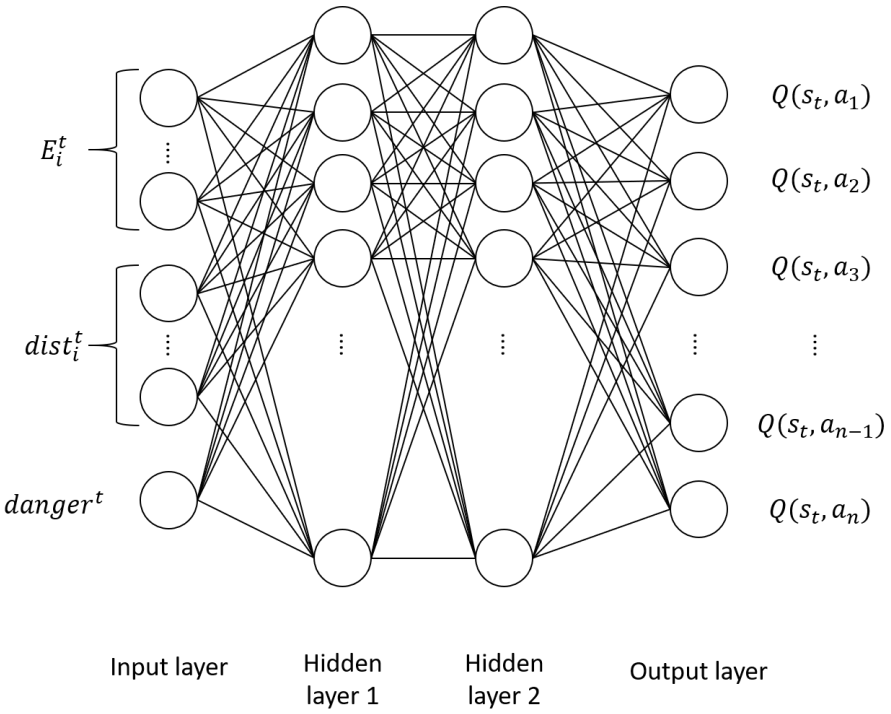


Figure 12: Q-network architecture of the threat-aware WMC (biases are not represented).

The neural network architecture we considered for the Q-network and the $\tilde{Q}-$network is made of two hidden layers of 256 neurons each, as depicted in Figure 12. The layers are fully interconnected. The input layer is fed with the state of the environment s_t, while the output layer is made of each possible action: the estimated Q-values. The function activation of the output layer is the ReLU (Rectified Linear Unit) function. ReLU is a good activation function due to its properties, compared to sigmoid or tanh activation functions. ReLU is more resilient to the vanishing gradient problem than sigmoid-based activation functions [175].

In the next section, we will present the settings of our simulations and the obtained results.

5.3 Performance evaluation

5.3.1 Simulation description

Parameter	Value	
Environment size	$100m * 100m$	
Number of devices $n_{devices}$	$[15, 20] + 25$	
Threshold of dead devices	$0.6n_{devices}$	
Maximum battery B_{max}	$100J$	
Consumption rates	$[0.02, 0.04]$	
Initial residual energy	$\mathcal{U}(0.45, 0.55) * B_{max}$	
Charging rate	5 W	
Moving speed v_{MC}	$1m/s$	
Time penalty t_p	60s	
Threat renewal period x	200 s	
r_d	-1	
r_{dm}	-50	
r_s	-10	
r_{ad}	-100	
r_e	$B_{max} - E_i^t$ (J)	
Total number of steps	2.10^6 steps	
Number of transition collection steps	1.10^5 steps	
Batch size	256	
Size of memory replay buffer	2.10^5	
Update interval of target network λ	1.10^5 steps	
Learning rate α	3.10^{-5}	
Discount factor γ	0.9	
Exploration probability ε	0.05	
Neural network architecture	$[256, 256]$, ReLU activation function	
Threat label	Range	Probability
No threat	0	0.1
Low threat	$]0, 25]$	0.5
Medium threat	$]25, 60]$	0.3
High threat	$[61, 100[$	0.1
Solution label	Strength	Energy consumption
Solution 1	25	0.015 J/s
Solution 2	60	0.09 J/s
Solution 3	101	0.25 J/s

Table 11: List of the parameters of the experiments.

We conducted our experiments on an HP Elitebook with an Intel Core i7-10610U CPU @1.8GHz with 8 cores. The simulations were done using Python 3.9 using OpenAI

Gym [176] and Stable-baselines3 libraries [177]. Stable-baselines3 is a Python library providing reliable implementations of RL and DRL algorithms based on the deep learning library PyTorch.

We randomly generate devices in a square $(100m * 100m)$ according to a uniform distribution. The threat level evolves between $[0, 100)$ and is separated into 4 levels: no threat, low threat, medium threat, and high threat. Each threat level has a range of threat values. First, a level is generated according to the distribution in Table 11. Then, the current threat value is generated according to a uniform distribution on the related level, described at the bottom of Table 11. A new threat level is generated every 200 seconds. The neural network architecture is made of 2 hidden layers with 256 neurons each, and the activation function is a ReLU activation. Hyperparameters were chosen empirically. We ran for each environment 100 simulations. Then we took the average of the results to plot them. Furthermore, for training and evaluation, we limited the simulation time to $4 * 24 * 3600 = 345600$ seconds. This is done to prevent endless training episodes and endless evaluation scenarios. The other parameters of the simulation are described in Table 11.

We compared our scheme, Threat-Aware DQN, with the following:

- A fully random approach where the MC randomly chooses the next device to charge;
- A random-aware approach where the MC randomly chooses the next device to charge among the remaining alive devices;
- A greedy approach that considers the device with the lowest remaining energy as the next device to charge;
- Another smart agent based on Deep Q-learning whose observation space is restricted to the remaining energy and the distance to the devices. It is named *not TA-DQN*. This agent is trained with the same parameters as TA-DQN described in Table 11;
- The baseline lifetime of the network when there is no threat, but the security needs are covered and when the threat level is maximum, with the security needs fulfilled. In these setups, the initial energy of all devices is set to B_{max}.

First, we study the loss training curves and mean reward curves of the TA-DQN and not TA-DQN agent. We evaluate the maximum average lifetime and the total reward received by the different schemes. The total reward is a good indicator of the effectiveness of a charging scheme.

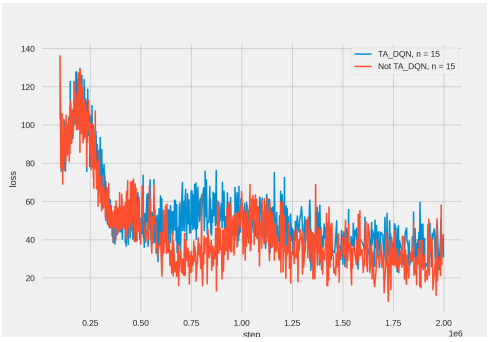
5.3.2 Simulation results

Training phase

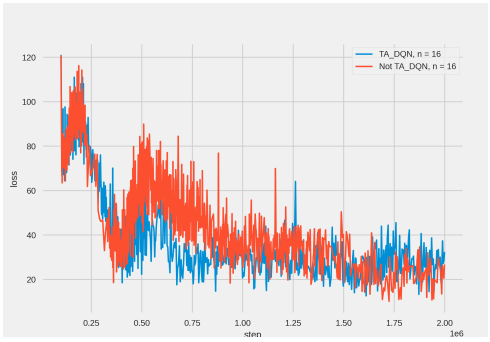
First of all, we present the training losses of our threat-aware agent (TA-DQN) for the different environment sizes in Figure 13. Furthermore, we compare these losses with the losses of an agent that is not threat-aware (written as Not TA-DQN). The loss values are computed thanks to equations 4.16 and 4.17 (in Chapter 4).

For both agents, we can observe that loss curves are high at the beginning of the experiments (after the phase of collection of transition steps has ended at $100k$ steps). The loss then decreases quickly for both agents. This decrease shows that the agent can predict much more reliably its future performance. However, when the number of steps increases, our threat-aware agent, TA-DQN, observes a higher loss than its not-threat-aware counterpart (not TA-DQN). It can be interpreted as the threat-aware agent makes fewer good decisions during the training phase compared to the not threat-aware agent according to the observed state. However, these decisions are better over time since the loss curves have decreasing tendencies.

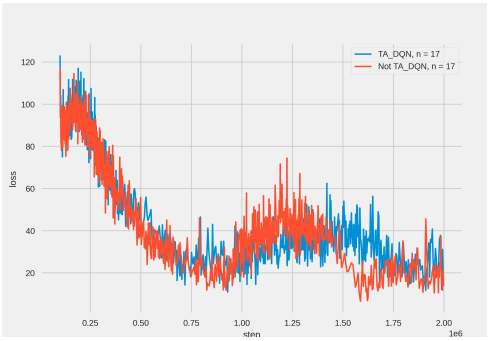
Then, we analyze the mean rewards obtained during the training phase for both TA-DQN and not TA-DQN in Figure 14. We can observe that, during the training phase, our agent may either earn less mean rewards (for $n = 15, 16, 20, 25$) or more (for $n = 17, 18, 19$). The difference is more present for $n = 20$ and $n = 25$, where the not TA-DQN agent earns more mean rewards than the TA-DQN agent. However, there are a lot of fluctuations for the not TA-DQN agent, and even some catastrophic forgetting for the case $n = 17$ where the reward was quite high for $1.4M$ timesteps, but then it dramatically dropped. The same observation can be done for the TA-DQN agent when $n = 19$. There is a drop from $600k$ until $1.3M$ timesteps, and then it increases again. The TA-DQN agent seems to have more difficulties getting good rewards during the training phase. Thus, we need to determine if this behavior may be observed during the evaluation phase, or if the TA-DQN agent can perform better than the not TA-DQN agent in some scenarios.



