People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Chadli Bendjedid University
Faculty of Sciences and Technology
Department of Computer Science

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة الشاذلي بن جديد
كلية العلوم و التكنولوجيا
قسم الاعلام الالي

جامعة الشاذلي بن جديد
UNIVERSITE CHADLI BENDJEDID

THESIS

Presented by

DAHMANI RAYANE

In partial fulfillment of the requirements for

MASTER

Field : Informatique

Specialty: Intelligent Computer Systems

Title

An Intelligent Security Architecture Based on Deep Q-Learning (DQN) for the Detection and Prevention of Botnet Attacks in IoT Healthcare Devices within an Edge Computing Environment.

Defended on: 15/06/ 2026

Before the Jury composed of:

Role	Full Name	Rank	University
President	Mrs. Majda Maatallah	MCB	Chadli Bendjedid El-Tarf
Supervisor	**Mrs. Makhlouf Amina**	**MCA**	**Chadli Bendjedid El-Tarf**
Co-Supervisor	Miss. Labiod Yasmine	MCB	Badji Mokhtar Annaba
Examinator	Mr. Betouil Ali Abdlatif	MCB	Chadli Bendjedid El-Tarf

Academic Year : 2025/2026

Acknowledgements

First and foremost, we thank Allah Almighty for granting us the health, strength, and determination to accomplish this work.

We would like to express our sincere gratitude to our supervisor, ***Mrs. Makhlouf Amina****, for her valuable guidance, continuous support, and availability throughout the completion of this thesis.*

We also extend our thanks to our co-supervisor, ***Miss. Labiod Yasmine****, for her assistance, insightful advice, and encouragement, which greatly contributed to the success of this work.*

Our sincere appreciation is also extended to the members of the jury for accepting to evaluate this thesis and for their valuable remarks and observations.

Finally, we would like to express our deepest gratitude to our beloved parents for their unconditional love, endless sacrifices, and constant prayers, which have guided us throughout our academic journey. We also thank our families for their unwavering support and encouragement during our university years.

Dedication

*To my dear sister, **Chahira**, whose unwavering support and encouragement have always been a source of motivation.*

*To my cousin, **Noura**, for her presence, valuable assistance, and confidence in me.*

*To my best friend, **Ikram**, my partner throughout this project, for her collaboration, dedication, and all the efforts we shared during this work.*

May this modest work stand as a testament to my sincere appreciation and profound gratitude to them.

Abstract

Securing connected medical infrastructures (IoMT) is a critical challenge given the sensitivity of health data and the direct risks to care continuity. This thesis proposes an intelligent and distributed security architecture, structured according to a three-tier hierarchical topology Edge, Fog, and Cloud for botnet attack detection and prevention in IoMT networks, within an Edge Computing context.

The approach is based on Deep Q-Learning (DQN) and systematically compares four algorithmic variants: the standard DQN, Double DQN, Dueling DQN, and DQN with Prioritized Experience Replay (PER). An adaptive medical rules engine, comprising eight expert rules hierarchized by clinical severity, is integrated into the reinforcement learning process to ensure institutional compliance of the agent's decisions. Experimental validation is conducted on the CIC-IoMT-2024 dataset through 28 systematic configurations per variant. Results demonstrate that the Dueling DQN offers the best trade-off with an F1-score of 0.9981, a false positive rate of 1.49%, and a latency of 0.28 ms, fully meeting the Edge layer real-time constraint (≤ 5 ms). All variants exhibit a margin factor exceeding ten relative to this constraint, validating the operational feasibility of deploying lightweight DRL systems on resource-limited nodes.

Keywords: *IoMT, e-health, intrusion detection, Deep Q-Learning, Dueling DQN, Double DQN, PER, Edge Computing, botnet, deep reinforcement learning, CIC-IoMT-2024.*

Abstract (French)

La sécurisation des infrastructures médicales connectées (IoMT) constitue un enjeu critique en raison de la sensibilité des données de santé et des risques directs sur la continuité des soins. Ce mémoire propose une architecture de sécurité intelligente et distribuée, structurée selon une topologie hiérarchique à trois niveaux Edge, Fog et Cloud pour la détection et la prévention des attaques de botnets dans les réseaux IoMT, en contexte Edge Computing. L'approche repose sur le Deep Q-Learning (DQN) et compare de manière systématique quatre variantes algorithmiques : le DQN standard, le Double DQN, le Dueling DQN et le DQN avec Prioritized Experience Replay (PER). Un moteur de règles médicales adaptatif, composé de huit règles expertes hiérarchisées par sévérité clinique, est intégré au processus d'apprentissage par renforcement afin d'assurer la conformité institutionnelle des décisions prises par l'agent. La validation expérimentale est conduite sur le jeu de données CIC-IoMT-2024, à travers 28 configurations systématiques par variante. Les résultats démontrent que le Dueling DQN offre le meilleur compromis avec un F1-score de 0,9981, un taux de faux positifs de 1,49 % et une latence de 0,28 ms, respectant pleinement la contrainte temps réel de la couche Edge (≤ 5 ms). L'ensemble des variantes présente un facteur de marge supérieur à dix par rapport à cette contrainte, validant la

faisabilité opérationnelle du déploiement de systèmes DRL légers sur des nœuds à ressources limitées.

Mot cles : IoMT, e-health, détection d'intrusions, Deep Q-Learning, Dueling DQN, Double DQN, PER, Edge Computing, botnet, apprentissage par renforcement profond, CIC-IoMT-2024.

Abstract(Arabic)

يُعدّ تأمين البنى التحتية الطبية المتصلة (IoMT) تحدياً بالغ الأهمية، نظراً لحساسية بيانات الصحة والمخاطر المباشرة على استمرارية الرعاية .يقترح هذا المذكرة معماريةً أمنيةً ذكيةً وموزعة، مُهيكَلة وفق طوبولوجيا هرمية ثلاثية المستوياتالحافة (Edge)، والضباب(Fog) ، والسحابة (Cloud) لاكتشاف هجمات البوت نت والحدّ منها في شبكاتIoMT ، في سياق الحوسبة على الحافة. يرتكز النهج المقترح على التعلم العميق(DQN) Q ، ويُجري مقارنةً منهجيةً بين أربع نسخ خوارزمية : DQNالقياسي، وDouble DQN، وDueling DQN، و DQNمع إعادة التشغيل ذات الأولوية .(PER) ويُدمج محركُ قواعد طبية تكيُّفي، مؤلَّف من ثماني قواعد خبيرة مُرتَّبة وفق الخطورة السريرية، في عملية التعلم المعزَّز لضمان امتثال قرارات الوكيل للمعايير المؤسسية.جرى التحقق التجريبي على مجموعة البيانات CIC-IoMT-2024 من خلال 28 تكويناً منهجياً لكل نسخة .تُظهر النتائج أن Dueling DQN يُقدّم أفضل توازن بمعيار F1 يبلغ0.9981 ، ومعدل إيجابيات كاذبة يبلغ 1.49%، وزمن استجابة يبلغ 0.28 مللي ثانية، مع احترام تام لقيد الوقت الحقيقي لطبقة الحافة 5 ≤) مللي ثانية .(وتُحقق جميع النسخ هامش أمان يتجاوز عشرة أضعاف هذا القيد، مما يُثبت الجدوى التشغيلية لنشر أنظمة DRL خفيفة على عُقد محدودة الموارد.

الكلمات المفتاحية: *IoMT، الصحة الإلكترونية، كشف التسلل، التعلم العميقQ، Dueling DQN، Double DQN، PER ، الحوسبة على الحافة، البوت نت، التعلم المعزز العميق، CIC-IoMT-2024*

Table of Contents

List of Figures

List of Tables

List of Acronyms

ACK	Acknowledgment
ARP	Address Resolution Protocol
AUC	Area Under the Curve
CNN	Convolutional Neural Network
CIC	Canadian Institute for Cybersecurity
C&C	Command and Control
DDPG	Deep Deterministic Policy Gradient
DDoS	Distributed Denial of Service
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DoS	Denial of Service
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
ECE	Explicit Congestion Notification Echo
FIN	Finish
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
GPU	Graphics Processing Unit
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IAT	Inter-Arrival Time
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IGMP	Internet Group Management Protocol
IoMT	Internet of Medical Things
IoT	Internet of Things
IP	Internet Protocol
IRC	Internet Relay Chat
k-NN	k-Nearest Neighbors
LLC	Logical Link Control
LSTM	Long Short-Term Memory
MDP	Markov Decision Process
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
NaN	Not a Number
NSL-KDD	New NSL Knowledge Discovery Dataset
PER	Prioritized Experience Replay
PR	Precision-Recall
PSH	Push
RAM	Random Access Memory
ReLU	Rectified Linear Unit
RL	Reinforcement Learning
ROC	Receiver Operating Characteristic
RST	Reset
SMOTE	Synthetic Minority Over-sampling Technique
SSH	Secure Shell
SMTP	Simple Mail Transfer Protocol

SYN	Synchronize
TCP	Transmission Control Protocol
TD	Temporal Difference
Telnet	Teletype Network
TN	True Negative
TP	True Positive
TPR	True Positive Rate
TTL	Time to Live
UDP	User Datagram Protocol
UNSW-NB15	University of New South Wales dataset 15

General Introduction

Contemporary healthcare infrastructures owe their digital transformation to the growing adoption of network-connected objects equipped with embedded computing capabilities. These devices enable continuous capture, processing, and delivery of clinical information, resulting in tangible gains across three dimensions: the precision of patient monitoring, the responsiveness of diagnostic procedures, and the overall efficiency level achieved by care institutions. This progress comes, however, with a counterpart: the attack surface exposed to malicious intrusions is correspondingly enlarged, making medical systems more vulnerable to adversaries whose methods are characterized by ever-increasing complexity and adaptability. It is within this configuration that the Internet of Medical Things (IoMT) finds its full significance. This sector-specific extension of the IoT is dedicated to clinical applications where data reliability and protection are vital. The resulting ecosystem comprises a multitude of connected care devices physiological sensors, AI-integrated biomedical instruments, and remote monitoring systems all contributing to the continuous transmission of highly sensitive information streams relating to the condition of patients under care. Nevertheless, the design of these objects faces severe hardware restrictions: limited computing capacity, restricted memory, energy constraints, and difficulties in deploying software updates all of which obstruct the implementation of sophisticated defensive mechanisms. These structural fragilities predispose IoMT networks to particularly formidable forms of intrusion, chief among which are botnets. Their operational principle consists in silently infecting a large number of devices to form a fleet of compromised machines capable of launching coordinated actions: distributed denial-of-service attacks (DDoS), exfiltration of confidential information, and dissemination of malicious programs. The Mirai botnet episode sufficiently illustrated the severity of potential damage when a large-scale connected infrastructure is subverted. Against polymorphic attacks that are perpetually reinvented, classical intrusion detection approaches based on fixed signatures quickly reach their limits as the following chapter will analyze in detail. Moreover, centralized cloud-hosted defense architectures suffer from prohibitive latency, scalability, and privacy concerns, making them poorly suited to the strict temporal requirements of the hospital environment. These difficulties find a potential response in Edge Computing, which advocates localizing analytical work in the immediate vicinity of emitting devices. Three major benefits follow: reduced latency, increased responsiveness, and limitation of personal data transfers to remote infrastructures. Nevertheless, having an edge infrastructure is not sufficient to resolve the fundamental problem. Attackers continuously renew their tactics, which demands constant adaptability from defensive systems in other words, the ability to learn in environments whose rules change over time. It is precisely in this niche that Deep Reinforcement Learning (DRL) makes an original contribution. Within this family, the Deep Q-Network (DQN) algorithm proves particularly well-suited to protecting IoMT ecosystems. Unlike conventional supervised approaches, DRL allows an agent to acquire a behavioral strategy through direct interaction with its environment, without relying on a pre-annotated dataset a property that makes it singularly adapted to the polymorphic and continuously evolving threats inherent to IoMT ecosystems. The work presented here describes an intelligent defensive architecture based on Deep Q-Learning (DQN), deployed according to a three-tier distributed topology (Edge–Fog–Cloud) for the identification and blocking of botnets within IoMT networks. In this configuration, lightweight agents are positioned at the Edge tier to conduct local, instantaneous inspection of network flows. Their learning process is governed by a multi-objective reward function that jointly values detection accuracy, decision speed, and

compliance with institutional security rules. As a complementary mechanism, a procedural filtering rule generation engine reinforces preventive capabilities and proactive response against hostile actions. The overall aim of this proposal is to provide a distributed, self-adaptive, and scalable solution capable of meeting the security, performance, and personal data protection requirements of IoMT environments where reliability demands are at their highest.

1. Problem Statement

The massive deployment of connected objects in health institutions has profoundly changed the cyber-risk landscape in which these organizations are exposed to. Within IoMT architectures, the proliferation of sensors, AI-embedded biomedical devices, and remote monitoring solutions generates significant technical heterogeneity, compounded by notable structural limitations scarcity of computing resources, limited memory capacity, and cumbersome software update processes. Such characteristics render classical defensive strategies either difficult to implement or greatly diminished in effectiveness within this specific context. Among recent digital threats, botnets represent a particularly concerning source of danger for digital medicine infrastructures. By exploiting the inherent vulnerabilities of connected devices, they enable the construction of decentralized fleets of infected devices capable of executing coordinated actions notably DDoS attacks, exfiltration of sensitive medical data, and operations aimed at deliberately degrading medical service operations. The potential consequences of such intrusions extend beyond the IT domain alone and may jeopardize care continuity as well as the physical integrity of patients. Faced with the constant evolution of these threats, conventional detection mechanisms whether based on fixed signatures or predefined rules prove structurally inadequate: their dependence on a priori knowledge of hostile behaviors compromises their ability to identify novel variants or evolving tactics. Furthermore, centralized cloud-hosted security architectures introduce additional difficulties in terms of response time, bandwidth consumption, and privacy, which are difficult to reconcile with the reactivity requirements and regulatory obligations of the medical world. Against this backdrop, a central question arises: how can a distributed defensive architecture be designed that is capable of self-adaptively detecting and neutralizing botnets evolving within IoMT networks, while remaining compatible with the low-latency, constrained-resource, and sensitive-data-preservation requirements that characterize e-health environments? This question underpins several major scientific challenges:

- Designing a detection function capable of continuous learning within a non-stationary environment.
- Ensuring localized and rapid decision-making at the Edge tier without sacrificing identification accuracy.
- Maintaining a balance between network performance, operational security effectiveness, and compliance with institutional guidelines.
- Guaranteeing the scalability of the solution in the face of the constant growth of the IoMT device pool.
- Addressing the lack of labeled datasets representative of real IoMT environments.