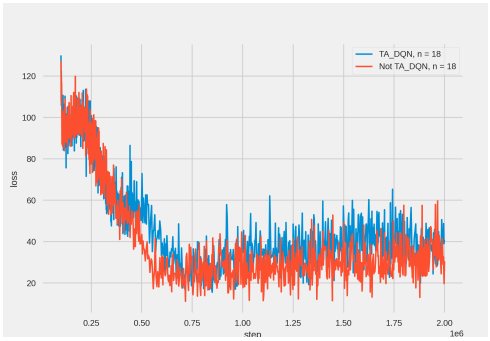
(a) $n = 15$



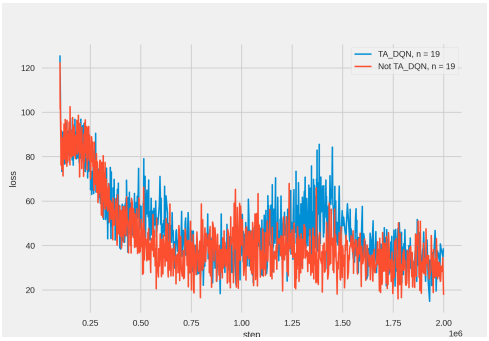
(b) $n = 16$



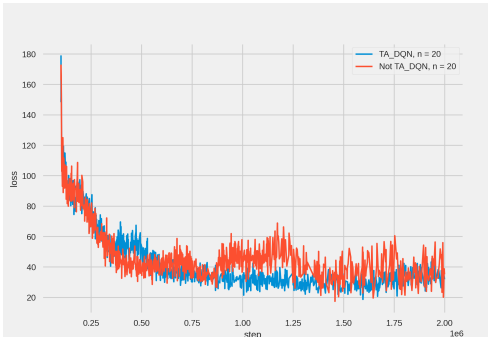
(c) $n = 17$



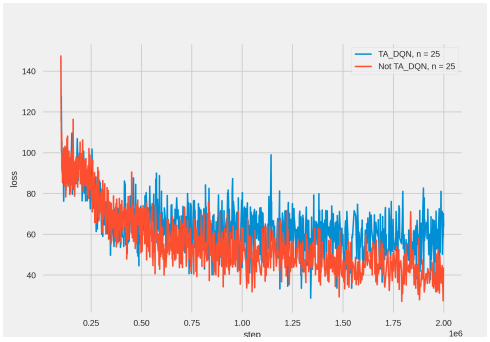
(d) $n = 18$



(e) $n = 19$

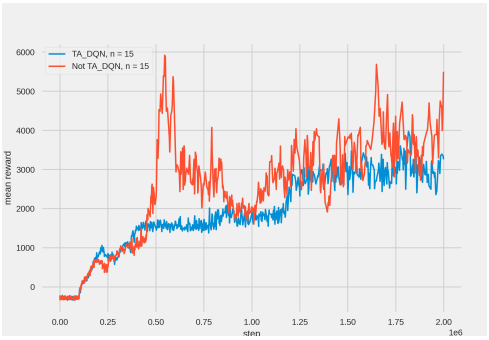


(f) $n = 20$

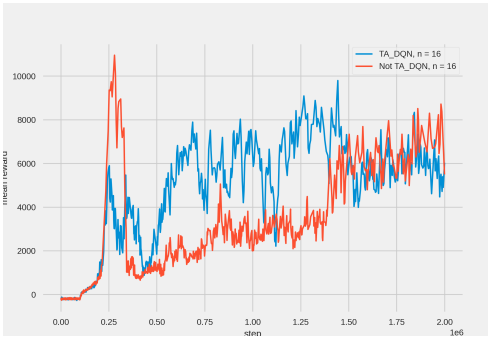


(g) $n = 25$

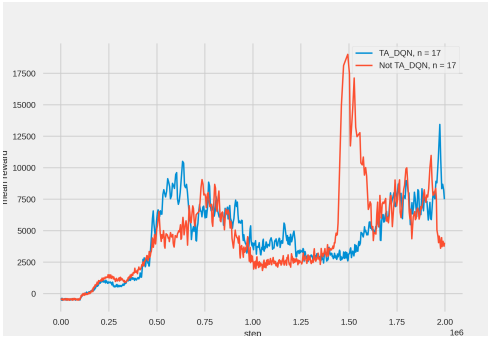
Figure 13: Loss curves during the training for both learning agents.



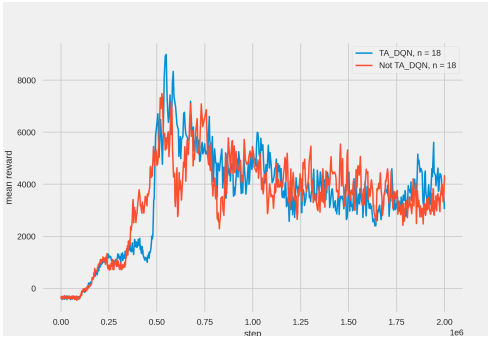
(a) $n = 15$



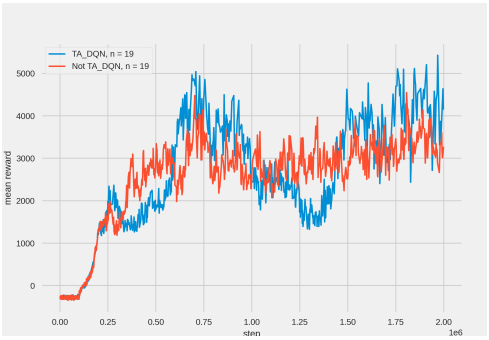
(b) $n = 16$



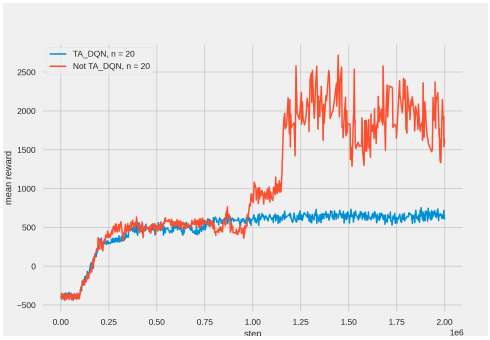
(c) $n = 17$



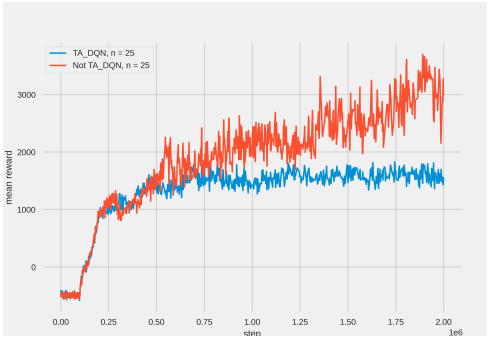
(d) $n = 18$



(e) $n = 19$



(f) $n = 20$



(g) $n = 25$

Figure 14: Mean reward curves during the training for both learning agents.

Evaluation phase

We observed that the TA-DQN agent may have difficulties outperforming a smart agent that does not know about the current threat (not TA-DQN) during the training

phase. However, is it still the case during evaluation? To determine if this assumption holds or not, we evaluate the performances of TA-DQN and not TA-DQN agents over 100 runs; then, we take the average of the results to plot them.

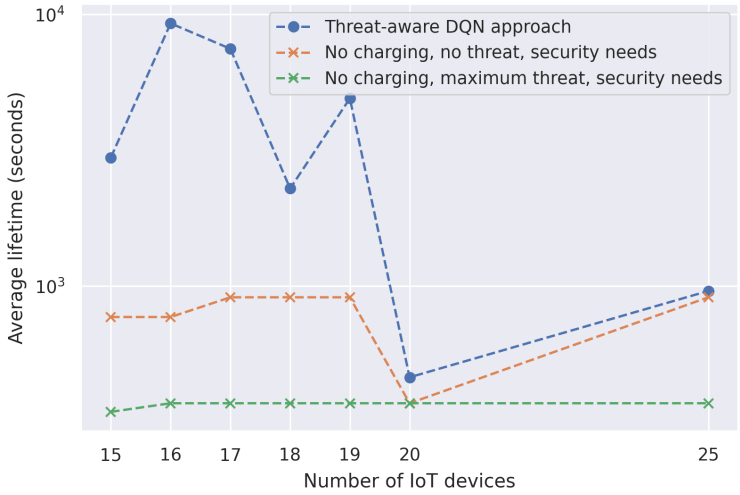


Figure 15: Comparison of our scheme to other baseline lifetimes.

First of all, we evaluate and plot in Figure 15 the effectiveness of the TA-DQN agent to the baseline lifetime, when there is no charging strategy and the devices have their battery full. The TA-DQN agent can extend the network lifetime compared to when the minimum security requirements are fulfilled, when the threat level is maximum, and there is no MC in the environment. Compared to an environment with 20 nodes in which there is no threat, but security is ensured, our TA-DQN agent can extend the network lifetime up to 462 seconds, from 371 seconds, thus, an increase of 24.52%.

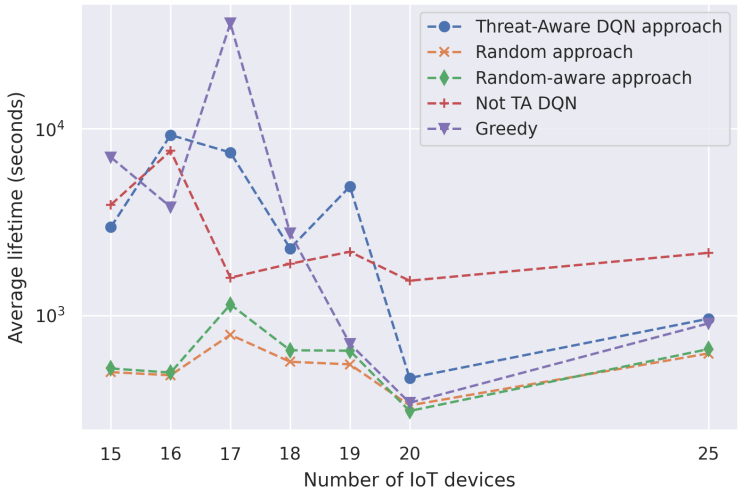


Figure 16: Comparison of the average lifetime of our scheme with other approaches.

In Fig. 16, we compare the effectiveness of our solution to other approaches. For network sizes of 16 to 19 nodes, our threat-aware agent performs better than a not threat-aware agent (using DRL). For instance, the average lifetime of our IoT network with 16 nodes using the threat-aware agent is 9259 seconds, whereas the lifetime with the Not TA-DQN agent is about 7615 seconds. Thus, lifetime is increased by 21.59%. For bigger network sizes, our TA-DQN approach performs better than greedy and random approaches but underperforms compared to a not threat-aware approach.

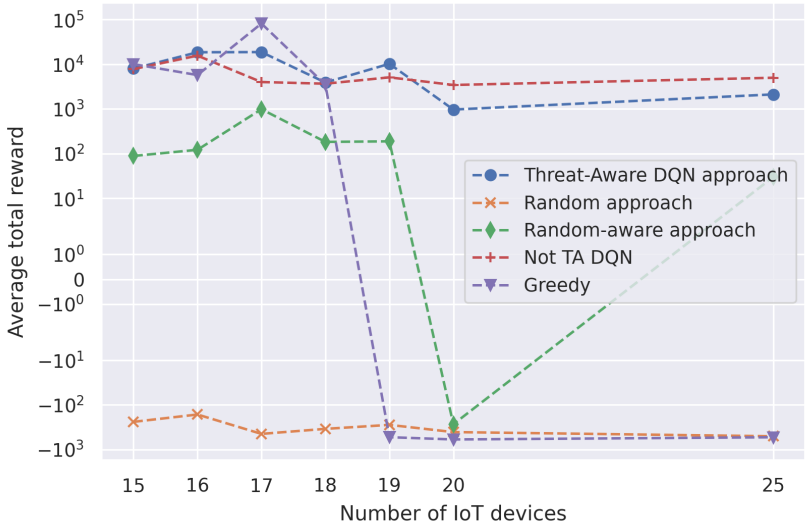


Figure 17: Mean total rewards of our scheme versus other approaches (log-scale).

In Fig. 17, we compare the total received rewards of the different methods. Our

approach receives more reward for network sizes of 16 to 19 devices than the Not TA-DQN agent and receives less for 15, 20, and 25 devices. This is tied to the average lifetime of the network with our threat-aware agent. However, against, random approaches and the greedy approach, the TA-DQN agent has better rewards for 19, 20, and 25 nodes since it can determine the best device to charge according to the current threat. However, since it aims to charge the device that will need the most energy for its security needs, with an emphasis on devices with lowest remaining energy, it may neglect overtime devices that may consume more, but have more energy at the beginning of a time step. Hence, it struggles against a not threat-aware agent that solely aims to charge the devices without this threat information and plans its charging tour to directly minimize the number of dead devices.

These results confirm the observations of the training phase where the TA-DQN agent earned less mean rewards compared to the not TA-DQN agent, especially for $n = 20$ and $n = 25$. However, for $n = 15$ and $n = 16$, TA-DQN earned similar rewards compared to the not TA-DQN agent. We can observe that during the evaluation phase, the TA-DQN agent was able to get more rewards than during the training phase for these cases.

Although TA-DQN does not perform better than other approaches on all network sizes, it shows that the consideration of threat-awareness to design a wireless charging strategy is a valid approach for small-sized, but spread networks. The obtained results with our agent considered that devices do not have an initial remaining energy at the maximum capacity of their battery. Even with this constraint, the network lifetime with our agent is increased compared to baseline lifetimes, greedy and random approaches. Thus, a threat-aware wireless charger is able to improve device lifetime in the same fashion, and even better than non-learning approaches and a smart agent.

5.3.3 Complexity study

Although the use of a neural network to approximate the Q-table is what made successful DRL approaches, there is the counterpart that training a neural network is time and memory-consuming [150].

Deep reinforcement learning approaches are hard to implement on memory and energy-constrained devices, especially if the neural network is huge. In that case, for deeper neural networks, there is a huge number of parameters θ and biases to learn. We are interested in determining if our models can run on a resource-constrained WMC, which does not have as much as resources as our experimental platform. To do so, multiple variables can be studied: the number of parameters of the deep Q-network, the size it may take, or the size of the state space, which directly impacts the number of parameters to

learn.

The number of parameters θ of the $Q-$network can be computed easily since the $Q-$network has dense connections, as shown in Figure 12. It means that each neuron of the k-th layer is connected to all the neurons of the $k+1$-th layer. For each connection between two neurons i and j, there is a weight parameter $\theta_{i,j}$.

Let $|h_k|$ be the size of the $k-th$ hidden layer and $|A|$ the number of possible actions (which is equal to the number of IoT devices n). Then, the number of weight parameters θ is:

$$|\theta| = |s_t||h_1| + |h_1||h_2| + |h_2||A| \tag{5.7}$$

The number of bias weights $|b|$ is:

$$|b| = |h_1| + |h_2| + |A| \tag{5.8}$$

Thus, the total number of parameters the model has to learn is: $|\theta| + |b|$.

In this chapter, the size of s_t is equal to $2n + |\text{danger}^t|$. The variable danger^t is a discrete variable. To be able to use it, it has to be implemented with a one-hot encoding vector. Thus, for four discrete threat levels, four neurons are needed, hence, the size of the state s_t is $s_t = 2n + 4$. In Table 12, we list for each network size, the size of the state, the total number of parameters that have to be learned, and the estimated size if the parameters are encoded using 8 bytes.

Network size n	$\|s_t\| = 2n + 4$	total number of parameters $\|\theta\| + \|b\|$	Estimated size (8 bytes per parameter, in Kebibytes)
15	34	78607	614.1 KiB
16	36	79376	620.1 KiB
17	38	80145	626.1 KiB
18	40	80914	632.1 KiB
19	42	81683	638.1 KiB
20	44	82452	644.1 KiB
25	54	86297	674.2 KiB

Table 12: Parameter complexity of our approach as a function of the size of the network.

Compared to existing works that use RL or DRL approaches, our state space is quite small. The state space is problem and model-dependent. However, as shown before, it has an impact on the number of parameters and the potential memory size of the model. In

Table 13, we compare the size of the state space of related works to our solution (presented in Chapter 4). Variable n is the number of IoT devices (or charging locations) in the network. Our approach has a lower state space size than the existing works, except the work of Van Quan et al. [163] that considered only the charging location as the variable in the state space (The environment is discretized into charging locations, in which IoT devices are present).

Reference and Year	RL or DRL?	State space size $\|s_t\|$
Wei et al. [162], 2018	RL	$3(n+2)+1$
Van Quan et al. [163], 2020	RL	l (number of charging locations)
Cao et al. [164], 2021	DRL	$5n+1$
Yang et al. [148], 2021	DRL	$3(n+1)$
Bui et al. [165], 2022	DRL	$5n+7$
Context-aware WMC, Chapter 4, 2022	DRL	$3n+1$
TA-DQN, this Chapter, 2023	DRL	$2n+4$
Not TA-DQN, this Chapter, 2023	DRL	$2n$

Table 13: Comparison of state space sizes of related works.

Furthermore, the total size of our agents is reported in Table 14, demonstrating that they can effectively run on memory-constrained devices.

Agent name	Network size n	File size
DQN_15_devices_simus_100J	15	1.22 MiB
DQN_16_devices_simus_100J	16	1.24 MiB
DQN_17_devices_simus_100J	17	1.25 MiB
DQN_18_devices_simus_100J	18	1.26 MiB
DQN_19_devices_simus_100J	19	1.28 MiB
DQN_20_devices_simus_100J	20	1.29 MiB
DQN_25_devices_simus_100J	25	1.35 MiB

Table 14: File sizes of the different threat-aware agents as a function of the network size (in Mebibytes).

The models can effectively run on IoT platforms with little available memory. However, the training phase has to be done on either servers or workstations, as the learning process of RL and DRL is time-consuming, especially if the number of training steps is huge.

5.3.4 Discussion and limitations

We observed that our approach extends network lifetime under the hypothesis that we can estimate the network threat. The strategy improves network lifetime compared to non-learning-based approaches and even a learning approach that does not know the current threat level in the IoT network.

We also considered that the threat detection module always outputs the right result, i.e. it is an oracle. In a real-life scenario, threat and intrusion detection modules are prone to either false negative or positive results or may even sometimes not detect threats. An interesting scenario to investigate would be the case when the threat prediction is false.

A major difficulty we encountered during this research is that it is near impossible to compare our results with existing research works. Indeed, for other deep reinforcement learning-based solutions, some papers did not present the parameters of their neural networks, what were the values of the hyperparameters, etc. This is a serious flaw in the reproducibility of RL-based experiments. Furthermore, for research papers that considered the use of GPU servers for their experiments, GPUs are prone to non-deterministic behaviors, leading to hard reproducibility of the experiments [178, 179]. Nowadays, open and reproducible science should be the default standard in all domains, and DRL-based solutions are no exception.

It is also important to note that DRL is a black box, i.e. it is not possible to fully explain why a decision was taken, due to the use of a neural network approximator for the Q-table. Moreover, if the IoT network gets new devices (e.g. a new monitoring area or a new factory), a whole new agent has to be trained, consuming time and computational resources. Furthermore, the larger the environment gets, the higher the number of input neurons will be (the state has a size of $2n + 4$ with n as the number of IoT devices). In Table 15, we list the advantages and drawbacks of the proposed solution.

Advantages	Drawbacks
Consideration of threat awareness	A model where the WMC with infinite energy may not be a realistic assumption.
Dynamic path planning	Threat estimation is a hard task.
Improvements compared to non-learning approaches	The strategy is not suited for critical environments.
May build on different categories of threat detection architectures	The strategy may be inefficient for dense networks.
Memory footprints of our agents are small	The agent has to be trained in a simulated world before. How good is this simulated world compared to reality?
	An attacker who has physical access to the network and knows the position of the WMC can impede it. It can also attack the furthest devices from the WMC.

Table 15: Advantages and drawbacks of the threat-aware charging strategy.

The network architecture we considered is that the threat detection module is located at the Base Station (BS). Data is sent from the devices to the BS. A possible network architecture is the use of Software-defined networking. Indeed, SDN along Virtual Network Functions can easily implement different security functionalities (threat detection, mitigation, etc.). Thus, it is possible to consider a network function that outputs a threat level value and sends it to the WMC.

5.4 Conclusion

In this chapter, we first discussed that the impact of security solutions is important on the energy consumption of IoT devices, especially if they have small batteries. Moreover, if the threat level increases, then the devices need to use stronger security solutions. Hence, the devices consume more energy to protect themselves and their communications. Thus, using a threat-aware charging strategy to improve network lifetime is a promising approach for IoT networks that use energy-consuming security solutions. Therefore, we proposed a new energy management scheme based on deep reinforcement learning. The scheme is threat-aware and relies on wireless mobile charging. Threat awareness enables the estimation of the future energy consumption of the security mechanisms, and thus, the charger determines which device will need energy for its security needs. Deep reinforcement learning is used to tackle large state and action spaces. The proposed scheme can extend network lifetime when there are varying threat levels compared to static approaches and a DQN-based approach that is not threat-aware.

Chapter 6

Conclusion and Perspectives

6.1 Conclusion

Nowadays, the Internet of Things (IoT) empowers many applications, such as smart cities, smart health, or industry 4.0. The IoT gained a lot of popularity in the past years and attracted many companies and researchers. However, there are still major flaws impeding its worldwide adoption such as energy efficiency and security. Security solutions may not be well implemented or may lack adaptability against threats or context changes, etc. while consuming a lot of energy. Furthermore, IoT devices have a finite lifetime since they are powered thanks to batteries or energy harvesting. That is why some IoT devices do not use any security solutions or use depreciated solutions that may consume less energy. In this thesis, we proposed different approaches to overcome the problem of energy-efficient security for IoT networks. The main goal of the conducted research is to efficiently secure IoT networks while maximizing their lifetime. To this end, we studied different categories of security solutions and energy-efficient approaches, such as wireless charging techniques.

We began this thesis by introducing the Internet of Things along with the important challenges of energy and security in Chapter 2. This chapter laid the background on energy management and recent IoT security solutions. We also discussed new approaches to IoT security such as Artificial Intelligence (AI) or Software-Defined Networking (SDN). Then, we presented a literature review of energy-efficient security solutions that may protect the IoT well in Chapter 3. These solutions are mandatory to reduce the energy consumption of IoT security compared to traditional approaches. We identified the main characteristics of each solution and we classified them into five categories. Furthermore, we investigated whether the surveyed IoT security solutions used energy management or energy harvesting approaches. Then, we discussed the usefulness of AI and SDN for the design of energy-efficient IoT security solutions. This state of the art gave us the main research direction of this thesis: wireless charging for IoT security.

We explored this research direction in chapter 4 in which we introduced a context-aware wireless charging strategy for IoT networks. Existing strategies, whether they are

offline or online, learning-based or not, did not consider context information to make a charging decision. Thus, we proposed a model for the problem of context-aware wireless charging in IoT networks. The first contribution was a framework to model the context with the current and future events with their impact on the energy consumption of the devices. Then, we presented the context-aware charging strategy based on deep reinforcement learning and presented Deep Q-learning as the solving tool. Subsequently, we modified the presented model to tackle the problem of threat-aware energy provisioning in Chapter 5. Threat awareness gives information on the security solutions that will be used to mitigate the detected threat. Thus, it is possible to determine the estimated energy consumption of the security solution(s) and identify the devices that need energy for their security needs. To this end, we proposed a threat-aware charging strategy for IoT networks based on Deep Q-learning. Experiments show that network lifetime can be improved compared to non-learning approaches, and even compared to an agent based on Deep Q-learning that does not know about the current threat. Finally, we concluded this manuscript in this chapter.