Resolving this problem calls for the exploration of adaptive paradigms such as Deep Reinforcement Learning, integrated within a hierarchical, distributed Edge–Fog–Cloud architecture, so as to reconcile protection requirements, technical limitations, and clinical necessities.

2. Motivations

Protecting IoMT ecosystems has now become a top strategic priority for e-health infrastructures, given the highly sensitive nature of the information processed and the direct impact that a compromise can have on the continuity of clinical services. Although the scientific literature abounds with proposals for intrusion detection in IoT networks, their transposition to medical contexts faces constraining specificities: criticality of services rendered, a particularly demanding regulatory framework, and great heterogeneity of connected devices. Conventional supervised learning methodologies rely primarily on pre-annotated datasets and models parameterized offline. While they demonstrate a certain effectiveness in relatively static environments, these approaches suffer from limited adjustment capacity when confronted with evolving threats such as contemporary botnets. The emergence of a new attack pattern generally requires a complete retraining phase, which penalizes their reactivity in dynamic and non-stationary contexts. Moreover, the scarcity of labeled datasets representative of real IoMT environments constitutes a further obstacle to their generalization in this domain. In this context, Deep Reinforcement Learning (DRL) emerges as a particularly promising alternative. Unlike supervised paradigms, DRL allows an agent to acquire an optimal behavioral policy through iterative interaction with its environment, by maximizing a reward function defined according to operational objectives without requiring pre-labeled data. This capacity for dynamic adaptation proves singularly relevant for the identification and neutralization of polymorphic and evolving attacks, such as IoMT botnets. Nevertheless, despite the growing interest in Deep Q-Network (DQN)-based approaches in the IoT cybersecurity field, several gaps persist in the scientific literature. Few studies conduct a systematic comparative examination of the major DQN variants : standard DQN, Double DQN, Dueling DQN, and PER in a framework specifically dedicated to IoMT. Furthermore, the simultaneous integration of an adaptive learning mechanism and a distributed Edge–Fog–Cloud architecture remains insufficiently explored. The motivation of this work therefore lies in the desire to address these shortcomings by proposing an intelligent, distributed, and self-adaptive security architecture specifically designed for constrained IoMT environments leveraging the learning capabilities of Deep Q-Learning while exploiting the Edge Computing paradigm to reconcile real-time detection, system scalability, and preservation of sensitive data.

3. Objectives

The primary aim of this research is to develop an autonomous and distributed cybernetic protection mechanism, built upon the theoretical foundations of Deep Q-Learning, whose architecture is tailored to the characteristics of botnet-orchestrated intrusion campaigns in IoMT environments. The projected system must reconcile three fundamental constraints: extreme reactivity, measured in terms of decision latency; operation confined to the computing resources available on edge devices; and strict preservation of the private nature of clinical data circulating throughout all phases of the detection process.

General Objective

The primary objective is to define a botnet detection and response architecture in an IoMT context, leveraging Deep Q-Learning mechanisms and organized according to an Edge–Fog–Cloud hierarchical stratification, so as to provide self-regulating protection operating continuously and in real time.

Specific Objectives

- ✓ **Analysis of the State of the Art:** Review of existing work: Conduct a state-of-the-art review of intrusion detection methods in IoT and IoMT environments, highlighting their current limitations: lack of adaptability, scalability difficulties, and insufficient real-time performance.
- ✓ **Design of a Distributed Architecture:** Propose a three-tier (Edge–Fog–Cloud) architecture to efficiently distribute workloads related to monitoring, diagnosis, and supervision, with the objective of constraining latency and network congestion.
- ✓ **Development of a Deep Q-Learning Agent:** An autonomous agent based on Deep Q-Learning is developed to learn the detection policy through continuous interaction with the environment, thereby reducing dependence on manually labeled training datasets.
- ✓ **Multi-Objective Reward Function Optimization:** A composite reward function is designed to simultaneously consider multiple performance objectives, including threat detection accuracy, reduction of false alarms, and response timeliness, in order to achieve a balanced and efficient detection process.
- ✓ **Comparison of DQN Variants:** Experimentally compare the main Deep Q-Network variants : standard DQN, Double DQN, Dueling DQN, and DQN with Prioritized Experience Replay (PER) in a highly variable and constrained IoMT environment, in order to identify the configuration offering the best trade-off between detection performance and computational efficiency.
- ✓ **Integration of an Adaptive Rule Engine:** A modular software architecture is proposed to continuously generate updated security rules, thereby strengthening proactive defense mechanisms and enabling dynamic responses to emerging and previously unknown threats.
- ✓ **Experimental Evaluation:** Test the robustness of the architecture on the CICIoMT2024 dataset, using conventional metrics: accuracy, recall, F1 score, and latency.

4. Thesis Content

The remainder of this manuscript is organized as follows:

Chapter 1: General Context and IoT Security in E-Health

This first chapter establishes the conceptual foundation underpinning the entire work. Three main themes are addressed in succession. The first concerns the technological foundations of IoT and IoMT, with particular attention to their distinguishing properties and operational limitations. The second focuses on the specific security requirements of connected medical environments: patient data confidentiality, integrity, and continuous service availability. The third provides an inventory of the main threats likely to affect these systems DoS/DDoS attacks, IoT-dedicated botnets, network intrusions, and illicit extraction of clinical information. The resulting analysis highlights the weaknesses of conventional defensive strategies and provides justification for developing intelligent and distributed solutions.

Chapter 2: State of the Art

This chapter is devoted to an in-depth examination of scientific publications addressing intrusion detection in IoT and IoMT ecosystems. Three main families of methods are reviewed: statistical machine learning, deep learning, and deep reinforcement learning. For each, demonstrated contributions are distinguished from unresolved challenges. A recurring gap is notably identified: the absence, in existing literature, of a methodical comparison of the different DQN variants applied to the specific context of IoMT.

Chapter 3: Methodology and Proposed Architecture

This third chapter details the distributed architecture proposed. It is based on Deep Q-Learning principles and organized according to a three-tier topology Edge, Fog, then Cloud. The adaptive rule module, the multi-objective reward function, and the data stream processing pipeline are described in succession.

Chapter 4: Implementation and Experimental Results

This chapter reports on the conditions under which experiments were conducted. It specifies the technical environment used, the datasets exploited with particular reference to CIC-IoMT-2024 and the performance observed for each DQN variant evaluated. A detailed comparative analysis accompanies the presentation of these results.

General Conclusion and Perspectives

The final chapter offers a synthesis of the contributions of this work, discusses its inherent limitations, and outlines several directions in which future research could engage to extend or improve the developed solutions.

Chapter 1: General Context of IoT and Security in E-Health

1.1. Introduction

Digital technology has profoundly transformed the healthcare sector. The Internet of Things (IoT) is one of the main drivers of this evolution. Adapted to the medical domain, it has given rise to the Internet of Medical Things (IoMT). This specialized extension opens new prospects, particularly in terms of remote patient monitoring, improvement of care quality, and optimization of medical data management. However, this growing interconnection of hospital equipment is not without consequences from a security standpoint. It multiplies the entry points exploitable by malicious actors. The data exchanged within these infrastructures is particularly sensitive, while connected medical devices operate under strong hardware and operational constraints. In this context, IoMT environments become prime targets for cyberattacks. Before designing intelligent, adaptive, and distributed defense mechanisms, it appears necessary to understand the architecture of IoT and IoMT, identify their vulnerabilities, and analyze the threats likely to affect them. This chapter thus presents the fundamental concepts of IoT applied to e-health, the main security requirements, and the various attacks that can compromise connected medical infrastructures.

1.2 Internet of Things (IoT)

1.2.1 Definition and Architecture

The Internet of Things (IoT) designates a technological paradigm whose founding principle lies in the interconnection of objects from the physical environment. These artifacts are equipped with capabilities allowing them to acquire, transmit, and analyze data, enabling informational transactions via digital networks. The International Telecommunication Union [1] offers a more rigorous characterization: the IoT is presented as a universal infrastructure connecting both concrete and abstract entities, using mechanisms for identification, data capture, and information transmission. This institutional definition inscribes the IoT in the continuity of the classical Internet, extending its scope of application to the material sphere. The Internet of Things can be understood as the convergence of three complementary approaches [2]:

- an object-centered approach;
- a sensor-oriented approach;
- a service-focused approach.

The integration of these three perspectives gives rise to a hierarchical structure whose purpose is to distribute and order the various missions assigned to the system.

1.2.1.1 Classic Three-Layer Model

According to the foundational literature, the IoT architecture is organized around three hierarchical levels, illustrated in Figure 1.1:

- **Acquisition Layer (Perception Layer):** Perception Layer: integrates all acquisition devices (sensors, actuators, embedded modules) whose function is to collect raw signals from the physical environment. It represents the primary interface between the real world and the digital system.
- **Communication Layer (Network Layer)** : This layer transmits data over the network. It uses protocols that fit IoT devices, like IP, MQTT, and CoAP .Its job is to set up and keep the connection working between sensors and the platforms where data gets analyzed
- **Service Layer (Application Layer) :** This layer gives end users access to advanced and useful features. Examples include watching patients from a distance and systems that send alerts for clinical situations.

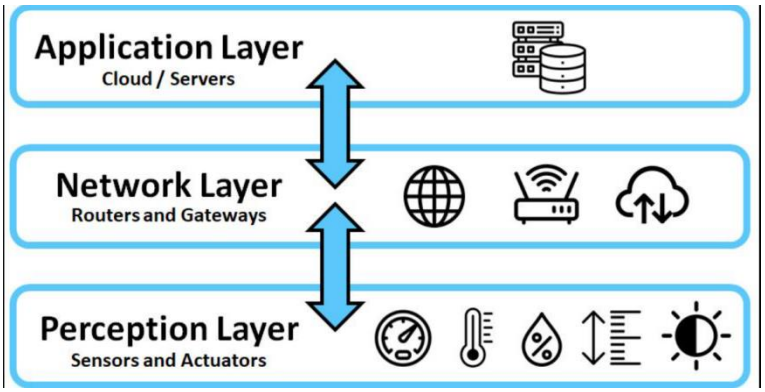

Figure 1.1: Classic three-layer IoT architecture

1.2.1.2 Extended Four-Layer Model

To handle the needs of distributed computing and intermediate analysis, newer architectures add one extra layer that focuses on processing and services [4]. This expanded four-layer structure, shown in Figure 1.2, includes the following levels:

- **Perception Layer:** This stratum integrates all physical acquisition devices sensors, actuators, and embedded modules whose function is to capture raw signals from the physical environment and constitute the primary interface between the real world and the digital system.
- **Network Layer:** This intermediate stratum is responsible for the transport of data between the perception layer and the upper processing tiers. It encompasses distributed processing functions, flow analysis, local decision-making, and intelligent mechanisms such as filtering, anomaly detection, and adaptive learning. It is at this level that computation is relocated in close proximity to data sources, in accordance with the Edge Computing and Fog Computing paradigms.
- **Application Layer:** This uppermost stratum delivers advanced, domain-specific services to end users across vertical sectors, including healthcare, transportation, and smart infrastructure management.

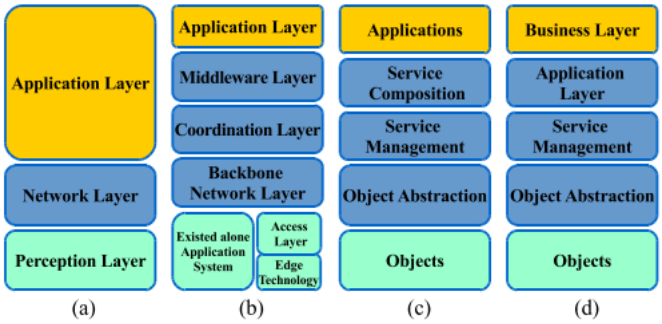

Figure 1.2: Four-layer IoT architecture [4].

1.2.2 Characteristics and Constraints of the Internet of Things

The Internet of Things (IoT) presents a set of structural traits that both govern its mode of operation and define its operational limits. Among its characteristic attributes, one can cite:

- Large-scale connection of a multitude of disparate objects capable of communicating with each other [10];
- Technological diversity, concerning both hardware components, communication protocols, and operating systems [10];
- Distribution across vast geographical areas, often within moving and poorly controlled environments [11];
- Permanent data generation, in the form of streams characterized by high volume and strong variability [8];
- Limited autonomy coupled with local computing capacity, granting objects basic processing and arbitration capabilities [10].

These characteristics are nevertheless accompanied by significant structural handicaps. First, IoT devices systematically face drastic restrictions in computing resources: modest processing power, reduced memory capacity, limited energy autonomy [9]. Such limitations hinder the implementation of sophisticated security mechanisms. Second, the attack surface, enlarged by permanent connectivity and device variety, considerably increases vulnerability to cyber threats [10]. Poor authentication mechanisms and the lack of regular software updates add another layer of risk, making already vulnerable systems even harder to defend. Third, scalability and traffic management concerns become worrying as the pool of communicating objects grows exponentially [8]. Centralized infrastructures quickly reveal their weaknesses in terms of response time and transmission capacity. Fourth, ever-strengthened requirements for data confidentiality and integrity constitute a cross-cutting obstacle, especially in sensitive domains such as e-health [11]. Ultimately, IoT is traversed by a fundamental contradiction between, on the one hand, its potential for functional extension and, on the other, its inherent vulnerability. This duality justifies the deployment of flexible and distributed defensive strategies, adapted to severely constrained environments. In the e-health universe, these constraints take on even greater acuity. IoMT devices handle medical data of high prognostic value and operate within infrastructures that demand total operational reliability and near-instantaneous temporal reactivity.

1.3. IoT in E-Health

1.3.1 Applications (Medical Sensors, Telemedicine, etc.)

The extension of the Internet of Things to the healthcare sector has given rise to a specialized branch designated as the Internet of Medical Things (IoMT). This domain is characterized by the networking of AI-embedded biomedical instruments, sensors dedicated to human physiology, and clinical platforms operating in digital environments [15].