6.2 Open Issues and Perspectives

With the conducted research, we identified several open issues that may lead to new contributions.

6.2.1 Energy-efficient security

Although there is still work to do in this field, we did not tackle this domain as we focused on energy provisioning for IoT security. However, there is still room to improve the energy efficiency of IoT security. First, researchers and companies should not underestimate the energy consumption of anomaly and intrusion detection. Since the most recent research works use deep learning-based algorithms, their energy consumption during the training phase, testing phase, and use phase cannot be overlooked. More research needs to be done on the energy consumption of online and offline learning algorithms. Then, more room should be given to lightweight encryption algorithms in IoT networks. The National Institute of Standards and Technology (NIST) has chosen the Ascon family of ciphers as the standard for lightweight cryptography, paving the way for optimized applications, extended research on the considered algorithms, benchmarks, etc. Third, threat-awareness and context-awareness should have more importance for the choice of a security level (at the link layer for instance, the choice of the authentication method). This awareness will lead to reduced energy consumption. We may tackle this area in the future, but we identified more research directions presented in the next subsection.

6.2.2 Improving energy provisioning for security

In this thesis, we considered wireless mobile charging to guarantee energy provisioning for IoT security using Deep Q-learning. Recent research showed that actor-critic methods have better efficiency in solving MDPs than Deep Q-networks. Actor-critic methods are based on the gradient descent and function approximators for the actor (the part responsible for the policy) and the critic (the value function). However, the drawback is that they also need computing power, depending on the underlying neural network architecture considered. Nevertheless, considering other reinforcement algorithms will allow us to compare on-policy and off-policy approaches and establish benchmarks in different scenarios. This is the first possible extension to our research work.

Moreover, the model we considered in Chapter 5 had a wireless mobile charger with an infinite amount of energy. In the future, we aim to investigate the scenario of how the Wireless Mobile Charger (WMC) behaves when it has a limited amount of energy. What decisions it should take when it has a limited amount of energy whereas the threat level varies and impacts the energy consumption rate of IoT devices? A possible system model to consider is the one presented in Chapter 4 in which there are many energy access points that a WMC with limited energy can visit to recharge its battery. Then, there is a new model that has to be considered, similar to the one proposed in Chapter 4 in which the remaining energy of the wireless mobile charger is part of the state s_t. Moreover, a problem we want to study is the problem of joint context-aware and threat-aware wireless charging. We studied context-aware and threat-aware wireless charging separately, but decision-making when both contexts and threats vary leverages new research questions. Does a charger that knows about the current threat and current context perform better than a charger that is only threat-aware (or context-aware)?

Furthermore, the environment we considered was a continuous world environment. Another interesting model for the environment is the square-grid world. In a square-grid world, the agent navigates from one square to another and can take actions for each square if they contain a device or a cluster of devices. With this model, it is possible to consider wider areas and widen the study to the recharge of clusters instead of devices in a field for instance. Then, it is possible to study the average energy consumption of a cluster and the impact of a network attack (or other attacks) against a cluster and determine if this cluster should be recharged first or not. These new models are a second research direction worth investigating.

A third research direction we may also study is the proposition of a new model based on a Partially Observable Markov Decision Process (POMDP). POMDPs can model environments with many uncertainties. In the context of threat-aware wireless charging,

uncertainties may arise from the following:

- The availability of the current threat level. How should the WMC behave when the threat level is not available or cannot be computed by the threat detection module?
- The remaining energy of IoT devices. The WMC may not be able to get all the remaining energy levels of IoT devices in the field. Furthermore, the remaining energy of each device may not be the same when the information is sent and when the WMC receives it.

A fourth research direction we will study are the interactions between an attacker and a smart wireless charger. A clear limit we identified to our research work presented in chapter 5 is that if the attacker knows the position of the WMC, they can impede it or destroy it. The attacker could also target a single device or a set of devices far away from the WMC. This will increase the threat level in the network and the energy consumption of the devices that will have to use stronger defense mechanisms to mitigate the attack. An interesting research direction would be to study a more precise model in which an attacker knows the position of the charger. How should the charger act to maximize network lifetime given this threat? Furthermore, if this device is important for the network/application, the wireless mobile charger will have to recharge it. Then, the attacker can identify the most critical devices in the network to make the charger loop between the attacked devices and neglect the other devices. Hence, we may propose a new mechanism that would be able to improve network lifetime with harsher conditions.

Bibliography

[1] C. Perera et al. "A Survey on Internet of Things From Industrial Market Perspective". In: *IEEE Access* 2 (2014), pp. 1660–1679. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2015.2389854.

[2] Shwet Ketu and Pramod Kumar Mishra. "Internet of Healthcare Things: A Contemporary Survey". In: *Journal of Network and Computer Applications* 192 (Oct. 15, 2021), p. 103179. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2021.103179.

[3] Meghna Raj et al. "A Survey on the Role of Internet of Things for Adopting and Promoting Agriculture 4.0". In: *Journal of Network and Computer Applications* 187 (Aug. 1, 2021), p. 103107. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2021.103107.

[4] Hugh Boyes et al. "The Industrial Internet of Things (IIoT): An Analysis Framework". In: *Computers in Industry* 101 (Oct. 1, 2018), pp. 1–12. ISSN: 0166-3615. DOI: 10.1016/j.compind.2018.04.015.

[5] Ruth Ande et al. "Internet of Things: Evolution and Technologies from a Security Perspective". In: *Sustainable Cities and Society* 54 (Mar. 1, 2020), p. 101728. ISSN: 2210-6707. DOI: 10.1016/j.scs.2019.101728.

[6] Bomin Mao et al. "AI Models for Green Communications Towards 6G". In: *IEEE Communications Surveys & Tutorials* 24.1 (2022), pp. 210–247. ISSN: 1553-877X. DOI: 10.1109/COMST.2021.3130901.

[7] A. Al-Fuqaha et al. "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications". In: *IEEE Communications Surveys Tutorials* 17.4 (Fourthquarter 2015), pp. 2347–2376. ISSN: 1553-877X. DOI: 10.1109/COMST.2015.2444095.

[8] Pallavi Sethi and Smruti R. Sarangi. "Internet of Things: Architectures, Protocols, and Applications". In: *Journal of Electrical and Computer Engineering* 2017 (Jan. 26, 2017), e9324035. ISSN: 2090-0147. DOI: 10.1155/2017/9324035.

[9] M. A. Al-Garadi et al. "A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security". In: *IEEE Communications Surveys Tutorials* 22.3 (3 thirdquarter 2020), pp. 1646–1685. ISSN: 1553-877X. DOI: 10.1109/COMST.2020.2988293.

[10] Vikas Hassija et al. "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures". In: *IEEE Access* 7 (2019), pp. 82721–82743. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2924045.

[11] International Telecommunication Union. *Series Y: Global Information Infrastructure, Internet Protocol Aspects, next-Generation Networks, Internet of Things and Smart Cities, ITU-T Recommendation Y.2060*. June 2012. URL: https://www.itu.int/rec/T-REC-Y.2060-201206-I/en.

[12] International Telecommunication Union. *Series Y: Global Information Infrastructure, Internet Protocol Aspects, next-Generation Networks, Internet of Things and Smart Cities, ITU-T Recommendation Y.4455*. Oct. 2017. URL: https://www.itu.int/rec/T-REC-Y.4455-201710-I/en.

[13] Miao Wu et al. "Research on the Architecture of Internet of Things". In: *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*. 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE). Vol. 5. Aug. 2010, pp. V5–484–V5–487. DOI: 10.1109/ICACTE.2010.5579493.

[14] Sarah A. Al-Qaseemi et al. "IoT Architecture Challenges and Issues: Lack of Standardization". In: *2016 Future Technologies Conference (FTC)*. 2016 Future Technologies Conference (FTC). Dec. 2016, pp. 731–738. DOI: 10.1109/FTC.2016.7821686.

[15] Antonio Vincenzo Taddeo, Marcello Mura, and Alberto Ferrante. "QoS and Security in Energy-Harvesting Wireless Sensor Networks". In: *2010 International Conference on Security and Cryptography (SECRYPT)*. 2010 International Conference on Security and Cryptography (SECRYPT). July 2010, pp. 1–10.

[16] Halil Yetgin et al. "A Survey of Network Lifetime Maximization Techniques in Wireless Sensor Networks". In: *IEEE Communications Surveys & Tutorials* 19.2 (2017), pp. 828–854. ISSN: 1553-877X. DOI: 10.1109/COMST.2017.2650979.

[17] Dipak K. Sah and Tarachand Amgoth. "Renewable Energy Harvesting Schemes in Wireless Sensor Networks: A Survey". In: *Information Fusion* 63 (Nov. 1, 2020), pp. 223–247. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2020.07.005.

[18] R. Arshad et al. "Green IoT: An Investigation on Energy Saving Practices for 2020 and Beyond". In: *IEEE Access* 5 (2017), pp. 15667–15681. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2686092.

[19] Teodora Sanislav et al. "Energy Harvesting Techniques for Internet of Things (IoT)". In: *IEEE Access* 9 (2021), pp. 39530–39549. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3064066.

[20] Patrick Schaumont et al. "Secure Authentication with Energy-Harvesting: A Multi-Dimensional Balancing Act". In: *Sustainable Computing: Informatics and Systems* 12 (Dec. 1, 2016), pp. 83–95. ISSN: 2210-5379. DOI: 10.1016/j.suscom.2015.10.002.

[21] D. Mishra et al. “Smart RF Energy Harvesting Communications: Challenges and Opportunities”. In: *IEEE Communications Magazine* 53.4 (4 Apr. 2015), pp. 70–78. ISSN: 1558-1896. DOI: 10.1109/MCOM.2015.7081078.

[22] Jiří Libich et al. “Supercapacitors: Properties and Applications”. In: *Journal of Energy Storage* 17 (June 1, 2018), pp. 224–227. ISSN: 2352-152X. DOI: 10.1016/j.est.2018.03.012.

[23] Borja Martinez et al. “The Power of Models: Modeling Power Consumption for IoT Devices”. In: *IEEE Sensors Journal* 15.10 (Oct. 2015), pp. 5777–5789. ISSN: 1558-1748. DOI: 10.1109/JSEN.2015.2445094.

[24] Isabel Dietrich and Falko Dressler. “On the Lifetime of Wireless Sensor Networks”. In: *ACM Transactions on Sensor Networks* 5.1 (Feb. 2009), pp. 1–39. ISSN: 1550-4859, 1550-4867. DOI: 10.1145/1464420.1464425.

[25] Tifenn Rault, Abdelmadjid Bouabdallah, and Yacine Challal. “Energy Efficiency in Wireless Sensor Networks: A Top-down Survey”. In: *Computer Networks* 67.4 (July 2014), pp. 104–122.

[26] Sana Benhamaid, Abdelmadjid Bouabdallah, and Hicham Lakhlef. “Recent Advances in Energy Management for Green-IoT: An up-to-Date and Comprehensive Survey”. In: *Journal of Network and Computer Applications* 198 (Feb. 1, 2022), p. 103257. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2021.103257.