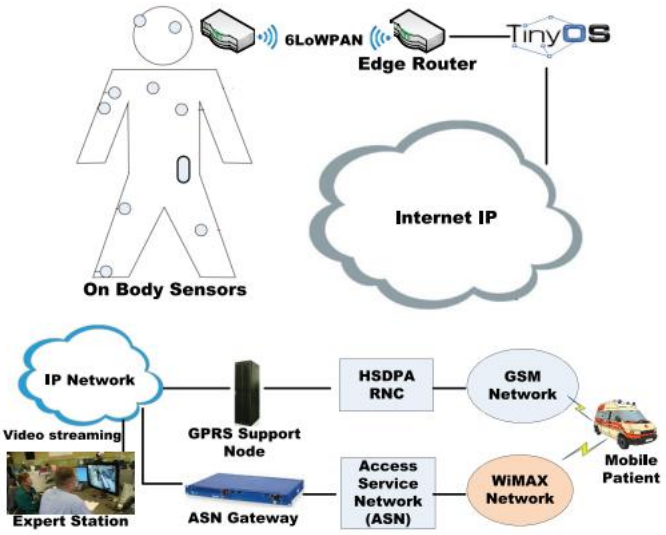

Figure 1.3: Remote monitoring architecture in medical IoT [19].

Among the most significant applications, one can distinguish:

- **Wearable medical devices (wearables):** Enable continuous acquisition of physiological parameters such as heart rate, blood glucose concentration, or oxygen saturation level [13].
- **Implantable smart devices:** Capable of transmitting real-time information to clinical supervision infrastructures [14].
- **Telemedicine and remote monitoring:** Facilitates monitoring of patients suffering from chronic conditions while contributing to reducing hospital stay lengths [15].
- **Interconnected clinical environments:** Bring together within interoperable networks devices such as infusion pumps, multi-parameter monitors, and imaging equipment [16].
- **Predictive analytics tools:** Exploit IoT data streams to predict clinical status deterioration and improve care relevance [13].

All these applications rest on three fundamental pillars: systematic acquisition of data in varied formats; their transfer under guaranteed security conditions; and their exploitation through advanced algorithmic processing, all frequently inscribed within the framework of distributed architectures.

1.3.2 Challenges and Benefits

The positive outcomes of integrating IoT technologies into the digital health universe prove particularly substantial, whether viewed from the clinical practice, care organization, or economic perspective. Regarding the strictly medical dimension, three advances merit

highlighting. The first concerns real-time supervision of patient parameters, which considerably increases the speed of decision-making in potentially dangerous situations [13]. The second relates to the emergence of individualized medicine, which draws its legitimacy from retrospective and prospective exploitation of personal medical histories [13]. The third concerns better allocation of technical and human resources within institutions, made possible by finer regulation of therapeutic pathways [15]. These incontestable advances do not, however, obscure the existence of equally considerable challenges. The question of protecting connected medical devices ranks first among concerns. Gaps affecting embedded software and deficient configurations make healthcare organizations vulnerable to methodically orchestrated cyberattacks [10].

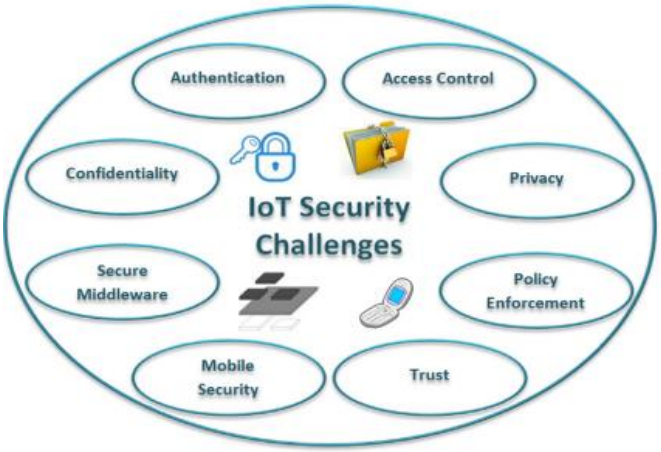

Figure 1.4: Main security challenges in IoT environments [11].

The preservation of the confidential nature of health data strictly regulated by binding normative frameworks calls for the deployment of proven mechanisms in terms of identity verification, cryptographic encoding of data streams, and differentiated management of access authorizations [11]. Operational permanence and continuity of services constitute a third imperative, as any service interruption can lead to immediate harmful effects on the integrity of persons under care [16]. Ultimately, IoT deployed in the context of digital health proves to be a promising tool for medical practice. It nonetheless requires sophisticated security solutions specifically designed to address the critical nature and high risk permeability of clinical ecosystems.

An alarming finding emerges from the most recent quantitative analyses: the resilience of digital systems in healthcare continues to deteriorate. The 2025 IBM Cost of a Data Breach Report reveals that the global median cost of a data breach amounts to USD 4.44 million, with healthcare organizations among the most severely affected [20]. In the specific IoMT environment, this fragility becomes critical. The scale of the problem becomes clearer when looking at institutional data. A 2023 study jointly conducted by the AHA, HHS, and HSCC found that an overwhelming majority of hospital networks 96% were still running on legacy systems or software with unresolved security flaws. This is not merely a technical oversight; outdated infrastructure of this kind provides fertile ground for malware to spread, especially botnets, which thrive in poorly secured networked environments. [21]. What is particularly alarming is that the situation showed no signs of improvement between 2022 and 2023. During this period, cyberattacks on networked medical devices made a strong comeback. Among the most notable threats were new malware strains built upon the Mirai botnet framework variants

that had been deliberately engineered to target medical equipment specifically. [22]. Over the same period, IoT infrastructures suffered an unprecedented wave of DDoS attacks: a fourfold increase in incidents during the first half of 2023, generating global financial damage estimated at USD 2.5 billion [23]. All these indicators converge toward the same unambiguous diagnosis: IoMT ecosystems have become prime targets for malicious actors employing ever-evolving tactics. The danger posed by botnets to these infrastructures is now quantified, objectified, and continuously growing. Faced with this reality, it becomes imperative to build detection systems capable of continuous adaptation, modest in resource consumption, and designed from the outset for environments where operational constraints are severe.

1. 4 Security in Medical IoT

Ensuring an adequate level of security within IoMT environments constitutes a fundamental requirement, given both the high sensitivity of the medical information processed and the repercussions that a cyber incident could have on care continuity. IoMT architectures are distinguished from conventional information systems by three characteristic traits: great device heterogeneity, severely restricted computational resources, and a particularly demanding set of regulatory obligations. As a result, the security question cannot be reduced to a mere technical problem; it rises to the level of a clinical and organizational challenge in its own right.

1.4.1 Data Confidentiality

IoMT devices continuously acquire and transmit nominative medical information including physiological signs, patient records, and therapeutic histories. Safeguarding this data aims to prevent any intrusion, fraudulent capture, or unauthorized disclosure. Preserving confidentiality relies on several mechanisms:

- encryption of transmission channels (TLS, DTLS);
- reinforced authentication of parties involved in exchanges;
- access control systems based on functional attributes;
- segmentation of the network into distinct zones.

A failure of these mechanisms is likely to engender privacy violations, regulatory penalties, and erosion of institutional trust. Research indicates that IoT networks dedicated to the medical domain prove particularly vulnerable to man-in-the-middle attacks, identity spoofing maneuvers, and mass data exfiltration operations [11], [14].

1.4.2 Integrity and Availability

In addition to confidentiality, two other fundamental properties integrity and availability underpin security in IoMT environments.

Integrity: This property aims to certify that clinical information has not undergone any alteration, whether during storage or transfer. Any fraudulent modification of an essential physiological parameter is likely to induce an inappropriate therapeutic decision. Cryptographic techniques notably digital signatures and hash functions are frequently employed to guarantee this property [14].

Availability: In the hospital context, inaccessibility of a service can have immediate repercussions on the physical integrity of patients. Distributed denial-of-service (DDoS) attacks, often launched from networks of infected devices (IoT botnets), represent a serious threat to

operational continuity [17]. The ability of systems to resist and recover (resilience) therefore asserts itself as a central element of security in the medical domain. It is from this perspective that Edge Computing holds particular interest: by relocating decision-making processing as close as possible to medical devices, it guarantees detection and intervention capability even when the centralized infrastructure experiences partial failure.

1.4.3 Specific Constraints

The deployment of effective security mechanisms within IoMT ecosystems faces several structural obstacles:

- **Restricted hardware capabilities:** Restricted hardware capabilities: low processor power, limited RAM, energy constraints;
- **Strong technological heterogeneity:** Strong technological heterogeneity: plurality of manufacturers, multiplicity of communication protocols and normative frameworks;
- **Extended software update cycles:** Extended software update cycles: these delays are imposed by regulatory approval procedures;
- **Extreme reactivity requirements:** Extreme reactivity requirements: the need to reach decisions in near-instantaneous timeframes.

These characteristics hinder the adoption of overly heavy or centralized security solutions. Traditional approaches relying on fixed signature analysis prove insufficiently effective when confronted with polymorphic and continuously renewed attacks. Recent scientific publications emphasize the necessity of designing adaptive and distributed mechanisms capable of operating at the network edge that is, in the immediate vicinity of medical devices [18].

1. 5 Threats and Attacks in Medical IoT

The attack surface exploitable by adversaries within IoMT architectures has undergone remarkable expansion, a phenomenon directly attributable to the widespread adoption of communications between medical devices and their connection to IP-based networks. This heightened vulnerability constitutes fertile ground for the proliferation of highly diverse cyber threats, whose negative repercussions cannot be limited to the digital infrastructure of healthcare institutions alone they can compromise the continuity of clinical activities and, more fundamentally, endanger the condition of persons under care. Cyberattacks directed against IoMT can typically be classified according to four main typologies: denial-of-service attacks, networks of infected devices commonly referred to as IoT botnets, intrusions targeting network infrastructure, and offensives aimed directly at medical information.

1.5.1 DoS / DDoS

Denial-of-service attacks aim to render a computing service inoperable by overwhelming its processing capacity or available bandwidth. When taking a distributed form (DDoS), these attacks exploit a large number of previously compromised machines to amplify the saturation effect. Within IoMT environments, the repercussions of such offensives can prove particularly severe:

- shutdown of remote patient monitoring systems;
- inaccessibility of servers hosting electronic medical records;

- disruption of communications between interconnected clinical devices.

These disruptions originate in a structural fragility specific to IoT networks: devices with insufficient protection levels constitute prime targets, particularly when operating with factory-default configurations or weak authentication procedures [17].

1.5.2 Botnets IoT

Networks of infected devices commonly referred to as IoT botnets rank among the most formidable cyber threats to connected medical ecosystems. Their operational pattern consists of discreetly infecting a multitude of communicating objects, which then transform into "bots" placed under the remote control of an attacker via a Command & Control (C&C) infrastructure.

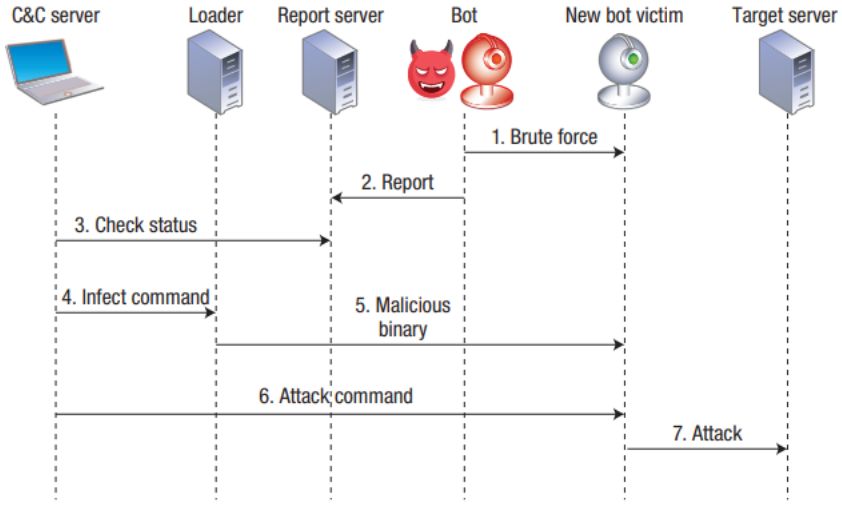

Figure 1.5: Architecture of a Mirai-type IoT botnet [17].

Once this network orchestration is established, a botnet is capable of:

- launching large-scale distributed denial-of-service (DDoS) campaigns;
- proceeding with exfiltration of confidential information;
- serving as a relay point for other attacks.

The Mirai botnet, first detected in 2016 and extensively documented in the literature [17], constitutes an emblematic case study: a trivial attack vector the retention of default manufacturer-set credential pairs sufficed to enslave over 600,000 IoT terminals, triggering large-scale failures on critical DNS infrastructures worldwide. This precedent foreshadowed a dangerous inflection: the emergence of derivative strains such as MedBIoT, specifically designed to compromise healthcare institution devices [22]. In the highly constrained context of IoMT networks, a successful intrusion via such a tool can lead to the blinding of life-support monitors, with the immediate consequence of a rupture in the clinical care chain.

1.5.3 Network Attacks

Within IoMT environments, network-layer attacks primarily exploit weaknesses in communication protocols and insufficient authentication mechanisms. These include:

- Man-in-the-Middle attacks;
- identity spoofing maneuvers (spoofing);

- injection of fraudulent packets into the data stream;
- passive capture of network traffic (sniffing).

The plurality of protocols used in IoT, combined with the absence of systematic encryption procedures, constitutes fertile ground for exploiting vulnerabilities present at the network and transport layers. In the medical sector, any alteration or interception of information streams is likely to induce inappropriate clinical decision-making [11].

1.5.4 Attacks on Health Data

Health data constitutes one of the most coveted resources on illicit markets, due to its diagnostic value and nominative nature. Cybercriminals deploy various techniques to obtain it:

- pillaging of patient records;
- falsification or destruction of therapeutic data;
- use of ransomware against hospital computer platforms.

When information accuracy is altered, the consequences can affect the diagnostic act or therapeutic decision concerning a patient. Moreover, the legal texts governing the storage of personal health data impose categorical prescriptions upon care organizations, so any breach carries particular gravity. The juxtaposition of diverse hardware and non-homogeneous security strategies has the effect of increasing the probability of violations against both the discretion and reliability of clinical information in IoT environments [19].

1.6 Conclusion

This introductory chapter has established the theoretical and security foundations indispensable to understanding this research. The study of the properties of IoT and IoMT revealed a fundamental opposition between, on the one hand, the clinical contribution of these technologies and, on the other, their inherent fragility in the face of cyberattacks. The analysis of the three essential security components confidentiality, integrity, and availability showed that IoMT environments are subject to constraints of particular severity, to which traditional defensive strategies fail to respond satisfactorily. The typology of attacks presented in Section 1.4 denial-of-service, IoT botnets, network intrusions, and offensives targeting medical data confirms that IoMT represents a privileged target for adversaries whose methods are constantly evolving. All these observations justify the adoption of a detection method that is both intelligent and decentralized, based on the Deep Q-Learning algorithm and integrated within an Edge–Fog–Cloud hierarchical organization. The theoretical foundations of this approach will be developed in the following chapter.