[27] S. F. Abedin et al. “A System Model for Energy Efficient Green-IoT Network”. In: *2015 International Conference on Information Networking (ICOIN)*. 2015 International Conference on Information Networking (ICOIN). Jan. 2015, pp. 177–182. DOI: 10.1109/ICOIN.2015.7057878.

[28] Ghada Jaber et al. “An Adaptive Duty-Cycle Mechanism for Energy Efficient Wireless Sensor Networks, Based on Information Centric Networking Design”. In: *Wireless Networks* 26.2 (2 Feb. 1, 2020), pp. 791–805. ISSN: 1572-8196. DOI: 10.1007/s11276-018-1823-z.

[29] Abdul Rashid et al. “Improving Energy Conservation in Wireless Sensor Networks Using Energy Harvesting System”. In: *International Journal of Advanced Computer Science and Applications* 9.1 (1 2018), p. 8.

[30] W.B. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan. “An Application-Specific Protocol Architecture for Wireless Microsensor Networks”. In: *IEEE Transactions on Wireless Communications* 1.4 (Oct. 2002), pp. 660–670. ISSN: 1558-2248. DOI: 10.1109/TWC.2002.804190.

[31] C. Wang et al. “Combining Solar Energy Harvesting with Wireless Charging for Hybrid Wireless Sensor Networks”. In: *IEEE Transactions on Mobile Computing* 17.3 (3 Mar. 2018), pp. 560–576. ISSN: 1558-0660. DOI: 10.1109/TMC.2017.2732979.

[32] J. Huang et al. “A Novel Deployment Scheme for Green Internet of Things”. In: *IEEE Internet of Things Journal* 1.2 (2 Apr. 2014), pp. 196–205. ISSN: 2327-4662. DOI: 10.1109/JIOT.2014.2301819.

[33] K. Wang et al. “Green Industrial Internet of Things Architecture: An Energy-Efficient Perspective”. In: *IEEE Communications Magazine* 54.12 (12 Dec. 2016), pp. 48–54. ISSN: 1558-1896. DOI: 10.1109/MCOM.2016.1600399CM.

[34] Teodora Sanislav et al. “Wireless Energy Harvesting: Empirical Results and Practical Considerations for Internet of Things”. In: *Journal of Network and Computer Applications* 121 (Nov. 1, 2018), pp. 149–158. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2018.08.002.

[35] Alexander J. Williams et al. “Survey of Energy Harvesting Technologies for Wireless Sensor Networks”. In: *IEEE Access* 9 (2021), pp. 77493–77510. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3083697.

[36] Francesco Fraternali et al. “Marble: Collaborative Scheduling of Batteryless Sensors with Meta Reinforcement Learning”. In: *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. BuildSys ’21: The 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation. Coimbra Portugal: ACM, Nov. 17, 2021, pp. 140–149. ISBN: 978-1-4503-9114-6. DOI: 10.1145/3486611.3486670.

[37] P. Tedeschi, S. Sciancalepore, and R. Di Pietro. “Security in Energy Harvesting Networks: A Survey of Current Solutions and Research Challenges”. In: *IEEE Communications Surveys Tutorials* 22.4 (4 Fourthquarter 2020), pp. 2658–2693. ISSN: 1553-877X. DOI: 10.1109/COMST.2020.3017665.

[38] Aditya Singh, Surender Redhu, and Rajesh M. Hegde. “Context-Aware RF-Energy Harvesting for IoT Networks”. In: *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*. 2021 IEEE 7th World Forum on Internet of Things (WF-IoT). June 2021, pp. 77–82. DOI: 10.1109/WF-IoT51360.2021.9595055.

[39] A. Kansal et al. “Harvesting Aware Power Management for Sensor Networks”. In: *2006 43rd ACM/IEEE Design Automation Conference*. 2006 43rd ACM/IEEE Design Automation Conference. July 2006, pp. 651–656. DOI: 10.1145/1146909.1147075.

[40] A. Cammarano, C. Petrioli, and D. Spenza. “Pro-Energy: A Novel Energy Prediction Model for Solar and Wind Energy-Harvesting Wireless Sensor Networks”. In: *2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012)*. 2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012). Oct. 2012, pp. 75–83. DOI: 10.1109/MASS.2012.6502504.

[41] S. Kosunalp. "A New Energy Prediction Algorithm for Energy-Harvesting Wireless Sensor Networks With Q-Learning". In: *IEEE Access* 4 (2016), pp. 5755–5763. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2016.2606541.

[42] Pouya Kamalinejad et al. "Wireless Energy Harvesting for the Internet of Things". In: *IEEE Communications Magazine* 53.6 (6 June 2015), pp. 102–108. ISSN: 0163-6804. DOI: 10.1109/MCOM.2015.7120024.

[43] S. Bi, C. K. Ho, and R. Zhang. "Wireless Powered Communication: Opportunities and Challenges". In: *IEEE Communications Magazine* 53.4 (4 Apr. 2015), pp. 117–125. ISSN: 1558-1896. DOI: 10.1109/MCOM.2015.7081084.

[44] D. Mishra, S. De, and K. R. Chowdhury. "Charging Time Characterization for Wireless RF Energy Transfer". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 62.4 (4 Apr. 2015), pp. 362–366. ISSN: 1558-3791. DOI: 10.1109/TCSII.2014.2387732.

[45] Yanheng Liu et al. "Scheduling Optimization of Charging UAV in Wireless Rechargeable Sensor Networks". In: *2021 IEEE Symposium on Computers and Communications (ISCC)*. 2021 IEEE Symposium on Computers and Communications (ISCC). Sept. 2021, pp. 1–7. DOI: 10.1109/ISCC53001.2021.9631448.

[46] Amar Kaswan, Prasanta K. Jana, and Sajal K. Das. "A Survey on Mobile Charging Techniques in Wireless Rechargeable Sensor Networks". In: *IEEE Communications Surveys & Tutorials* 24.3 (2022), pp. 1750–1779. ISSN: 1553-877X. DOI: 10.1109/COMST.2022.3189387.

[47] Djamel Eddine Kouicem, Abdelmadjid Bouabdallah, and Hicham Lakhlef. "Internet of Things Security: A Top-down Survey". In: *Computer Networks* 141 (Aug. 4, 2018), pp. 199–221. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2018.03.012.

[48] N. Neshenko et al. "Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations". In: *IEEE Communications Surveys Tutorials* 21.3 (3 thirdquarter 2019), pp. 2702–2733. ISSN: 1553-877X. DOI: 10.1109/COMST.2019.2910750.

[49] I. Butun, P. Österberg, and H. Song. "Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures". In: *IEEE Communications Surveys Tutorials* 22.1 (1 Firstquarter 2020), pp. 616–644. ISSN: 1553-877X. DOI: 10.1109/COMST.2019.2953364.

[50] Jie Lin et al. "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications". In: *IEEE Internet of Things Journal* 4.5 (Oct. 2017), pp. 1125–1142. ISSN: 2327-4662. DOI: 10.1109/JIOT.2017.2683200.

[51] Farhana Javed et al. "Internet of Things (IoT) Operating Systems Support, Networking Technologies, Applications, and Challenges: A Comparative Review". In: *IEEE*

Communications Surveys Tutorials 20.3 (2018), pp. 2062–2100. ISSN: 1553-877X. DOI: 10.1109/COMST.2018.2817685.

[52] F. Meneghello et al. "IoT: Internet of Threats? A Survey of Practical Security Vulnerabilities in Real IoT Devices". In: *IEEE Internet of Things Journal* 6.5 (5 Oct. 2019), pp. 8182–8201. ISSN: 2327-4662. DOI: 10.1109/JIOT.2019.2935189.

[53] Luke E. Kane et al. "Security and Performance in IoT: A Balancing Act". In: *IEEE Access* 8 (2020), pp. 121969–121986. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3007536.

[54] prefix=and useprefix=false family=Technology given=National Institute of Standards. *Advanced Encryption Standard (AES)*. Federal Information Processing Standard (FIPS) 197. U.S. Department of Commerce, Nov. 26, 2001. DOI: 10.6028/NIST.FIPS.197.

[55] S. Alharby et al. "The Cost of Link Layer Security in IoT Embedded Devices ∗∗I Wish to Present My Special Thanks to Majmaah University in Saudi Arabia for Their Care and Funding." In: *IFAC-PapersOnLine*. 15th IFAC Conference on Programmable Devices and Embedded Systems PDeS 2018 51.6 (Jan. 1, 2018), pp. 72–77. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2018.07.132.

[56] Sultan Awad N. Alharby. "Practical Adaptive Security for Resource-Constrained IoT Nodes". PhD thesis. University of Southampton, Mar. 2020. 153 pp.

[57] Bomin Mao et al. "Harvesting and Threat Aware Security Configuration Strategy for IEEE 802.15.4 Based IoT Networks". In: *IEEE Communications Letters* 23.11 (Nov. 2019), pp. 2130–2134. ISSN: 1558-2558. DOI: 10.1109/LCOMM.2019.2932988.

[58] Minhaj Ahmad Khan and Khaled Salah. "IoT Security: Review, Blockchain Solutions, and Open Challenges". In: *Future Generation Computer Systems* 82 (May 1, 2018), pp. 395–411. ISSN: 0167-739X. DOI: 10.1016/j.future.2017.11.022.

[59] M. A. Ferrag et al. "Security and Privacy for Green IoT-Based Agriculture: Review, Blockchain Solutions, and Challenges". In: *IEEE Access* 8 (2020), pp. 32031–32053. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2973178.

[60] M. Roopak, G. Yun Tian, and J. Chambers. "Deep Learning Models for Cyber Security in IoT Networks". In: *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC). Jan. 2019, pp. 0452–0457. DOI: 10.1109/CCWC.2019.8666588.

[61] F. Hussain et al. "Machine Learning in IoT Security: Current Solutions and Future Challenges". In: *IEEE Communications Surveys Tutorials* 22.3 (3 thirdquarter 2020), pp. 1686–1721. ISSN: 1553-877X. DOI: 10.1109/COMST.2020.2986444.

[62] Yosra Ben Saied et al. "Trust Management System Design for the Internet of Things: A Context-Aware and Multi-Service Approach". In: *Computers & Security* 39 (Nov. 1, 2013), pp. 351–365. ISSN: 0167-4048. DOI: 10.1016/j.cose.2013.09.001.

[63] Hamed Hellaoui, Abdelmadjid Bouabdallah, and Mouloud Koudil. "TAS-IoT: Trust-Based Adaptive Security in the IoT". In: *2016 IEEE 41st Conference on Local Computer Networks (LCN)*. 2016 IEEE 41st Conference on Local Computer Networks (LCN). Nov. 2016, pp. 599–602. DOI: 10.1109/LCN.2016.101.

[64] Chaimaa Boudagdigue et al. "Trust Management in Industrial Internet of Things". In: *IEEE Transactions on Information Forensics and Security* 15 (2020), pp. 3667–3682. ISSN: 1556-6021. DOI: 10.1109/TIFS.2020.2997179.

[65] Aashma Uprety and Danda B. Rawat. "Reinforcement Learning for IoT Security: A Comprehensive Survey". In: *IEEE Internet of Things Journal* 8.11 (June 2021), pp. 8693–8706. ISSN: 2327-4662. DOI: 10.1109/JIOT.2020.3040957.

[66] Dinh C. Nguyen et al. "Federated Learning for Internet of Things: A Comprehensive Survey". In: *IEEE Communications Surveys & Tutorials* 23.3 (2021), pp. 1622–1658. ISSN: 1553-877X. DOI: 10.1109/COMST.2021.3075439.

[67] Wenxuan Huang, Thanassis Tiropanis, and George Konstantinidis. "Federated Learning-Based IoT Intrusion Detection on Non-IID Data". In: *Internet of Things*. Ed. by Aurora González-Vidal et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2022, pp. 326–337. ISBN: 978-3-031-20936-9. DOI: 10.1007/978-3-031-20936-9_26.

[68] D. Kreutz et al. "Software-Defined Networking: A Comprehensive Survey". In: *Proceedings of the IEEE* 103.1 (Jan. 2015), pp. 14–76. ISSN: 1558-2256. DOI: 10.1109/JPROC.2014.2371999.

[69] Iqbal Alam et al. "A Survey of Network Virtualization Techniques for Internet of Things Using SDN and NFV". In: *ACM Computing Surveys* 53.2 (Apr. 16, 2020), 35:1–35:40. ISSN: 0360-0300. DOI: 10.1145/3379444.

[70] Iman Akbari et al. "ATMoS: Autonomous Threat Mitigation in SDN Using Reinforcement Learning". In: *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*. NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium. Apr. 2020, pp. 1–9. DOI: 10.1109/NOMS47738.2020.9110426.

[71] Alejandro Molina Zarca et al. "Security Management Architecture for NFV/SDN-Aware IoT Systems". In: *IEEE Internet of Things Journal* 6.5 (Oct. 2019), pp. 8005–8020. ISSN: 2327-4662. DOI: 10.1109/JIOT.2019.2904123.

[72] Olivier Flauzac et al. "SDN Based Architecture for IoT and Improvement of the Security". In: *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*. 2015 IEEE 29th International Conference

on Advanced Information Networking and Applications Workshops. Mar. 2015, pp. 688–693. DOI: 10.1109/WAINA.2015.110.

[73] Alessio Di Mauro. "On the Impact of Energy Harvesting on Wireless Sensor Network Security". In: (2015).

[74] Hamed Hellaoui, Mouloud Koudil, and Abdelmadjid Bouabdallah. "Energy-Efficient Mechanisms in Security of the Internet of Things: A Survey". In: *Computer Networks* 127 (Nov. 9, 2017), pp. 173–189. ISSN: 1389-1286. DOI: 10.1016/j.comnet.2017.08.006.

[75] Mohammad Sadegh Yousefpoor et al. "Secure Data Aggregation Methods and Countermeasures against Various Attacks in Wireless Sensor Networks: A Comprehensive Review". In: *Journal of Network and Computer Applications* 190 (Sept. 15, 2021), p. 103118. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2021.103118.

[76] F. Conceição, N. Oualha, and D. Zeghlache. "An Energy Model for the IoT: Secure Networking Perspective". In: *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC). Sept. 2018, pp. 1–5. DOI: 10.1109/PIMRC.2018.8580885.

[77] Ljubomir M. Vračar et al. "Influence of Encryption Algorithms on Power Consumption in Energy Harvesting Systems". In: *Journal of Sensors* 2019 (Apr. 8, 2019). Ed. by Jaime Lloret, p. 8520562. ISSN: 1687-725X. DOI: 10.1155/2019/8520562.

[78] Sudip Maitra and Kumar Yelamarthi. "Rapidly Deployable IoT Architecture with Data Security: Implementation and Experimental Evaluation". In: *Sensors* 19.11 (11 Jan. 2019), p. 2484. DOI: 10.3390/s19112484.

[79] Ehsan Aerabi et al. "Design Space Exploration for Ultra-Low-Energy and Secure IoT MCUs". In: *ACM Transactions on Embedded Computing Systems* 19.3 (May 16, 2020), 19:1–19:34. ISSN: 1539-9087. DOI: 10.1145/3384446.

[80] Sultan Alharby et al. "The Security Trade-Offs in Resource Constrained Nodes for IoT Application". In: *International Journal of Electronics and Communication Engineering* 12.1 (2018), p. 9.

[81] P. Schaumont. "Security in the Internet of Things: A Challenge of Scale". In: *Design, Automation Test in Europe Conference Exhibition (DATE), 2017*. Design, Automation Test in Europe Conference Exhibition (DATE), 2017. Mar. 2017, pp. 674–679. DOI: 10.23919/DATE.2017.7927075.

[82] David J. Wheeler and Roger M. Needham. "TEA, a Tiny Encryption Algorithm". In: *Fast Software Encryption*. International Workshop on Fast Software Encryption. Springer, Berlin, Heidelberg, Dec. 14, 1994, pp. 363–366. DOI: 10.1007/3-540-60590-8_29.

[83] Roger M. Needham and David J. Wheeler. "Tea Extensions". In: *Report, Cambridge University* (1997).

[84] Ernest F Brickell et al. "SKIPJACK Review". In: *Interim Report: The Skipjack Algorithm* (1993).

[85] Benedetto Girgenti et al. "On the Feasibility of Attribute-Based Encryption on Constrained IoT Devices for Smart Systems". In: *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*. 2019 IEEE International Conference on Smart Computing (SMARTCOMP). June 2019, pp. 225–232. DOI: 10.1109/SMARTCOMP.2019.00057.

[86] V. A. Thakor, M. A. Razzaque, and M. R. A. Khandaker. "Lightweight Cryptography Algorithms for Resource-Constrained IoT Devices: A Review, Comparison and Research Opportunities". In: *IEEE Access* 9 (2021), pp. 28177–28193. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3052867.

[87] "NIST Selects 'Lightweight Cryptography' Algorithms to Protect Small Devices". In: *NIST* ().

[88] Dae-Hwi Lee and Im-Yeong Lee. "A Lightweight Authentication and Key Agreement Schemes for IoT Environments". In: *Sensors* 20.18 (18 Jan. 2020), p. 5350. DOI: 10.3390/s20185350.

[89] Byoungjin Seok et al. "Secure D2D Communication for 5G IoT Network Based on Lightweight Cryptography". In: *Applied Sciences* 10.1 (1 Jan. 2020), p. 217. DOI: 10.3390/app10010217.

[90] G. Ateniese et al. "HELIOS: Outsourcing of Security Operations in Green Wireless Sensor Networks". In: *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*. 2017 IEEE 85th Vehicular Technology Conference (VTC Spring). June 2017, pp. 1–7. DOI: 10.1109/VTCSpring.2017.8108500.

[91] K. J. S. R. Kommuru, K. K. Y. Kadari, and B. K. R. Alluri. "A Novel Approach to Balance the Trade-Off between Security and Energy Consumption in WSN". In: *2018 2nd International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE)*. 2018 2nd International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE). Sept. 2018, pp. 85–90. DOI: 10.1109/ICMETE.2018.00030.

[92] Charles Suslowicz, Archanaa S. Krishnan, and Patrick Schaumont. "Optimizing Cryptography in Energy Harvesting Applications". In: *Proceedings of the 2017 Workshop on Attacks and Solutions in Hardware Security*. ASHES '17. New York, NY, USA: Association for Computing Machinery, Nov. 3, 2017, pp. 17–26. ISBN: 978-1-4503-5397-7. DOI: 10.1145/3139324.3139329.

[93] Xiaolin Fang, Ming Yang, and Wenjia Wu. "Security Cost Aware Data Communication in Low-Power IoT Sensors with Energy Harvesting". In: *Sensors* 18.12 (12 Dec. 2018), p. 4400. DOI: 10.3390/s18124400.

[94] Floriano De Rango et al. "Energy-Aware Dynamic Internet of Things Security System Based on Elliptic Curve Cryptography and Message Queue Telemetry Transport Protocol for Mitigating Replay Attacks". In: *Pervasive and Mobile Computing* 61 (Jan. 1, 2020), p. 101105. ISSN: 1574-1192. DOI: 10.1016/j.pmcj.2019.101105.

[95] Bassam Jamil Mohd et al. "Power-Aware Adaptive Encryption". In: *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*. 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT). Apr. 2019, pp. 711–716. DOI: 10.1109/JEEIT.2019.8717426.

[96] A. Yazdinejad et al. "An Energy-Efficient SDN Controller Architecture for IoT Networks With Blockchain-Based Security". In: *IEEE Transactions on Services Computing* 13.4 (4 July 2020), pp. 625–638. ISSN: 1939-1374. DOI: 10.1109/TSC.2020.2966970.

[97] Umer Farooq et al. "Efficient Adaptive Framework for Securing the Internet of Things Devices". In: *EURASIP Journal on Wireless Communications and Networking* 2019.1 (Aug. 27, 2019), p. 210. ISSN: 1687-1499. DOI: 10.1186/s13638-019-1531-0.

[98] H. Hellaoui, M. Koudil, and A. Bouabdallah. "Energy Efficiency in Security of 5G-Based IoT: An End-to-End Adaptive Approach". In: *IEEE Internet of Things Journal* 7.7 (7 July 2020), pp. 6589–6602. ISSN: 2327-4662. DOI: 10.1109/JIOT.2020.2974618.

[99] S. Boudko and H. Abie. "Adaptive Cybersecurity Framework for Healthcare Internet of Things". In: *2019 13th International Symposium on Medical Information and Communication Technology (ISMICT)*. 2019 13th International Symposium on Medical Information and Communication Technology (ISMICT). May 2019, pp. 1–6. DOI: 10.1109/ISMICT.2019.8743905.

[100] Filipe Conceicao. "Network Survival with Energy Harvesting : Secure Cooperation and Device Assisted Networking". PhD thesis. Université Paris Saclay (COmUE), Nov. 29, 2019.

[101] Bizhu Wang, Yan Sun, and Xiaodong Xu. "A Scalable and Energy-Efficient Anomaly Detection Scheme in Wireless SDN-Based mMTC Networks for IoT". In: *IEEE Internet of Things Journal* 8.3 (Feb. 2021), pp. 1388–1405. ISSN: 2327-4662. DOI: 10.1109/JIOT.2020.3011521.

[102] Thaha Mohammed et al. "UbiPriSEQ—Deep Reinforcement Learning to Manage Privacy, Security, Energy, and QoS in 5G IoT HetNets". In: *Applied Sciences* 10.20 (20 Jan. 2020), p. 7120. DOI: 10.3390/app10207120.

[103] B. Chatterjee et al. "Context-Aware Intelligence in Resource-Constrained IoT Nodes: Opportunities and Challenges". In: *IEEE Design Test* 36.2 (Apr. 2019), pp. 7–40. ISSN: 2168-2364. DOI: 10.1109/MDAT.2019.2899334.

[104] Gregory D. Abowd et al. "Towards a Better Understanding of Context and Context-Awareness". In: *Handheld and Ubiquitous Computing*. International Symposium on Handheld and Ubiquitous Computing. Springer, Berlin, Heidelberg, Sept. 27, 1999, pp. 304–307. DOI: 10.1007/3-540-48157-5_29.