Chapter 2: State of the Art (IDS and Deep Reinforcement Learning)

2.1 Introduction

The recent rise of digital medicine has conferred a strategic role on IoMT that is now widely recognized. This paradigm encompasses a heterogeneous set of connected devices — biological sensors, physiological monitoring systems, embedded clinical equipment with local processing capabilities whose primary function is to ensure the continuous acquisition and transmission of patient data, often covered by medical confidentiality. The reliability of operations and the integrity of signals directly condition the relevance of treatments and patient safety. This functional importance, however, exposes IoMT to a high cybernetic risk. Botnets are among the most formidable attack vectors in this sector. Their mechanism relies on the silent compromise of a large pool of devices, subsequently mobilized in a coordinated manner for malicious purposes: exfiltration of clinical data, network service congestion, or disruption of hospital infrastructure operations. Faced with adversaries whose operational patterns are constantly becoming more sophisticated, conventional Intrusion Detection Systems (IDS) exhibit prohibitive structural weaknesses. Methods that proceed by matching against pre-established signature databases systematically fail against novel or morphologically variable attacks. In an environment as labile as IoMT, the effectiveness of these traditional tools degrades critically. To overcome these deficiencies, the scientific community has turned to statistical machine learning techniques, then to deep learning. A more recent path is now attracting particular interest: Deep Reinforcement Learning (DRL). The originality of this approach lies in the fact that an autonomous agent progressively builds an optimal decision policy through simple interaction with its environment, without requiring human-labeled corpora. Within this algorithmic family, the Deep Q-Network (DQN) and its many variants :Double DQN, Dueling DQN, DQN with Prioritized Experience Replay (PER) have established themselves as reference models in the literature. This chapter critically analyzes these algorithms and their contribution to intrusion detection in IoMT contexts, with a dual objective: to expose the limitations of each variant, and to use this critique to justify the architecture proposed in this work.

2.2 Intrusion Detection Systems (IDS)

Among the tools contributing to the protection of computer systems, IDS occupy a central position. Their vocation is to bring to light any abnormal or malicious activity whether traversing a network or taking place on a machine. In IoT and IoMT ecosystems, their contribution becomes even more decisive. The reason is twofold: these environments are characterized by marked heterogeneity of their components and high permeability to network-based attacks.

2.2.1 Definitions and Role of an IDS

An Intrusion Detection System (IDS) is a tool that continuously monitors network or computer system activities to identify malicious behaviors and generate alerts. Unlike a firewall, which

applies static filtering rules on packet headers, an IDS examines content, temporal patterns, and relationships between flows in order to detect activities not captured by traditional filtering rules.

According to deployment type:

- **Host-based Intrusion Detection System (HIDS):** This category of IDS operates at the level of a particular entity a server, workstation, or connected medical device. This type of system analyzes operating system traces; application calls, and files integrity checks. When applied to IoMT environments, two main challenges hinder its deployment. First, medical devices possess limited computational capabilities. Second, updating their embedded software is often impossible because of strict regulatory certification requirements.
- **Network-based Intrusion Detection System (NIDS):** This device monitors all communications transiting the network infrastructure. It captures frames, examines flows, and identifies abnormal or suspicious patterns. Particularly well-suited to the constrained IoMT environments, it serves as the foundation of our proposed defensive strategy, as it requires no alteration of the medical devices themselves.

2.2.2 Types of IDS

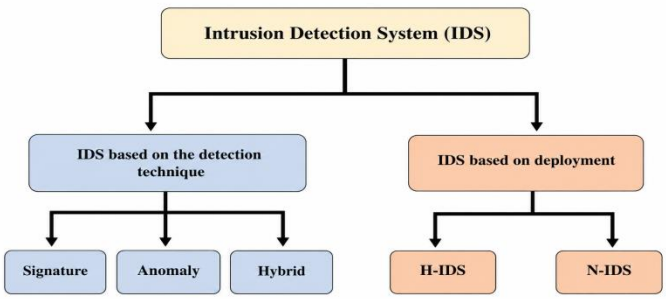

Figure 2.6: Classification of Intrusion Detection Systems.

According to detection method :

- **Signature-Based IDS:** The operating principle of signature-based IDS is as follows: the traffic traversing the network is compared against a library of digital fingerprints characterizing previously recorded cyberattacks. If a match is established, the system generates an alarm signal. The popularity of this methodology is explained by two strengths: its relatively simple design and its capacity to effectively counter already-referenced threats. It nonetheless suffers from a critical weakness namely its inability to recognize attacks that have never been observed or that have been subject to transformation. The effectiveness of signature-based IDS is strictly limited to threats that have been previously catalogued, which compromises their deployment in highly unstable environments such as IoT [41].
- **Anomaly-based IDS:** Anomaly-based intrusion detection systems model the activities considered normal within the monitored infrastructure. Any sufficiently marked deviation from this normality reference is then interpreted as potentially malicious. This methodology has the advantage of being able to reveal previously unknown attacks. It

nonetheless faces a recurring difficulty: a generally high false positive rate, particularly in complex and volatile environments such as IoMT. While anomaly detection constitutes an interesting research avenue, its stabilization remains delicate due to the great variability of behaviors considered normal [42].

- **Hybrid IDS:** A particular category of intrusion detection systems, qualified as hybrid, proceeds by fusing signature-based techniques with behavioral anomaly identification. This operational complementarity offers a more advantageous balance than either family taken in isolation: diagnostic accuracy is improved, as is resilience against attacks still unknown to existing repositories. The dual modality thus deployed effectively recognizes already-documented assaults while triggering alerts when as-yet-unreported activity patterns appear. These advantages are accompanied, however, by a non-negligible operational cost: such architecture is more delicate to design and its computational requirements exceed those of simpler approaches. In the particular context of IoT environments, where hardware capabilities are often limited, these prerequisites can hinder the practical adoption of hybrid solutions [43].

2.3 IDS in IoT and E-Health

An observation emerges from the examination of recent scientific literature: intrusion detection systems (IDS) applied to IoT and IoMT ecosystems are attracting considerable interest, driven by the desire to adapt traditional cybersecurity tools to meet the constraints of networks where connected objects proliferate. The common ambition of these approaches can be simply stated: to ensure uninterrupted monitoring of network traffic, capable of signaling malicious activity, without ever transgressing the operational limits characteristic of the medical environment.

2.3.1 Existing Solutions

In the domains of the Internet of Things and digital health, intrusion detection mechanisms are organized schematically according to three broad families of strategies that prove mutually enriching. A first series of proposals employs statistical machine learning algorithms to scrutinize network communications in order to identify significant behavioral deviations. Among these methods are supervised classifiers Support Vector Machines (SVM), Random Forests, K-Nearest Neighbors (KNN) regularly used to establish a reliable distinction between legitimate exchanges and hostile flows [43]. A second category of research favors deep learning architectures, with the ambition of automatically producing elaborate descriptions from raw network information. Models built on deep neural networks notably Convolutional Neural Networks (CNN) and recurrent LSTM networks have demonstrated their effectiveness for intrusion identification in IoT environments with high device diversity [44]. A third, more contemporary, orientation turns to decentralized infrastructures underpinned by Fog Computing and Edge Computing concepts. Such a configuration enables data processing as close as possible to its sources, thereby reducing dependence on centralized cloud platforms and increasing the response speed of defensive mechanisms [45].

2.3.2 Encountered Problems

Notwithstanding the advances recorded in this domain, the implementation of intrusion detection systems within IoT and e-health environments remains hindered by several major obstacles.

- **Limited Resources:** A first obstacle lies in the resource frugality characterizing IoT devices. These suffer from reduced computing power, limited memory, and restricted energy autonomy. These limitations compromise the hosting of sophisticated models directly on connected devices [45].
- **Latency:** A second significant difficulty relates to time constraints.In the medical world, decision-making processes must operate in real time or within extremely narrow time windows. Centralized cloud-hosted architectures introduce routing and processing delays that are difficult to reconcile with the critical requirements inherent to the care sector.
- **Scalability:** A third technological obstacle relates to system extensibility. The rapid growth of the connected device pool generates a considerable expansion of the information volume to be analyzed, a phenomenon that saturates conventional IDS and degrades their operational performance. Current available approaches consequently struggle to jointly satisfy the three fundamental requirements of IoMT environments: detection accuracy, response speed, and large-scale deployment.

The set of weaknesses identified in the preceding paragraphs reveals a structural limitation common to most conventional IDS: they rely either on time-fixed models or on supervised learning schemes that require periodic parameter revision. Yet IoMT environments are characterized precisely by the continuous emergence of threats capable of self-transformation such as distributed botnet architectures or intrusions exploiting unknown signature vulnerabilities. In such a context, only an adaptive and self-learning approach can claim to maintain an acceptable level of protection. These observations naturally lead to advocating for a mechanism capable of acquiring decision-making competence through direct experience, without awaiting external reconfiguration. This necessity forms the basis for choosing reinforcement learning.

2.4 Machine Learning for IDS

Faced with the constant evolution of cyber threats and their growing capacity for polymorphism, machine learning and deep learning approaches constitute the foundation on which the majority of modern IDS rest. What fundamentally distinguishes them from classical signature-based approaches is two essential properties: their capacity to automatically extract characteristic patterns from data streams, and their ability to generalize against malicious behaviors never previously encountered.

2.4.1 Machine Learning for IDS

In the field of intrusion detection, historical machine learning methods rely on two preliminary operations: a manual feature construction phase (feature engineering), followed by a supervised or semi-supervised learning phase performed on network flows whose labels have been pre-assigned.

- **Support Vector Machines (SVM):** These classifiers prove useful in problems involving the separation of two categories within high-dimensional spaces. Their central

mechanism consists of positioning a hyperplane that maximally separates legitimate observations from hostile ones. Experimental literature, particularly that exploiting KDD and NSL-KDD datasets, has repeatedly confirmed their relevance for intrusion identification [46], [47].

- **Random Forests**: These models function by aggregating a large set of decision trees. Each tree is constructed from a randomly selected subset of data and predictor variables. They combine remarkable robustness to random perturbations, satisfactory generalization capacity, and measured computational load. These qualities explain their prevalence in IDS designed for IoT [46].
- **K-Nearest Neighbors (KNN) and Naive Bayes Classifier :** These algorithms, the former based on distance calculations, the latter on a probabilistic approach are also employed to identify anomalies. Their performance degrades, however, when data dimensionality increases or when the statistical distribution of samples departs from certain assumptions.

➢ These models nevertheless present recurring weaknesses. Their effectiveness is closely correlated to the relevance of manually constructed features, and they exhibit an inability to account for complex dependencies or temporal sequences within network flows. These restrictions limit their power against cyberattacks that transform or deploy in a coordinated manner.

2.4.2 Deep Learning for IDS

The limitations inherent in manual feature construction pushed research toward deep learning. This paradigm deploys multi-layer neural networks whose distinctive characteristic is to autonomously generate representations organized on a hierarchical scale from unstructured data.

- **Convolutional Neural Networks (CNN)**: These networks first emerged in the domain of image processing. Their extension to intrusion detection required reinterpreting network flows as matrices with a defined structure. Under this representation, two principal functions emerge: identifying characteristic local configurations and bringing to light the spatial relationships linking different attributes [48].
- **Recurrent Neural Networks (RNN)** et **Long Short-Term Memory (LSTM)** :These networks are specifically conceived to capture the dependencies that connect consecutive elements within a time series. When applied to network packet inspection, they demonstrate a particular aptitude for identifying malicious activities that develop over extended periods such as botnet infections or distributed denial-of-service campaigns. [49].

➢ These architectures generally outperform conventional methods on two indicators: prediction accuracy and their ability to transfer learning to previously unseen attacks. They are, however, burdened by two substantial disadvantages: high computational resource consumption and dependence on massive data volumes to achieve satisfactory learning. In IoT and IoMT ecosystems, characterized by often limited hardware means and severe temporal requirements, these constraints can prove difficult to accommodate. The opposition between the two major methodological families can be summarized as follows. Classical statistical learning approaches combine honorable operational efficiency with appreciable computational frugality, at

the cost of dependence on a prior phase of manual feature design. Deep learning approaches, conversely, offer finer rendering of phenomena marked by complexity and temporal variability at the cost of significantly increased computational resource requirements. This comparative assessment has gradually steered research orientations toward more flexible methodological frameworks. Deep Reinforcement Learning is situated within this dynamic, its particularity being to articulate the decision dimension with the evolutionary nature of cyber threats.

2.5 Reinforcement Learning (RL)

2.5.1 Definition

Reinforcement Learning (RL) is distinguished from other branches of machine learning by a specific operational principle. A software agent evolves in an environment whose configuration changes over time. At each step, the agent perceives the current state of the environment, selects an action from a set of possibilities, triggers it, and then receives a positive or negative numerical signal evaluating the relevance of the action taken. This feedback allows it to progressively adjust its behavioral strategy. This sequential acquisition mode contrasts with supervised learning, which requires a stock of pre-categorized data. RL is therefore particularly suited to domains where uncertainty reigns and configurations evolve, as is the case in cybersecurity. The mathematical formalization of RL draws on the theory of Markov Decision Processes (MDP). [50] [51].

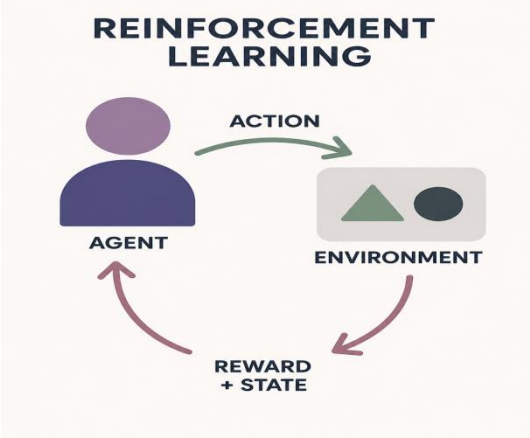

Figure 2.7: The fundamental interaction loop between the agent and its environment.

2.5.2 Fundamental Concepts

A reinforcement learning system is organized around several fundamental components, described below.

Agent: The agent designates the entity responsible for decision-making. In the context of an intelligent IDS, this entity can be assimilated to a module whose function is to examine the state of the network and determine the action to execute (triggering an alert, implementing a block, isolating a network segment, etc.)