[105] M. Loske, L. Rothe, and D. G. Gertler. "Context-Aware Authentication: State-of-the-Art Evaluation and Adaption to the IIoT". In: *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. 2019 IEEE 5th World Forum on Internet of Things (WF-IoT). Apr. 2019, pp. 64–69. DOI: 10.1109/WF-IoT.2019.8767327.

[106] Amit Kumar Sikder et al. "Aegis: A Context-Aware Security Framework for Smart Home Systems". In: *Proceedings of the 35th Annual Computer Security Applications Conference*. ACSAC '19. New York, NY, USA: Association for Computing Machinery, Dec. 9, 2019, pp. 28–41. ISBN: 978-1-4503-7628-0. DOI: 10.1145/3359789.3359840.

[107] Y. Hussain et al. "Context-Aware Trust and Reputation Model for Fog-Based IoT". In: *IEEE Access* 8 (2020), pp. 31622–31632. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2972968.

[108] P. Zhou et al. "Privacy-Preserving and Residential Context-Aware Online Learning for IoT-Enabled Energy Saving With Big Data Support in Smart Home Environment". In: *IEEE Internet of Things Journal* 6.5 (Oct. 2019), pp. 7450–7468. ISSN: 2327-4662. DOI: 10.1109/JIOT.2019.2903341.

[109] Swapnoneel Roy et al. "Modeling Context-Adaptive Energy-Aware Security in Mobile Devices". In: *2018 IEEE 43rd Conference on Local Computer Networks Workshops (LCN Workshops)*. 2018 IEEE 43rd Conference on Local Computer Networks Workshops (LCN Workshops). Oct. 2018, pp. 105–109. DOI: 10.1109/LCNW.2018.8628577.

[110] Asai Asaithambi et al. "Online Context-Adaptive Energy-Aware Security Allocation in Mobile Devices: A Tale of Two Algorithms". In: *Distributed Computing and Internet Technology*. Ed. by Dang Van Hung and Meenakshi D´Souza. Cham: Springer International Publishing, 2020, pp. 281–295. ISBN: 978-3-030-36987-3.

[111] M. A. Massad and B. A. Alsaify. "MQTTSec Based on Context-Aware Cryptographic Selection Algorithm (CASA) for Resource-Constrained IoT Devices". In: *2020 11th International Conference on Information and Communication Systems (ICICS)*.

2020 11th International Conference on Information and Communication Systems (ICICS). Apr. 2020, pp. 349–354. DOI: 10.1109/ICICS49469.2020.239541.

[112] Bomin Mao, Yuichi Kawamoto, and Nei Kato. "AI-Based Joint Optimization of QoS and Security for 6G Energy Harvesting Internet of Things". In: *IEEE Internet of Things Journal* 7.8 (Aug. 2020), pp. 7032–7042. ISSN: 2327-4662. DOI: 10.1109/JIOT.2020.2982417.

[113] J. Recas Piorno et al. "Prediction and Management in Energy Harvested Wireless Sensor Nodes". In: *2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology*. 2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology. May 2009, pp. 6–10. DOI: 10.1109/WIRELESSVITAE.2009.5172412.

[114] Moumita Deb and Sarbani Roy. "Enhanced-Pro: A New Enhanced Solar Energy Harvested Prediction Model for Wireless Sensor Networks". In: *Wireless Personal Communications* 117.2 (2 Mar. 1, 2021), pp. 1103–1121. ISSN: 1572-834X. DOI: 10.1007/s11277-020-07913-y.

[115] Sang-Yoon Chang et al. "Power-Positive Networking: Wireless-Charging-Based Networking to Protect Energy against Battery DoS Attacks". In: *ACM Transactions on Sensor Networks* 15.3 (May 17, 2019), 27:1–27:25. ISSN: 1550-4859. DOI: 10.1145/3317686.

[116] Syeda Manjia Tahsien, Hadis Karimipour, and Petros Spachos. "Machine Learning Based Solutions for Security of Internet of Things (IoT): A Survey". In: *Journal of Network and Computer Applications* 161 (July 1, 2020), p. 102630. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2020.102630.

[117] M. B. Mohamad Noor and W. H. Hassan. "Current Research on Internet of Things (IoT) Security: A Survey". In: *Computer Networks* 148 (2019), pp. 283–294.

[118] R. Yugha and S. Chithra. "A Survey on Technologies and Security Protocols: Reference for Future Generation IoT". In: *Journal of Network and Computer Applications* 169 (Nov. 1, 2020), p. 102763. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2020.102763.

[119] Seyyed Keyvan Mousavi et al. "Security of Internet of Things Based on Cryptographic Algorithms: A Survey". In: *Wireless Networks* 27.2 (2 Feb. 1, 2021), pp. 1515–1555. ISSN: 1572-8196. DOI: 10.1007/s11276-020-02535-5.

[120] Muhammad Moid Sandhu et al. "Towards Energy Positive Sensing Using Kinetic Energy Harvesters". In: *2020 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. 2020 IEEE International Conference on Pervasive Computing and Communications (PerCom). Mar. 2020, pp. 1–10. DOI: 10.1109/PerCom45495.2020.9127356.

[121] F. K. Shaikh, S. Zeadally, and E. Exposito. "Enabling Technologies for Green Internet of Things". In: *IEEE Systems Journal* 11.2 (2 June 2017), pp. 983–994. ISSN: 1937-9234. DOI: 10.1109/JSYST.2015.2415194.

[122] D. Chen et al. "Energy-Efficient Secure Transmission Design for the Internet of Things With an Untrusted Relay". In: *IEEE Access* 6 (2018), pp. 11862–11870. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2805818.

[123] Kisong Lee, Jun-Pyo Hong, and Woongsup Lee. "Deep Learning Framework for Secure Communication With an Energy Harvesting Receiver". In: *IEEE Transactions on Vehicular Technology* 70.10 (Oct. 2021), pp. 10121–10132. ISSN: 1939-9359. DOI: 10.1109/TVT.2021.3103521.

[124] Meltem Sonmez Turan et al. *Status Report on the Second Round of the NIST Lightweight Cryptography Standardization Process*. National Institute of Standards and Technology, July 20, 2021. DOI: 10.6028/NIST.IR.8369.

[125] Eva García-Martín et al. "Estimation of Energy Consumption in Machine Learning". In: *Journal of Parallel and Distributed Computing* 134 (Dec. 1, 2019), pp. 75–88. ISSN: 0743-7315. DOI: 10.1016/j.jpdc.2019.07.007.

[126] Azzam Mourad et al. "Ad Hoc Vehicular Fog Enabling Cooperative Low-Latency Intrusion Detection". In: *IEEE Internet of Things Journal* 8.2 (Jan. 2021), pp. 829–843. ISSN: 2327-4662. DOI: 10.1109/JIOT.2020.3008488.

[127] Yinxin Wan et al. "IoTArgos: A Multi-Layer Security Monitoring System for Internet-of-Things in Smart Homes". In: *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*. IEEE INFOCOM 2020 - IEEE Conference on Computer Communications. July 2020, pp. 874–883. DOI: 10.1109/INFOCOM41043.2020.9155424.

[128] R. V. Kulkarni and G. K. Venayagamoorthy. "Neural Network Based Secure Media Access Control Protocol for Wireless Sensor Networks". In: *2009 International Joint Conference on Neural Networks*. 2009 International Joint Conference on Neural Networks. June 2009, pp. 1680–1687. DOI: 10.1109/IJCNN.2009.5179075.

[129] Yanqing Yang et al. "Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational AutoEncoder and Deep Neural Network". In: *Sensors* 19.11 (11 Jan. 2019), p. 2528. DOI: 10.3390/s19112528.

[130] Mustafizur R. Shahid et al. "Anomalous Communications Detection in IoT Networks Using Sparse Autoencoders". In: *2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)*. 2019 IEEE 18th International Symposium on Network Computing and Applications (NCA). Sept. 2019, pp. 1–5. DOI: 10.1109/NCA.2019.8935007.

[131] Ly Vu et al. "Deep Transfer Learning for IoT Attack Detection". In: *IEEE Access* 8 (2020), pp. 107335–107344. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3000476.

[132] Yulin Fan et al. “IoTDefender: A Federated Transfer Learning Intrusion Detection Framework for 5G IoT”. In: *2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE)*. 2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE). Dec. 2020, pp. 88–95. DOI: 10.1109/BigDataSE50710.2020.00020.

[133] Xiaopeng Tan et al. “Intrusion Detection of UAVs Based on the Deep Belief Network Optimized by PSO”. In: *Sensors* 19.24 (24 Jan. 2019), p. 5529. DOI: 10.3390/s19245529.

[134] S. Tu et al. “Security in Fog Computing: A Novel Technique to Tackle an Impersonation Attack”. In: *IEEE Access* 6 (2018), pp. 74993–75001. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2884672.

[135] Woongsoo Na et al. “Energy-Efficient Mobile Charging for Wireless Power Transfer in Internet of Things Networks”. In: *IEEE Internet of Things Journal* 5.1 (Feb. 2018), pp. 79–92. ISSN: 2327-4662. DOI: 10.1109/JIOT.2017.2772318.

[136] Khaled Abid et al. “An Energy Efficient Architecture of Self-Sustainable WSN Based on Energy Harvesting and Wireless Charging with Consideration of Deployment Cost”. In: *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*. Q2SWinet ’20. New York, NY, USA: Association for Computing Machinery, Nov. 16, 2020, pp. 109–114. ISBN: 978-1-4503-8120-8. DOI: 10.1145/3416013.3426450.

[137] Niayesh Gharaei et al. “Energy-Efficient Tour Optimization of Wireless Mobile Chargers for Rechargeable Sensor Networks”. In: *IEEE Systems Journal* 15.1 (Mar. 2021), pp. 27–36. ISSN: 1937-9234. DOI: 10.1109/JSYST.2020.2968968.

[138] M. Min et al. “Learning-Based Computation Offloading for IoT Devices With Energy Harvesting”. In: *IEEE Transactions on Vehicular Technology* 68.2 (2 Feb. 2019), pp. 1930–1941. ISSN: 1939-9359. DOI: 10.1109/TVT.2018.2890685.

[139] Jiayu Li, Ji Hoon Hyun, and Dong SamHa. “A Multi-Source Energy Harvesting System to Power Microcontrollers for Cryptography”. In: *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*. IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society. Oct. 2018, pp. 901–906. DOI: 10.1109/IECON.2018.8591833.

[140] Sultan Alharby et al. “Impact of Duty Cycle Protocols on Security Cost of IoT”. In: *2018 9th International Conference on Information and Communication Systems (ICICS)*. 2018 9th International Conference on Information and Communication Systems (ICICS). Apr. 2018, pp. 25–30. DOI: 10.1109/IACS.2018.8355436.

[141] Danda B. Rawat and Swetha R. Reddy. “Software Defined Networking Architecture, Security and Energy Efficiency: A Survey”. In: *IEEE Communications Surveys*

Tutorials 19.1 (2017), pp. 325–346. ISSN: 1553-877X. DOI: 10.1109/COMST.2016.2618874.

[142] Xing Liu et al. "Toward Deep Transfer Learning in Industrial Internet of Things". In: *IEEE Internet of Things Journal* 8.15 (Aug. 2021), pp. 12163–12175. ISSN: 2327-4662. DOI: 10.1109/JIOT.2021.3062482.

[143] Fangyu Li et al. "Enhanced Cyber-Physical Security in Internet of Things Through Energy Auditing". In: *IEEE Internet of Things Journal* 6.3 (June 2019), pp. 5224–5231. ISSN: 2327-4662. DOI: 10.1109/JIOT.2019.2899492.

[144] Michele De Donno et al. "Sustainable Security for Internet of Things". In: *2019 International Conference on Smart Applications, Communications and Networking (SmartNets)*. 2019 International Conference on Smart Applications, Communications and Networking (SmartNets). Dec. 2019, pp. 1–4. DOI: 10.1109/SmartNets48225.2019.9069776.

[145] Mohammed H. Alsharif, Sunghwan Kim, and Nuri Kuruoğlu. "Energy Harvesting Techniques for Wireless Sensor Networks/Radio-Frequency Identification: A Review". In: *Symmetry* 11.7 (7 July 2019), p. 865. DOI: 10.3390/sym11070865.

[146] Tharindu D. Ponnimbaduge Perera et al. "Simultaneous Wireless Information and Power Transfer (SWIPT): Recent Advances and Future Challenges". In: *IEEE Communications Surveys Tutorials* 20.1 (2018), pp. 264–302. ISSN: 1553-877X. DOI: 10.1109/COMST.2017.2783901.

[147] Michal Prauzek et al. "Energy Harvesting Sources, Storage Devices and System Topologies for Environmental Wireless Sensor Networks: A Review". In: *Sensors* 18.8 (8 Aug. 2018), p. 2446. ISSN: 1424-8220. DOI: 10.3390/s18082446.

[148] Meiyi Yang et al. "Dynamic Charging Scheme Problem With Actor–Critic Reinforcement Learning". In: *IEEE Internet of Things Journal* 8.1 (Jan. 2021), pp. 370–380. ISSN: 2327-4662. DOI: 10.1109/JIOT.2020.3005598.

[149] Shuai Zhang, Weiqi Liu, and Nirwan Ansari. "Joint Wireless Charging and Data Collection for UAV-Enabled Internet of Things Network". In: *IEEE Internet of Things Journal* 9.23 (Dec. 2022), pp. 23852–23859. ISSN: 2327-4662. DOI: 10.1109/JIOT.2022.3190813.

[150] Volodymyr Mnih et al. "Human-Level Control through Deep Reinforcement Learning". In: *Nature* 518.7540 (Feb. 26, 2015), pp. 529–533. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature14236.

[151] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Red. by Francis Bach. 2nd ed. Adaptive Computation and Machine Learning Series. Cambridge, MA, USA: A Bradford Book, Nov. 13, 2018. 552 pp. ISBN: 978-0-262-03924-6.

[152] Charles Desjardins and Brahim Chaib-draa. "Cooperative Adaptive Cruise Control: A Reinforcement Learning Approach". In: *IEEE Transactions on Intelligent Transportation Systems* 12.4 (Dec. 2011), pp. 1248–1260. ISSN: 1558-0016. DOI: 10.1109/TITS.2011.2157145.

[153] Kai Arulkumaran et al. "A Brief Survey of Deep Reinforcement Learning". In: *IEEE Signal Processing Magazine* 34.6 (Nov. 2017), pp. 26–38. ISSN: 1053-5888. DOI: 10.1109/MSP.2017.2743240. arXiv: 1708.05866 [cs, stat].

[154] Lei Lei et al. "Deep Reinforcement Learning for Autonomous Internet of Things: Model, Applications and Challenges". In: *IEEE Communications Surveys Tutorials* 22.3 (thirdquarter 2020), pp. 1722–1760. ISSN: 1553-877X. DOI: 10.1109/COMST.2020.2988367.

[155] Mohamed Said Frikha et al. "Reinforcement and Deep Reinforcement Learning for Wireless Internet of Things: A Survey". In: *Computer Communications* 178 (Oct. 1, 2021), pp. 98–113. ISSN: 0140-3664. DOI: 10.1016/j.comcom.2021.07.014.

[156] Wuhui Chen et al. "Deep Reinforcement Learning for Internet of Things: A Comprehensive Survey". In: *IEEE Communications Surveys & Tutorials* 23.3 (2021), pp. 1659–1692. ISSN: 1553-877X. DOI: 10.1109/COMST.2021.3073036.

[157] Christopher J. C. H. Watkins and Peter Dayan. "Q-Learning". In: *Machine Learning* 8.3 (May 1, 1992), pp. 279–292. ISSN: 1573-0565. DOI: 10.1007/BF00992698.

[158] Kurt Hornik. "Approximation Capabilities of Multilayer Feedforward Networks". In: *Neural Networks* 4.2 (Jan. 1, 1991), pp. 251–257. ISSN: 0893-6080. DOI: 10.1016/0893-6080(91)90009-T.

[159] David Silver et al. "Mastering the Game of Go without Human Knowledge". In: *Nature* 550.7676 (2017), pp. 354–359. ISSN: 0028-0836. DOI: 10.1038/nature24270.

[160] Abdulmajid Murad et al. "Autonomous Management of Energy-Harvesting IoT Nodes Using Deep Reinforcement Learning". In: *2019 IEEE 13th International Conference on Self-Adaptive and Self-Organizing Systems (SASO)*. 2019 IEEE 13th International Conference on Self-Adaptive and Self-Organizing Systems (SASO). June 2019, pp. 43–51. DOI: 10.1109/SASO.2019.00015.

[161] Sana Benhamaid, Hicham Lakhlef, and Abdelmadjid Bouabdallah. "Energy-Efficient and Context-aware Trajectory Planning for Mobile Data Collection in IoT Using Deep Reinforcement Learning". In: *2022 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. 2022 International Conference on Software, Telecommunications and Computer Networks (SoftCOM). Sept. 2022, pp. 1–6. DOI: 10.23919/SoftCOM55329.2022.9911304.

[162] Zhenchun Wei et al. "Reinforcement Learning for a Novel Mobile Charging Strategy in Wireless Rechargeable Sensor Networks". In: *Wireless Algorithms, Systems, and Applications*. International Conference on Wireless Algorithms, Systems, and

Applications. Springer, Cham, June 20, 2018, pp. 485–496. DOI: 10.1007/978-3-319-94268-1_40.

[163] La Van Quan et al. "Q-Learning-Based, Optimized On-demand Charging Algorithm in WRSN". In: *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*. 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA). Nov. 2020, pp. 1–8. DOI: 10.1109/NCA51143.2020.9306695.

[164] Xianbo Cao et al. "A Deep Reinforcement Learning-Based on-Demand Charging Algorithm for Wireless Rechargeable Sensor Networks". In: *Ad Hoc Networks* 110 (Jan. 1, 2021), p. 102278. ISSN: 1570-8705. DOI: 10.1016/j.adhoc.2020.102278.

[165] Ngoc Bui et al. "A Deep Reinforcement Learning-based Adaptive Charging Policy for WRSNs". In: *2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*. 2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS). Oct. 2022, pp. 661–667. DOI: 10.1109/MASS56207.2022.00097.

[166] Abhinav Tomar and Prasanta K. Jana. "A Multi-Attribute Decision Making Approach for on-Demand Charging Scheduling in Wireless Rechargeable Sensor Networks". In: *Computing* 103.8 (8 Aug. 1, 2021), pp. 1677–1701. ISSN: 1436-5057. DOI: 10.1007/s00607-020-00875-w.

[167] Valentin Cristea, Ciprian Dobre, and Florin Pop. "Context-Aware Environments for the Internet of Things". In: *Internet of Things and Inter-cooperative Computational Technologies for Collective Intelligence* (2013), pp. 25–49. DOI: 10.1007/978-3-642-34952-2_2.

[168] Charith Perera et al. "Context Aware Computing for The Internet of Things: A Survey". In: *IEEE Communications Surveys Tutorials* 16.1 (2014), pp. 414–454. ISSN: 1553-877X. DOI: 10.1109/SURV.2013.042313.00197.

[169] A. S. Rao, A. V. Sharma, and C. S. Narayan. "A Context Aware System for an IoT-based Smart Museum". In: *2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*. 2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech). July 2017, pp. 1–5.

[170] Yong Ding, Hedda R. Schmidtke, and Michael Beigl. "Beyond Context-Awareness: Context Prediction in an Industrial Application". In: *Proceedings of the 12th ACM International Conference Adjunct Papers on Ubiquitous Computing - Ubicomp '10*. The 12th ACM International Conference Adjunct Papers. Copenhagen, Denmark: ACM Press, 2010, p. 401. ISBN: 978-1-4503-0283-8. DOI: 10.1145/1864431.1864457.

[171] Rodrigo Schmitt Meurer, Antônio Augusto Fröhlich, and Jomi Fred Hübner. "Ambient Intelligence for the Internet of Things Through Context-Awareness". In: *2018 International Symposium on Rapid System Prototyping (RSP)*. 2018 International

Symposium on Rapid System Prototyping (RSP). Oct. 2018, pp. 83–89. DOI: 10.1109/RSP.2018.8631989.

[172] Mariana Rodrigues, Daniel F. Pigatto, and Kalinka R. L. J. C. Branco. "Context-Aware Operation for Unmanned Systems with HAMSTER". In: *2020 IEEE Symposium on Computers and Communications (ISCC)*. 2020 IEEE Symposium on Computers and Communications (ISCC). July 2020, pp. 1–6. DOI: 10.1109/ISCC50000.2020.9219657.

[173] Peter J. Huber. "Robust Estimation of a Location Parameter". In: *Breakthroughs in Statistics: Methodology and Distribution*. Ed. by Samuel Kotz and Norman L. Johnson. Springer Series in Statistics. New York, NY: Springer, 1992, pp. 492–518. ISBN: 978-1-4612-4380-9. DOI: 10.1007/978-1-4612-4380-9_35.

[174] Bruno Bogaz Zarpelão et al. "A Survey of Intrusion Detection in Internet of Things". In: *Journal of Network and Computer Applications* 84 (Apr. 15, 2017), pp. 25–37. ISSN: 1084-8045. DOI: 10.1016/j.jnca.2017.02.009.

[175] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. "Deep Sparse Rectifier Neural Networks". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings, June 14, 2011, pp. 315–323.

[176] Greg Brockman et al. *OpenAI Gym*. June 5, 2016. DOI: 10.48550/arXiv.1606.01540. arXiv: 1606.01540 [cs]. (Visited on 02/09/2023). preprint.

[177] Antonin Raffin et al. "Stable-Baselines3: Reliable Reinforcement Learning Implementations". In: *Journal of Machine Learning Research* 22.268 (2021), pp. 1–8.

[178] Rishabh Agarwal et al. "Deep Reinforcement Learning at the Edge of the Statistical Precipice". In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 29304–29320.

[179] Peter Henderson et al. "Deep Reinforcement Learning That Matters". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (1 Apr. 29, 2018). ISSN: 2374-3468. DOI: 10.1609/aaai.v32i1.11694.