Environment: The environment corresponds to the sphere with which the agent interacts. In an IoT or IoMT context, this sphere integrates the network infrastructure, the data flows circulating within it, both legitimate and malicious behaviors, and temporal fluctuations of traffic.

States: At each discrete instant t, the environment is characterized by a state denoted $\boldsymbol{s_t}$. This state assembles all relevant information describing the current configuration of the system namely flow statistics, observed packet rate, anomaly indicators, etc.

Actions: The agent selects an action at from a repertoire of possible actions. In a detection architecture, this selection can lead to several types of operations: classifying traffic as normal, reporting an attack, blocking a flow, or adjusting a detection threshold.

Reward : Upon executing each action, the agent receives a scalar numerical signal, referred to as the reward r_t, which quantifies the immediate appropriateness of the decision taken. A correctly identified attack yields a positive reward, whereas a false alarm or a missed intrusion incurs a negative penalty, thereby guiding the agent toward progressively more accurate detection policies.

Policy: The policy, denoted π(a|s), formally defines the agent's behavioral strategy by specifying the probability of selecting action aa a given the current environmental state **s**. Two policy families are distinguished:

- **Deterministic policy:** maps each state to a single, fixed action with certainty.
- **Stochastic policy:** associates each state with a probability distribution over the action space, allowing multiple actions to be selected with varying likelihoods.

The ultimate objective of reinforcement learning is to identify an optimal policy $\pi*$ that maximizes the expected cumulative discounted reward accumulated over time, formally expressed as $E[\sum_{t=0}^{\infty} \gamma^t \, r_t]$

The Bellman Function : The following equations anticipate the DQN formalism, presented in detail in Section 2.6. Determining an optimal behavioral policy, denoted π*, classically relies on the action-value function. For a given state-action pair, this function expresses the mathematical expectation of the discounted sum of future rewards when an agent takes action a from state s before conforming to π for the remainder of the trajectory. The expression of the optimal value function derives from a recursive principle known as the Bellman equation applied to the optimal case:

$$Q^*(s,a) = E[r_t + \gamma \, max_{\mathrm{a}}{}' Q^*(s_{t+1}, a`) \mid s_t = s, a_t = a] \qquad (2.1)$$

Où :

- r_t **:** the reward the agent gets at that moment.
- $\gamma \in [0,1]$ s the discount factor.
- s_{t+1} : the next state.

This step-by-step relationship is the key mathematical idea behind Q-Learning. It also supports Deep Reinforcement Learning methods like the Deep Q-Network.

Loss Function

Many DQN papers use the Mean Squared Error (MSE) as the loss function. However, some versions choose the Smooth L1 loss, also called Huber loss. This choice is made because Smooth L1 handles extreme values (outliers) better than MSE.This function behaves like MSE for small errors and like MAE for large errors, making it particularly suited to IoMT environments. Indeed, in these networks, sudden traffic variations during botnet attacks can generate significant momentary prediction errors. Using MSE alone would amplify these errors and destabilize learning. The Smooth L1 function is defined by:

$$L_\delta(y,\hat{y}) = \begin{cases} \frac{1}{2}(y-\hat{y})^2 & si\ |y-\hat{y}| \le \delta \\ \delta.|y-\hat{y}| - \frac{1}{2}\delta^2 & \text{else} \end{cases} \quad (2.2)$$

This approach stabilizes learning by reducing the influence of abnormal transitions, frequent during botnet attack phases, while preserving precise convergence for low and moderate errors.

Target Network

In the DQN algorithm, direct use of the Bellman equation can cause learning instability, as the target value depends on the same parameters currently being optimized. To remedy this difficulty, a second neural network, called the target network and denoted $Q(s,a';\theta^-)$, is introduced. This network has the same architecture as the main network $Q(s,a';\theta)$, but its parameters θ^- are kept constant for a certain number of iterations. It intervenes exclusively in the computation of the target:

$$y_t = r_t + \gamma\, max_{a'} Q(s_{t+1}, a`; \theta^-) \quad (2.3)$$

The target network parameters are periodically updated by copying the main network parameters:

$$\theta^- \leftarrow \theta \quad (2.4)$$

This strategy stabilizes the learning process by avoiding the moving target problem, reducing oscillations, and promoting more reliable convergence toward the optimal value function.

2.5.3 Decision Process

The formal representation adopted by reinforcement learning is that of the Markov Decision Process (MDP), expressed through the quintuple (S,A,P,R,γ), where each component is defined as follows:

- **S:** the state space, encompassing all possible configurations of the environment observable by the agent.
- **A:** the action space, comprising the complete set of interventions the agent is capable of performing.

- **P(s′|s,a):** the state transition probability function, which specifies the likelihood of transitioning to state s′ upon executing action aa a in state s, thereby characterizing the stochastic dynamics of the environment.
- **R(s,a):** the reward function, which maps each state-action pair to an immediate scalar signal quantifying the desirability of the agent's decision within the current environmental context.
- **γ∈[0,1]:** the discount factor, which governs the relative weight assigned to future rewards with respect to immediate ones. Values of γ approaching unity encourage the agent to adopt long-term optimization strategies, whereas lower values prioritize immediate gains.

The Markov property stipulates that the transition to the next state s s′ is conditionally independent of all prior history, depending exclusively on the current state ss s and the action aa a executed therein. This memoryless assumption formally justifies the MDP framework as a suitable model for sequential decision-making under uncertainty. Within this formalism, the agent seeks to derive an optimal policy π∗(a|s)that is, a mapping from states to actions that maximizes the expected cumulative discounted reward:

$$\pi^{*} = \arg max_{\pi} = E_{\pi}\left[\sum_{t=0}^{\infty} \gamma^{t} r_{t} \mid s_{0} = s\right] \quad (2.5)$$

When applied specifically to intrusion detection in IoT and IoMT environments, reinforcement learning offers several theoretically grounded advantages:

- a continuous adaptation capacity to non-stationary environments whose traffic characteristics and threat patterns evolve over time;
- an inherent aptitude to model the sequential and temporal dependencies underlying network flow behavior;
- a capacity to simultaneously optimize multiple, partially conflicting objectives namely, maximizing attack detection accuracy, minimizing false alarm rates, and controlling the operational overhead imposed on constrained Edge nodes [50], [51].

These properties confer particular relevance on reinforcement learning for connected medical environments, which are characterized by high service criticality, strict regulatory constraints, and strong dynamic variability in network traffic.

2.6 Deep Reinforcement Learning (DRL)

Deep Reinforcement Learning (DRL) corresponds to an extended version of standard RL, in which multi-layer neural networks are employed to estimate either value functions or behavioral strategies. This device makes accessible the processing of large-scale, continuous, or highly nonlinear state spaces where conventional tabular techniques cannot apply. In IoT and IoMT spheres, the information transiting the network is multidimensional, marked by temporal fluctuations and great hardware heterogeneity. Employing neural networks for approximation then enables the highlighting of complex relationships between traffic variables, while maintaining a continuous adjustment capability. In the field of intrusion detection, DRL grants an agent the capacity to establish an optimal decision rule based directly on traffic characteristic

signs, without needing to explicitly formalize state transition probabilities. The agent progressively modifies its behavior with the goal of increasing a total reward, typically defined from detection quality and the cost associated with spurious alerts. The Deep Q-Network (DQN) and its improved forms notably Double DQN and Dueling DQN are among the most emblematic architectures of DRL, seeking to increase learning consistency and reduce recurring estimation errors.

2.6.1 Standard DQN

The Deep Q-Network (DQN) performs a fusion between the Q-Learning algorithm and a deep neural network, whose mission is to approximate the value function Q(s, a) associated with state-action pairs [52]. The standard Q-Learning update procedure is expressed as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \cdot max_{a'} Q(s', a') - Q(s,a) \right] \quad (2.6)$$

When the number of states becomes large or the state space is continuous, tabular representation becomes unusable. To overcome this limitation, DQN substitutes a parameterized neural network, denoted Q(s, a; θ), for the Q-table. Two fundamental mechanisms contribute to the stability of the learning process:

- **Experience Replay Memory :** This technique consists of storing transitions (s, a, r, s') in a replay buffer, with the objective of eliminating temporal correlations between samples used for learning.
- **Target Network:** A target network whose parameters are updated at regular intervals, which avoids oscillatory behavior and stabilizes learning.

A DQN applied to IDS offers the capacity to incrementally acquire a decision policy that mirrors the dynamic evolution of network traffic. Static approaches based exclusively on signature repositories prove rigid in comparison: the DQN reacts to fluctuations of normal traffic, modulates its choices in light of the global context, and seeks an optimal trade-off between attack detection rate and generated false alarm rate.

With respect to botnet-orchestrated attacks whose behavioral signature rests on distributed and shifting actions, such as coordinated DDoS or malware propagation the DQN's flexibility finds particularly favorable expression. Progressively through learning, the model becomes capable of discerning sophisticated behavioral configurations, even when they correspond to no pattern already recorded in signature databases. This property ultimately consolidates the IDS resilience against emerging threats.

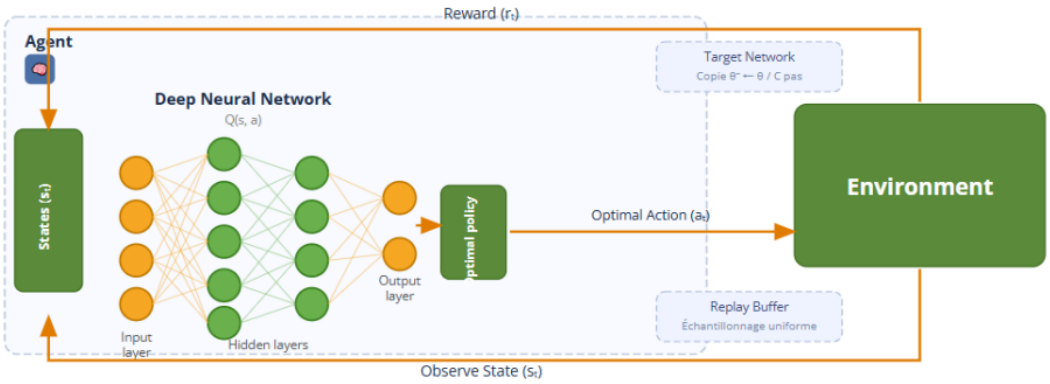

Figure 2.8: Standard DQN architecture.

2.6.2 Double DQN

The DQN algorithm in its initial formulation is affected by an over-optimism bias. This excess finds its source in the use of the maximization operator to estimate future gains, resulting in frequent overestimation of the various action options. A modified version, called Double DQN, was proposed to remedy this shortcoming [53]. This variant introduces a disjunction between two roles:

- the action selection role, assigned to the primary network;
- the action value evaluation role, assigned to the secondary (target) network.

$$r + \gamma\, Q\,(s', \mathit{argmax}_{a'}\ Q\,(s', a'\,; \theta), \theta^{-}) \qquad (2.7)$$

This functional disjunction attenuates overestimation and increases learning robustness. In IoMT environments, where estimation errors could generate harmful clinical decision-making, this characteristic is of paramount importance.

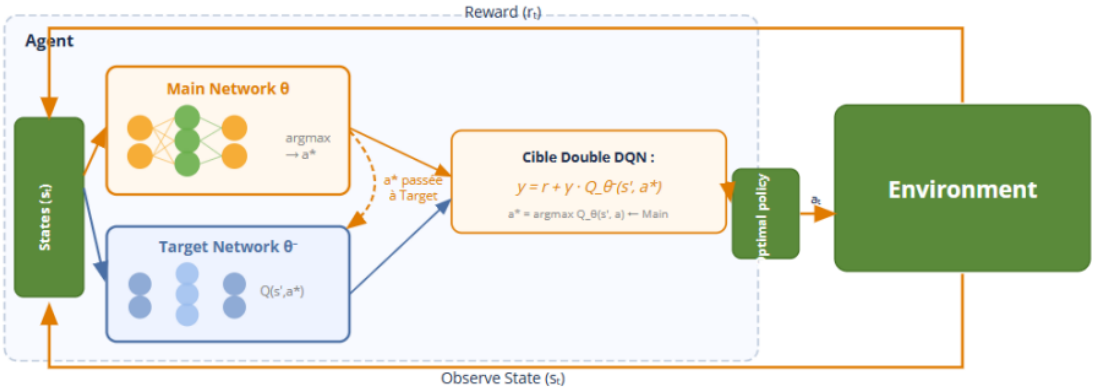

Figure 2.9: Double DQN architecture.

When detecting intrusions and specifically against botnets, i.e., sets of infected terminals under adversary control abusive amplification of values assigned to certain actions generates several deleterious effects: event classification errors, excessive attenuation of the severity of slowly

evolving malicious behaviors, and ineffective regulation of flows with suspicious characteristics. It is precisely to remedy these limitations that Double DQN increases the robustness of decision-making processes, reducing false negatives in two critical use cases: distributed denial-of-service attacks (DDoS) and malicious code propagation phases.

2.6.3 Dueling DQN

A particular neural architecture, called Dueling DQN, is distinguished by a reorganization of the internal network structure, establishing a separation between two distinct estimates [54]:

- the estimate of the state's intrinsic value, denoted V(s);
- the estimate of the relative advantage of each action, denoted A(s, a).

The Q function is then recomposed from these two components according to the following expression:

$$Q(s, a) = V(s) + A(s, a) - \frac{1}{|A|}\sum_{a`} A(s, a`) \quad (2.8)$$

This decomposition of the Q function confers on the model a better ability to detect pertinent configurations, even in situations where actions have relatively similar effects. When transposed to an IDS, an improvement in the capacity to discriminate normal from abnormal flows is observed.

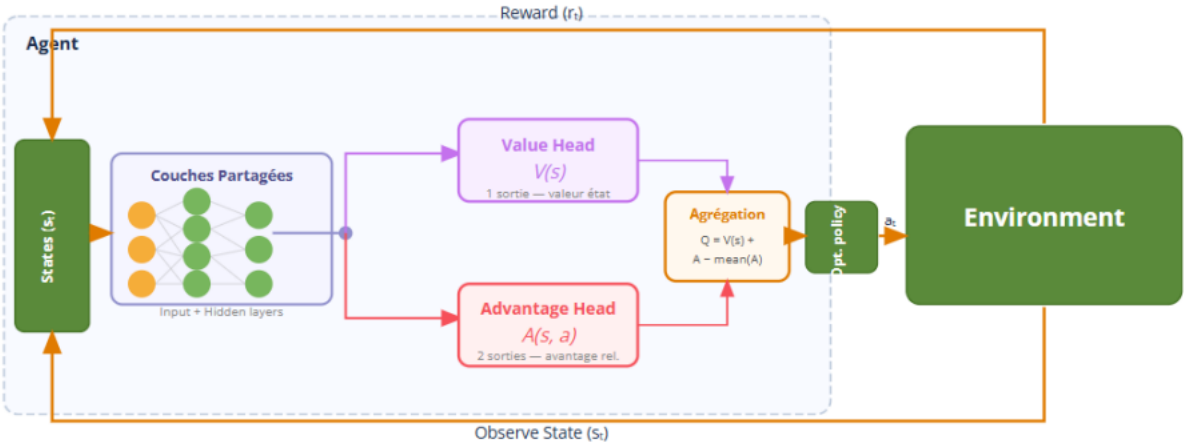

Figure 2.10: Dueling DQN architecture.

In an IoT or IoMT network, certain states may already be highly suspicious (abnormal traffic, sudden peaks, coordinated behaviors). Even if several possible actions produce similar effects, it is essential to identify that the state itself is critical. In the context of IoMT botnets, where several flows may appear individually innocuous but collectively reveal a malicious network state, this decomposition allows the agent to characterize the global dangerousness of the state independently of the immediate action to be taken. Dueling DQN thus improves the discrimination capacity between normal traffic and traffic from a distributed attack.

2.6.4 Prioritized Experience Replay (PER)

PER constitutes an optimized version of the standard DQN, designed to strengthen learning from data stored in memory. It is based on the observation that not all stored transitions present the same informational value [55]. In the conventional DQN, quadruples (s, a, r, s') are drawn

randomly with equal probability from the replay buffer. Yet certain experiences prove more valuable than others for model refinement, particularly those with high prediction error. PER associates each transition with a priority score pi, typically calculated from the Bellman error (or temporal difference error, TD):

$$\delta_i = r + \gamma \, max_{a'} Q(s', a') - Q(s, a) \tag{2.9}$$

The probability of extracting a given transition is then expressed as:

$$P(i) = \frac{P_i^{\alpha}}{\sum_k P_k^{\alpha}} \tag{2.10}$$

Where :

- α : is a parameter regulating the amplitude of prioritization (α = 0 reverts to uniform sampling);
- $p_i = |\delta_i| + \varepsilon$, ε where ε is a very small positive constant. Transitions associated with a large TD error are therefore used more often.

The use of a non-uniform sampling procedure generates statistical distortion. To correct for this, a corrective coefficient is applied during the update step:

$$w_i = \left(\frac{1}{N.P(i)}\right)^{\beta} \tag{2.11}$$

Où :

- **β** : is a parameter whose value is progressively increased to dose the correction
- **N :** represents the total capacity of the memory buffer.

This mechanism guarantees the theoretical convergence of the process toward a stable estimate.

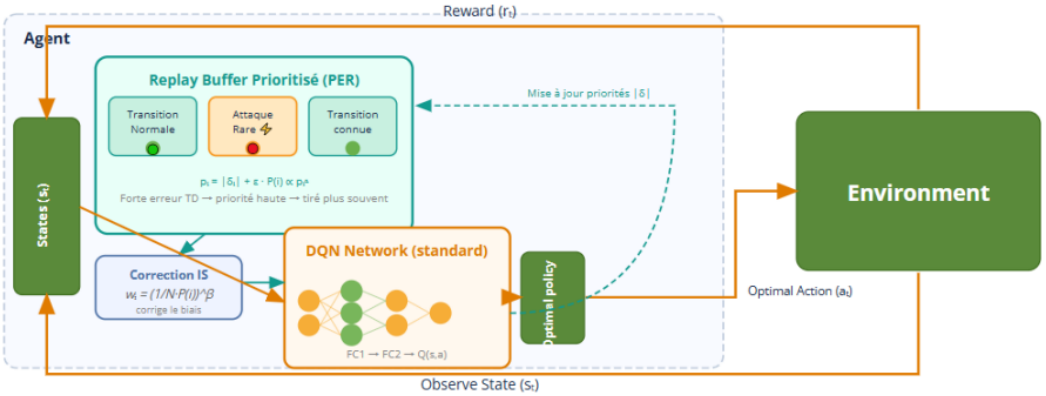

Figure 2.11:PER architecture.

In IoT/IoMT environments:

- attacks are often rare compared to legitimate traffic;
- certain botnet attacks appear intermittently;
- malicious behaviors evolve.

Uniform sampling risks diluting these critical events. PER enables:

- giving greater importance to attack-related transitions;
- improving learning on rare events;
- increasing system sensitivity to new botnet variants.

Thus, PER contributes to reinforcing the IDS adaptation capacity against distributed and dynamic threats.

The choice of the Deep Q-Network rests on several methodological considerations. First, the action space considered in our IDS is discrete (ALLOW, BLOCK, QUARANTINE, MARK), which naturally corresponds to the framework of value-based methods such as DQN. Second, IoMT environments present computational constraints at the Edge level; DQN remains less costly than policy-based methods such as PPO. Finally, the possibility of integrating a multi-objective reward function and an expert rule engine makes DQN particularly suited to a hybrid system combining learning and domain expertise.

2.7 Detailed Review of Existing Works

In order to better delineate the state of the art and clearly identify scientific gaps, we propose grouping existing works according to four main themes: (1) generic DRL approaches for IoT, (2) works specifically dedicated to IoMT and medical data, (3) distributed Edge–Fog–Cloud architectures, and (4) comparative studies of DQN variants.

2.7.1 Generic DRL Approaches for IoT

This category encompasses works that apply deep reinforcement learning to intrusion detection in general IoT environments, without specific targeting of the medical domain. **Khayat et al [24]** propose a coupling between automatic extraction of deep network traffic representations and an RL agent. The originality lies in the joint adaptation of features and decision policy, evaluated on NSL-KDD and CICIDS. The system demonstrates an ability to adapt to new attack signatures without complete retraining, but does not compare DQN variants nor deploy an Edge architecture. **Rookard & Khojandi [26]** present RRIoT, an RL agent equipped with recurrent memory (RNN) to model the temporal dimension of botnet behaviors. Evaluation on N-BaIoT (Mirai, BASHLITE) shows the contribution of temporal memory for discriminating periodic attack patterns. However, this work is limited to detection without prevention, does not explore DQN variants, and ignores the constraints of medical environments. **Alavizadeh et al [37]** constitute a foundational reference for the application of DQN to network intrusion detection. The authors rigorously formalize the problem as an MDP, defining state space, actions, and reward function. Evaluation on NSL-KDD and UNSW-NB15 confirms the superiority of DQN over supervised classifiers. The main limitation is the exclusive use of simple DQN, without exploring variants (Double, Dueling, PER), and the absence of validation in an IoMT context.

Naeem et al [27] explore the use of RL for proactive detection of zero-day vulnerabilities. The agent learns normal system behavior and detects any significant deviation without prior signatures. While the behavioral approach is promising for unknown threats, the work remains on synthetic data without validation on real IoMT environments.

2.7.2 Works Specific to IoMT and Medical Data

Category context. Two profiles of works make up this section: those that deliberately fall within the scope of the Internet of Medical Things (IoMT), and those that exploit authentic databases such as CIC-IoMT-2024. **Shaikh et al [25]** propose an IDS based on Deep Reinforcement Learning (DRL), validated on CIC-IoMT-2024. The main finding is that the method outperforms conventional supervised classifiers against evolving attacks (DDoS, reconnaissance, botnets). Weaknesses include a centralized architecture (neither Edge nor Fog considered), absence of comparison with other DRL versions, and no dynamic rules module. **Gabriel [31] (2025)** Describes a flexible IDS powered by deep reinforcement learning, intended for use in medical IoT networks. The experimental protocol is based on CIC-IoMT-2024. The same structural gaps are observed: single DRL model without comparative study of variants, absence of Edge-Fog-Cloud architecture, and no adaptive rule generation mechanism. **Mahbub et al [39]** implement a domain adaptation technique to transfer the skills of an IDS initially calibrated on CICIDS2017 toward the CIC-IoMT-2024 environment. This transfer approach manages to attenuate the distributional gap between the two domains. However, it ignores reinforcement learning, does not consider Edge architecture, and proposes no confrontation between different DQN implementations. **Nadhira et al [40]** apply reinforcement learning to securing healthcare IoT systems, taking into account the particularities of medical devices (latency tolerance, exchange criticality). Evaluation uses IoT-23 and N-BaIoT datasets. The identified gaps are: absence of comparison between DQN versions, absence of a three-tier architecture (Edge–Fog–Cloud), and absence of a rules generator.

2.7.3 Distributed Edge–Fog–Cloud Architectures

This category encompasses works that explore the deployment of intrusion detection at the network edge (Edge/Fog Computing). **Siregar & Hanafi [36]** propose a hybrid framework combining classical Machine Learning for initial traffic filtering and DRL for fine-grained decision on ambiguous cases, in an Edge IoT architecture. Evaluation on NSL-KDD and CICIDS2018 under real Edge conditions demonstrates the feasibility of DRL on resource-constrained nodes. The work uses only a single DRL model without comparing variants (Double/Dueling/PER) and proposes no adaptive rule engine specific to IoMT. **Fu et al [30]** present PriFedIDS, which applies federated reinforcement learning to intrusion detection in digital health systems. Multiple institutions collaborate to improve a shared model without sharing patient data (GDPR/HIPAA compliance). While the approach is promising for confidentiality, federated differs from local Edge (each institution remains centralized). Furthermore, no DQN variant comparison is provided, and there is no local rule engine.

2.7.4 Comparative Studies of DQN Variants

This category encompasses works that explicitly compare multiple variants of the Deep Q-Network against one another. **Sharma & Kumar [38]** apply Rainbow DQN (combining six improvements: Double, Dueling, PER, NoisyNet, C51, Multistep) to intrusion detection on three benchmarks (NSL-KDD, CICIDS2017, UNSW-NB15). The authors demonstrate that combining all improvements outperforms each taken individually, with F1-score gains of 2 to 8 points. This

is one of the rare studies to systematically compare variants. However, Rainbow is designed for maximum performance, not for deployment on resource-constrained Edge nodes. The work addresses neither the medical IoMT context nor decentralized architectures, and does not integrate an expert rule engine. **Kanimozhi & Ramesh [33]** couple DRL and SDN to detect intrusions and trigger automatic countermeasures via control plane reprogramming. The DRL agent learns to select optimal countermeasures (blocking, redirection, isolation) by attack type on CICIDS2017. While this work achieves an interesting detection-prevention coupling, it is limited to the SDN context, does not explore DQN variants (standard, double, dueling, PER), and concerns neither Edge IoT nor IoMT. **Ren et al [34]** propose an original use of DRL for optimal feature selection as IDS classifier input. The RL agent identifies the most discriminating traffic variables on NSL-KDD and CICIDS, reducing dimensionality while improving performance. This work uses RL for feature selection, not for the classification decision itself. It does not compare DQN variants and is interested in neither Edge nor IoMT.

2.7.5 Systematic Literature Reviews

This category encompasses survey works that analyze and map DRL approaches for IDS in IoT and IoMT environments. **Gueriani et al [28]** propose a complete taxonomy of DRL approaches for intrusion detection in IoT networks. They classify works by agent type, deployment environment, datasets, and metrics, and identify open challenges. As a review, this work carries no experimental contribution of its own. **Kheddar et al [32]** publish a comprehensive IEEE review on RL applied to IDS in communication networks. They propose a rigorous taxonomy, analyze datasets and metrics, and identify active research axes. The review explicitly highlights the absence of multi-variant DQN solutions deployed on Edge for IoMT. **Naghib et al [35]** conduct a systematic review using the PRISMA methodology covering the entire spectrum of IDS for IoMT: rule-based approaches, classical ML, Deep Learning, and RL. The authors produce a detailed mapping of research gaps and confirm the absence of solutions combining multi-variant DQN comparison, Edge deployment, and proactive prevention for IoMT.

2.7.6 Other Complementary Approaches

This category encompasses works with specific contributions (data generation, generative AI, etc.) that do not fall directly within the first four categories. **Javed et al [29]** present ReGAIN, which combines RL and generative AI to address class imbalance in IoT datasets. A generator produces synthetic examples of rare attacks, improving the coverage of the RL agent's training space. Evaluated on CICIDS2017 and N-BaIoT, the system achieves gains on minority classes. However, the generative AI overhead remains incompatible with resource-limited Edge nodes.

Author(s)	Year	Dataset	Technique	Advantages	Gaps vs Our Work
Khayat et al [24]	2024	NSL-KDD, CICIDS2017	DRL + deep features	Dynamic adaptation without retraining; rich learned representations	Standard DQN only; no Edge deployment; no rule generation
Shaikh et	2024	CIC-IoMT-	Centralized	Validated on recent	Centralized

Author(s)	Year	Dataset	Technique	Advantages	Gaps vs Our Work
al [25]		2024	DRL	IoMT dataset; outperforms supervised classifiers on evolving attacks	architecture; single DRL variant; no adaptive rule engine
Rookard & Khojandi [26]	2023	N-BaIoT	Recurrent RL (RNN+RL)	Temporal modelling of botnet behavior; good results on Mirai/BASHLITE	Detection only, no prevention; heavy RNN for Edge; no DQN comparison
Naeem et al [27]	2024	Synthetic	RL	Proactive zero-day vulnerability detection without prior signatures	Validated on synthetic data only; no Edge Computing
Gueriani et al [28]	2023	Multiple (survey)	DRL-IDS Survey	Complete DRL taxonomy for IDS IoT; identification of research gaps	Survey only; no experimental contribution
Javed et al [29]	2024	CICIDS2017, N-BaIoT	RL + Generative AI	Data augmentation for rare classes; improved RL generalization	Generative AI overhead incompatible with Edge nodes
Fu et al [30]	2024	CICIDS2018	Federated RL	Privacy-preserving via federation; GDPR/HIPAA compliance	Federated ≠ local Edge; no local rule engine
Gabriel [31]	2025	CIC-IoMT-2024	Adaptive DRL	Very recent results on IoMT; adaptability to evolving attacks	Single DRL model; no Edge-Fog-Cloud architecture

Author(s)	Year	Dataset	Technique	Advantages	Gaps vs Our Work
				demonstrated	
Kheddar et al [32]	2024	Multiple (review)	IEEE RL-IDS Review	Rigorous taxonomy; exhaustive analysis of RL challenges for IDS in networks	Review only; noted absence of multi-variant Edge solutions
Kanimozhi & Ramesh [33]	2023	CICIDS2017	DRL + SDN	Detection-prevention coupling via RL + automated SDN reprogramming	SDN context only; no Edge IoT; no IoMT
Ren et al [34]	2022	NSL-KDD, CICIDS	DRL + Feature Selection	Optimal feature selection by RL; effective dimensionality reduction	RL for feature selection (not decision); no Edge
Naghib et al [35]	2024	Multiple (review)	Systematic IoMT Review	Complete PRISMA review IDS IoMT; detailed mapping of research gaps	Review only; confirms absence of combined solutions
Siregar & Hanafi [36]	2024	NSL-KDD, CICIDS2018	Hybrid ML + DRL on Edge IoT	Classic ML and DRL combination on Edge; evaluation in real IoT conditions	Single DRL model; no IoMT-specific adaptive rules
Alavizadeh et al [37]	2022	NSL-KDD, UNSW-NB15	Deep Q-Learning (DQN)	Foundational DQN reference for network IDS; rigorous convergence	Standard DQN only; no IoMT validation; no Edge

Author(s)	Year	Dataset	Technique	Advantages	Gaps vs Our Work
				analysis	
Sharma & Kumar [38]	2024	NSL-KDD, CICIDS2017, UNSW-NB15	Rainbow DQN	Combines six DQN improvements; best performance on three benchmarks	Rainbow too heavy for Edge; no medical context
Mahbub et al [39]	2024	CIC-IoMT-2024, CICIDS2017	Domain Adaptation + DL	Effective domain adaptation between datasets; handles distribution shift for IoMT	No RL/DQN; no Edge Computing; adaptation only
Nadhir et al [40]	2024	IoT-23, N-BaIoT	RL for healthcare IoT cybersecurity	Direct RL application to healthcare IoT; medical protocols considered	No DQN variant comparison; no Edge-Fog-Cloud

Table 2.1: Comparative synthesis of 17 analyzed works (IDS IoT/IoMT, 2022–2025).

2.7.7 Advantages of Existing Approaches

A number of positive findings emerge from the study of the aforementioned works. First, the use of Deep Reinforcement Learning has overcome a barrier that conventional supervised methods encountered: DRL agents do not merely classify they build their conduct through interaction with the surrounding environment, granting them flexibility to respond to cyberattacks that transform themselves, while static rule-based approaches reveal their structural limitations. Second, certain studies bring architectural refinements worthy of interest: the use of recurrent memories to account for the temporal development of network exchanges, the deployment of federated learning to preserve the private nature of health data, and interfacing with SDN networks to trigger automated responses. Third, proven DQN versions (Double DQN, Dueling DQN, PER, Rainbow) have provided quantified advances: improved learning process stability, attenuation of the systematic overestimation error, and higher F1 scores on benchmark datasets such as NSL-KDD, CICIDS2017, and UNSW-NB15.

2.7.8 Methodological Limitations and Identified Gaps

Notwithstanding the progress noted, a cross-reading of publications reveals several recurring weaknesses.

- **Absence of systematic comparison of DQN variants in IoMT:** Most works limit themselves to evaluating a single DRL architecture without conducting a methodical comparison between standard DQN, Double DQN, Dueling DQN, and PER in a specifically medical framework. This lack of comparative studies prevents determining which configuration best meets the requirements of IoMT environments.
- **Excessive dependence on non-specific data corpora:** A large body of research depends on standard datasets including NSL-KDD, UNSW-NB15, and CICIDS2017. These datasets do not accurately reflect the specific protocols, timing patterns, or data semantics found in real IoMT networks. Validations conducted on genuinely medical datasets remain too rare.
- **Insufficient consideration of edge network constraints:** Although certain proposals mention deployment in an Edge environment, decision latency, memory occupancy, and computational load indicators are rarely examined in depth. Yet in the IoMT world, response speed is an absolute imperative.
- **Overly rudimentary reward functions:** Most DRL approaches rely on reward functions based essentially on global accuracy or false alarm reduction. The explicit incorporation of expert knowledge relating to medical protocols or clinical priorities remains exceptional.
- **Potential experimental validity issues:** Certain publications do not sufficiently describe their preprocessing steps (notably the order of normalization operations relative to data splitting, or the absence of temporal partitioning). These omissions can generate evaluation biases and lead to overestimation of reported results.

2.7.9 Foundation of the Proposed Contribution

In light of the previously established findings, several avenues for improvement are required to consolidate both the scientific rigor and the operational utility of DRL-based IDS in IoMT environments. The work presented here is situated precisely at the intersection of identified insufficiencies and provides the following elements:

- An organized confrontation of the main DQN variants (standard DQN, Double DQN, Dueling DQN, and PER) in a framework explicitly dedicated to IoMT;
- A three-tier hierarchical organization (Edge–Fog–Cloud) designed to balance local reactivity and global processing capacity;
- A more elaborate reward function, taking into account false alarms, the sensitivity of medical flows, and decision consistency;
- A strict experimental protocol, incorporating chronological data separation and a controlled preprocessing pipeline, designed to ensure the reliability of the reported performance indicators.

This contribution is therefore not limited to the introduction of a new algorithmic variant. It aims to provide a thorough methodological evaluation adapted to the context of DRL for the protection of IoMT infrastructures.

2.8 Conclusion

This chapter has critically reviewed the full set of existing strategies for intrusion detection, spanning the spectrum from legacy solutions based on predefined signature databases to more recent theoretical constructions employing deep reinforcement learning techniques. A finding emerges at the conclusion of this examination: traditional IDS do not meet the requirements posed by IoMT environments, which present a triple difficulty: extreme device heterogeneity, very limited computing means, and continuously renewed attack forms that escape fixed repositories. Furthermore, while supervised learning and deep network approaches have undeniably improved the capacity to detect new patterns, this advance faces two major obstacles: they require large volumes of labeled data, and their computational footprint makes them difficult to deploy on network edge devices. The critical examination of seventeen surveyed works has identified several persistent gaps: absence of systematic comparison of DQN variants in IoMT contexts, insufficient reward functions integrating clinical priorities, and deficit of hierarchical Edge–Fog–Cloud architectures evaluated on representative datasets such as CIC-IoMT-2024. It is in this perspective that the following chapter presents a distributed, adaptive DQN architecture, specifically designed to address these insufficiencies in e-health environments.

Chapter 3: Design and Implementation of a Secure Architecture against Botnet Attacks

3.1 Introduction

A review of the specialized literature leads to the identification of three major shortcomings in the field of anomaly detection applied to the Internet of Medical Things (IoMT). First, no in-depth comparative evaluation has yet been conducted, in the specific environment of digital healthcare, between the main implementations of deep Q-networks namely the original DQN, the Double DQN architecture, the Dueling DQN variant, and the Prioritized Experience Replay (PER) mechanism. Second, the reward functions encountered in prior work remain conceptually simplistic: they fail to adequately model the essential trade-offs between security performance, the real-time requirements characteristic of intensive care services, and compliance with the regulatory frameworks in force within hospital infrastructures. Third, current approaches offer no distributed model capable of simultaneously combining low response times, rigorous protection of personal health data, and scalability adapted to the exponential growth of connected devices. In response to these limitations, this chapter proposes an AI-based security architecture structured according to a hierarchical topology combining Edge, Fog, and Cloud computing. It relies on agents exploiting deep reinforcement learning and includes a methodical comparison of the four aforementioned DQN versions. The fundamental contribution of this work consists in enriching the reward function by incorporating clinical expertise from medical professionals. Such a design allows the learning algorithm to be guided toward decisions that genuinely respect the operational realities and normative obligations specific to healthcare institutions.

3.2 Proposed Global Architecture

3.2.1 Overview

The proposed architecture rests on a four-level hierarchy: IoMT, Edge, Fog, and Cloud. The choice of four layers is motivated by the specificity of the medical domain: IoMT devices constitute a data source in their own right, distinct from the processing layers, unlike generic IoT architectures that often merge the perception layer and the processing layer into a single level. This explicit separation makes it possible to better reflect the operational reality of hospital environments, where connected medical devices are autonomous entities subject to strict regulatory constraints and cannot be modified to embed additional processing functions.

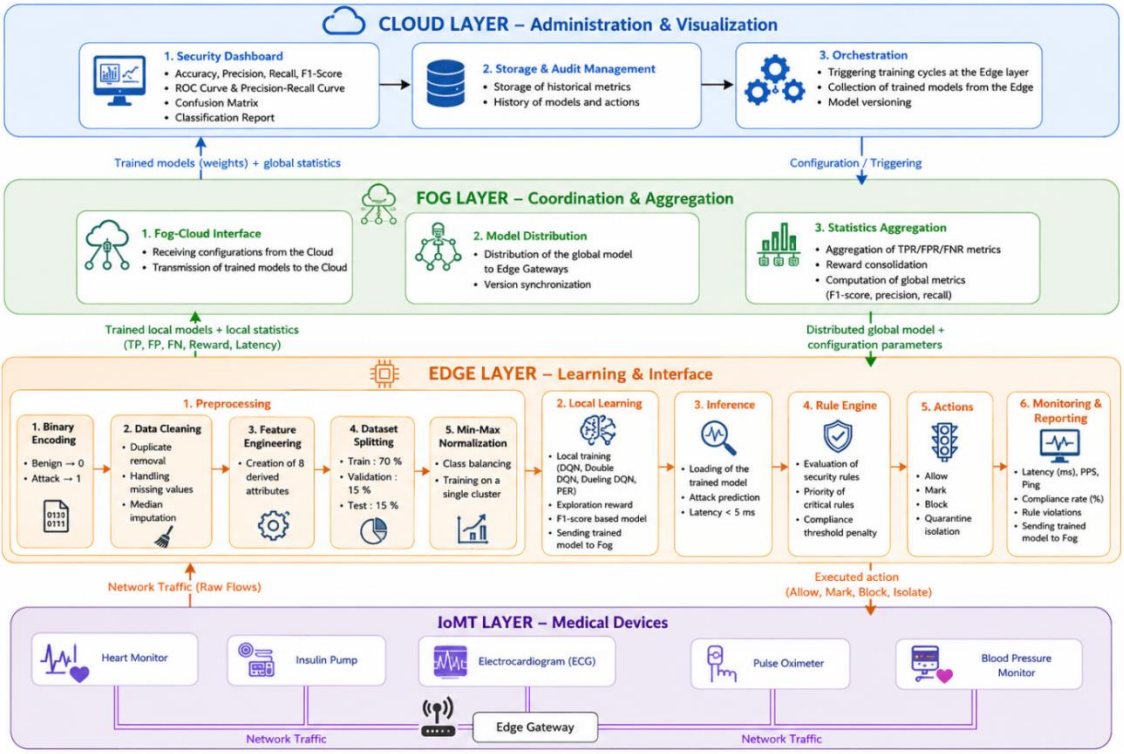

Figure 3.12: Hierarchical Edge–Fog–Cloud architecture for intrusion detection in IoMT networks.

3.2.2 IoMT Layer (Medical Devices)

The IoMT layer constitutes the entry point of the architecture. It groups all connected medical devices deployed in the hospital environment: cardiac monitors, insulin pumps, electrocardiographs, pulse oximeters, and surveillance cameras, among others. The role of this layer is limited to the generation and transmission of raw network traffic to the Edge layer. No local processing is performed, due to the severe hardware constraints that characterize these devices limited computing power, restricted memory, and regulatory certification constraints prohibiting any non-approved software modification. The generated flows cover six traffic classes in accordance with the CIC-IoMT-2024 dataset: benign traffic, ARP spoofing, malformed MQTT data, port scan reconnaissance, TCP SYN distributed denial-of-service, and TCP/UDP denial-of-service. This raw traffic is received as input by the Edge layer for processing.

3.2.3 Edge Layer: Learning and Inference

The Edge layer constitutes the operational core of the system. It is the only layer that executes both local learning and real-time inference, under a strict total latency constraint of less than five milliseconds. It accomplishes six successive and interdependent missions.

1. **Data preprocessing:** Upon receipt of raw traffic from the IoMT layer, a preprocessing pipeline is triggered. Labels are encoded as binary values (benign = 0, attack = 1). The cleaning phase begins with the removal of duplicate records. Missing numerical values are then replaced by the median of their respective attributes. Following this step, eight

derived variables are created through a process of feature construction. The resulting dataset is subsequently divided via stratified random sampling into three separate subsets : 70 % of the samples are allocated to the training set, 15 % to the validation set, and the remaining 15 % to the test set. Min-Max normalization is applied to all features, followed by class rebalancing using the SMOTE-Tomek method, applied exclusively to the training set to avoid any information leakage into the validation and test sets. The thus-prepared data is passed to the next step.

2. **Local learning:** The preprocessed data feeds the training of four variants of the deep reinforcement learning agent: the standard deep Q-network (DQN), the Double DQN, the Dueling DQN, and the DQN with prioritized experience replay (PER). Exploration is ensured by an ε-greedy strategy with progressive decay. Training is performed entirely locally, without any transfer of raw data to the upper layers, in compliance with medical confidentiality requirements. Upon completion of training, the variant offering the best F1 score on the validation set is automatically selected. This retained model is then passed to the inference step.
3. **Real-time inference:** The selected model is loaded to process each incoming network flow. A state vector is constructed from the flow's features, then submitted to the neural network, which produces a decision among four possible actions: Allow, Mark, Block, or Quarantine. This decision is then transmitted to the rules engine before any execution on the network, in order to assess its clinical compliance.
4. **Adaptive medical rules engine:** This component intervenes systematically downstream of each agent decision, without ever modifying it. It evaluates the compliance of the chosen action against a reference framework of eight expert rules hierarchized by clinical severity, then calculates a compliance penalty. This penalty is integrated into the agent's reward function R to guide future learning toward respect of regulatory constraints. The original decision is then passed as-is to the execution step.
5. Action execution: Once the rules engine has confirmed the admissibility of the selected action, it is enforced without delay on the active network flow. The outcome of this enforcement whether it corresponds to a correctly detected attack, an erroneous alert on benign traffic, or an undetected intrusion is systematically logged and fed back into the reward computation mechanism, thereby closing the reinforcement learning loop.
6. **Monitoring and transmission to the Fog layer:** In parallel with its inference and learning activities, the Edge layer maintains a continuous self-assessment by computing a set of local performance indicators, including mean decision latency, the 95th and 99th latency percentiles, the degree of adherence to clinical security rules, and the evolution of accumulated rewards alongside observed rule infractions. At regular intervals, these aggregated statistics and the locally trained model parameters are forwarded to the Fog layer, which consolidates them to produce a system-wide performance overview and ensures coordinated supervision across all deployed agents.

3.2.4 Fog Layer: Coordination and Aggregation

The Fog layer acts as an intermediary between the peripheral agents and the central infrastructure. It performs neither training nor inference; its functions are exclusively the aggregation of local statistics and the distribution of global resources.

Upon receiving the trained models and local statistics transmitted by Edge agents, the Fog layer consolidates each agent's indicators true positives, false positives, false negatives, and cumulative rewards to calculate global performance metrics: F1 score, precision, and recall at the scale of the entire system. This aggregated view is then forwarded to the Cloud layer, without the latter needing to access the raw data from medical devices. In the reverse direction, the Fog layer receives from the Cloud the global configuration and the distributed model, which it redistributes to all Edge gateways while ensuring synchronization of deployed versions, thus guaranteeing consistency of security policy across all agents.

Note: In the proof-of-concept implemented on Google Colaboratory, the functions of the three layers are simulated on the same physical machine. Real deployment on distributed infrastructure, with dedicated communication protocols, constitutes a future development perspective.

3.2.5 Cloud Layer: Administration and Global Supervision

The Cloud layer constitutes the administration and supervision center for the entire system. In accordance with the principle of minimizing centralized processing, it performs neither intensive training nor inference on network flows; its scope is strictly limited to three functions.

First, a security dashboard continuously displays the global performance indicators aggregated by the Fog layer: accuracy, precision, recall, F1 score, areas under the ROC and PR curves, confusion matrix, and detailed classification report. Second, the Cloud layer ensures the storage and audit of anonymized event traces, in compliance with medical data confidentiality requirements, as well as archiving of metrics histories and versioning of all successive models. Third, it manages the orchestration of training cycles: it periodically triggers updates on Edge agents via the Fog layer, collects the produced models, and ensures centralized version management.

3.2.6 Justification for the Distributed Design

The centralized architecture typical of traditional cloud computing presents three major limitations that make it unsuitable for the IoMT context:

- **Latency constraints:** Each request making a round trip to a remote infrastructure exposes the system to delays on the order of several hundred milliseconds. Such values far exceed the admissible time windows for triggering alarms in a sensitive clinical context. This configuration therefore makes real-time security monitoring within hospital premises structurally impossible.
- **Bandwidth saturation:** IoMT devices are capable of generating an information throughput reaching several gigabits per second. Such volume invariably causes saturation of upstream channels, which significantly degrades transmission quality.
- **Personal data protection:** Systematically transferring raw medical information to a remote cloud infrastructure would contradict the principle of data minimization. The latter stipulates that only strictly necessary information for the pursued objective may be processed.

All three of these limitations justify the deployment of an infrastructure distributed across three strata: Edge, Fog, and Cloud. The Edge stratum is entrusted with on-site, near-real-time analysis

of network flows in the immediate vicinity of clinical devices, minimizing decision latency. The Fog stratum provides an intermediate aggregation function and synchronization between different Edge agents, offering a unified view of the network without excessively burdening the central infrastructure. The Cloud stratum, for its part, is confined to latency-tolerant operations, notably event history archiving, periodic updates of algorithmic models, and overall supervision. Such distribution of roles enables the harmonious reconciliation of local responsiveness, controlled device scalability, and strict compliance with medical information confidentiality requirements.

3.3 Datasets and Preprocessing

3.3.1 Presentation of the CIC-IoMT-2024 Dataset

The empirical validation of this study is based on the CIC-IoMT-2024 corpus, developed by the Canadian Institute for Cybersecurity (CIC) affiliated with the University of New Brunswick. Unlike generalist datasets such as KDD99, NSL-KDD, or UNSW-NB15, which originate from standardized network infrastructures, the CIC-IoMT-2024 collection was recorded within a controlled IoMT environment, incorporating both authentic medical devices and communication protocols dedicated to this sector, in particular MQTT for machine-to-machine exchanges and TCP/IP for network transmissions. This singularity confers remarkable suitability upon this corpus for evaluating an intrusion detection mechanism designed for connected hospital infrastructures. The original database includes more than fifty files distributed across eighteen distinct attack classes. For the purposes of this study, six categories were selected, each represented by a file deemed characteristic of its class. This selection follows three complementary criteria: the heterogeneity of attack vectors each retained family targeting a different protocol layer and thus ensuring representative coverage of the most recurrent intrusion paths in IoMT networks; clinical relevance the chosen classes corresponding to the types of offensives identified in the literature as most damaging to hospital infrastructures; and the computational constraints of the Google Colab execution environment, which impose a limit on the volume of exploitable data without performance degradation.

The six retained categories are as follows:

- **Benign:** normal network traffic generated by medical devices, serving as a reference for modeling legitimate behavior;
- **ARP Spoofing:** identity spoofing at the link layer, allowing an attacker to redirect network traffic to a malicious machine;
- **MQTT Malformed Data:** exploitation of the MQTT protocol by injecting malformed data, specifically targeting machine-to-machine communications of IoMT devices;
- **Recon Port Scan:** network reconnaissance through port scanning, generally constituting the preliminary phase of any targeted intrusion;
- **TCP SYN DDoS:** transport layer saturation through SYN packet flooding, aimed at exhausting the resources of connected medical devices;
- **TCP/UDP DoS:** denial of service exploiting TCP and UDP to exhaust target resources, directly threatening the availability of critical medical services.

The dataset originally comprises 39 features distributed across seven functional categories. After the Feature Engineering phase, 8 additional derived attributes are generated, resulting in a final state representation of 47 features used by the DRL agent.

Category	Feature	Description
Packet Features	Header_Length	Packet header length in bytes
	Protocol Type	Type of protocol used in network communication
	Time_To_Live	Packet lifetime (TTL), indicating the maximum number of network hops allowed
	Rate	Flow transmission rate in packets per second
	IAT	Inter-Arrival Time, average time interval between consecutive packets of the same flow
	Number	Total number of packets in the flow
TCP Indicators (Flags)	fin_flag_number	Number of FIN flag occurrences indicating TCP connection closure
	syn_flag_number	Number of SYN flag occurrences, key indicator of SYN Flood attacks
	rst_flag_number	Number of RST flag occurrences signaling abrupt connection reset
	psh_flag_number	Number of PSH flag occurrences forcing immediate data transmission
	ack_flag_number	Number of ACK flag occurrences confirming TCP segment receipt
	ece_flag_number	Number of ECE flag occurrences signaling network congestion
Flag Counts	cwr_flag_number	Number of CWR flag occurrences indicating congestion window reduction
	ack_count	Total number of packets containing the ACK flag in the flow
	syn_count	Total number of packets containing the SYN flag in the flow

	fin_count	Total number of packets containing the FIN flag in the flow
	rst_count	Total number of packets containing the RST flag in the flow
Application Protocols	HTTP	Binary variable (0/1) indicating presence of HTTP protocol
	HTTPS	Binary variable (0/1) indicating presence of HTTPS protocol
	DNS	Binary variable (0/1) indicating presence of DNS protocol
	Telnet	Binary variable (0/1) indicating presence of Telnet protocol
	SMTP	Binary variable (0/1) indicating presence of SMTP protocol
	SSH	Binary variable (0/1) indicating presence of SSH protocol
	IRC	Binary variable (0/1) indicating presence of IRC protocol
Transport Protocols	TCP	Binary variable (0/1) indicating use of connection-oriented TCP protocol
	UDP	Binary variable (0/1) indicating use of connectionless UDP protocol
Transport and Network Protocols	DHCP	Binary variable (0/1) indicating presence of DHCP protocol
	ARP	Binary variable (0/1) indicating presence of ARP protocol
	ICMP	Binary variable (0/1) indicating presence of ICMP protocol
	IGMP	Binary variable (0/1) indicating presence of IGMP protocol

	IPv	Binary variable (0/1) indicating use of Internet protocol (IPv4 or IPv6)
	LLC	Binary variable (0/1) indicating presence of LLC protocol at data link layer
Descriptive Statistics	Tot sum	Total cumulative sum of packet sizes within the flow
	Min	Minimum packet size value observed in the flow
	Max	Maximum packet size value observed in the flow
	AVG	Arithmetic mean of packet size across the entire flow
	Std	Standard deviation of packet size, measuring dispersion around the mean
	Tot size	Total flow size in bytes, representing the overall volume of data exchanged
	Variance	Statistical variance of packet size within the flow

Table 3.2: Description of CIC-IoMT-2024 dataset features.

3.3.2 Preprocessing Pipeline

The preprocessing pipeline comprises six sequential steps ensuring the quality and representativeness of the data submitted for learning.

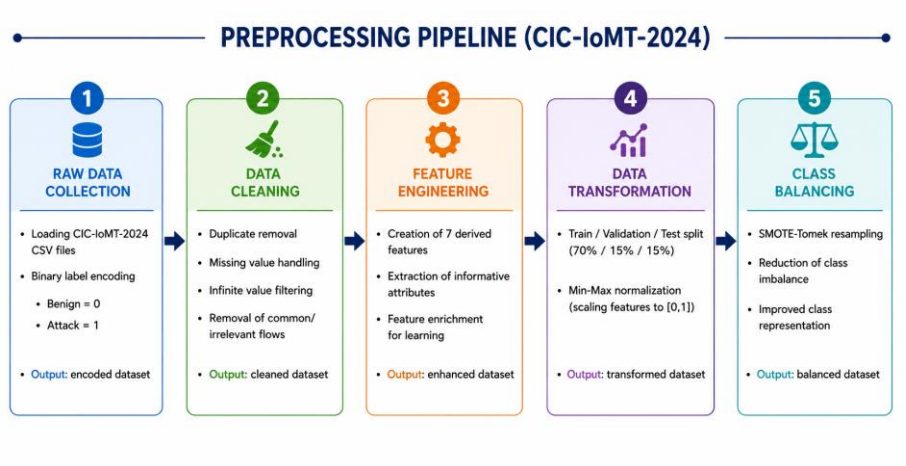

Figure 3.13:CIC-IoMT-2024 data preprocessing pipeline

Step 1: Binary Encoding

The original textual labels are changed into a binary numeric format. In this representation, the class labeled "Benign" becomes 0, while every type of attack is mapped to 1. This transformation

simplifies the classification task while maintaining the fundamental distinction between legitimate and malicious traffic.

Step 2: Data Cleaning

Data cleaning consists of removing strict duplicates, which reduces the total count from 460,850 to 355,222 samples, as well as eliminating any conflicting flows appearing simultaneously in both classes (no such case was detected). Moreover, infinite values, totaling 15,095, are converted to missing values (NaN). All resulting missing values 15,105 in total, of which 10 were originally missing are then imputed by the median calculated for each attribute.

Step 3: Feature Engineering

In order to enrich the behavioral modeling of network traffic, eight derived attributes were constructed. These additional descriptors aim to better capture the dynamics and particularities of flows specific to IoMT environments:

Name	Formula	Objective
Flag_Intensity	$\frac{SYN+FIN+RST}{ACK+\varepsilon}$ where : $\varepsilon = 10^{-6}$	Detection of abnormal behaviors related to SYN Flood attacks by identifying imbalances between connection initiation packets (SYN) and acknowledgments (ACK), indicative of server saturation attempts
TCP_Flag_Sum	Σ(fin, syn, rst, psh, ack, ece, cwr)	Measurement of overall TCP exchange intensity to identify unusual network behaviors associated with scans, intrusion attempts, or automated malicious communications
Protocol_Activity	HTTP+HTTPS+DNS+SMTP+SSH+IRC+Telnet	Quantification of cumulative activity of major application protocols to detect suspicious communication patterns or abnormal usages characteristic of IoMT botnets
Transport_Layer_Type	argmax(TCP, UDP, ICMP)	Dynamic identification of the predominantly used transport protocol in the network flow to characterize traffic type and facilitate differentiation between legitimate and malicious behaviors
Rate_Std_Ratio	$\frac{Rate}{Std+\varepsilon}$	Highlighting flows with excessive throughput regularity, a behavior often observed in automated communications generated by botnets or malicious scripts
Header_Size_to_Payload	$\frac{Header\_Length}{Tot\_size+\varepsilon}$	Detection of packets with an abnormal ratio between header size and payload, a frequent indicator of network scans, control packets, or discrete malicious activities

IAT_Anomaly	$\frac{IAT - AVG}{Std + \varepsilon}$	Analysis of temporal anomalies in packet inter-arrival intervals to identify irregular network behaviors or automated communications characteristic of IoT attacks
Variance_Ratio	$\frac{Variance}{AVG + \varepsilon}$	Identification of sudden variations in the statistical behavior of network traffic, allowing revelation of state changes potentially associated with compromise or malicious activity

Table 3.3: Derived attributes constructed during the Feature Engineering step.

Step 4: Stratified Data Partitioning

The data is divided into three disjoint subsets using a stratification strategy that preserves class proportions in each partition.

Training set : holds 70% of all samples, which amounts to 248,645 records. This set is used solely for updating the parameters of the Q-network. The update procedure relies on gradient backpropagation applied to temporal targets obtained from the Bellman equation. During each iteration, the agent adjusts its weights in order to make the predicted Q-values come closer to the target Q-values.

Validation set : makes up 15% of the entire dataset, which corresponds to 53,293 samples. Its purpose is twofold : adjusting the hyperparameters and picking the configuration that performs best. No training occurs on this subset. Instead, only evaluation metrics namely F1-score, precision, and recall are computed. These values help track how well the model is improving and warn against possible overfitting.

Test set: Also constitutes 15% of the data, totaling 53,284 samples. It is solicited only once, at the end of the training phase, to perform the final evaluation of the system's performance. This data remains entirely unknown to the agent during the learning phase, which guarantees a realistic and unbiased estimate of its generalization capability.

Set	**Size**	**Class 0 (Normal)**	**Class 1 (Attack)**
Train	248 645	13 503	235 142
Validation	53 293	2 894	50 399
Test	53 284	2 894	50 390

Table 3.4: Class distribution in each subset (before balancing).

Step 5 : Normalization

Among all existing normalization techniques, the Min-Max method was selected for this study. Its main advantage lies in its ability to constrain all explanatory variables to evolve within a fixed interval, in this case [0,1]. The applied transformation follows this mathematical expression:

$$x_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (3.1)$$

In this equation, the symbols $\boldsymbol{X_{min}}$ and $\boldsymbol{X_{max}}$ represent respectively the lower bound and upper bound of each attribute. These bounds are determined solely from the data constituting the training corpus. This methodological precaution is essential: any normalization operation performed on the entire dataset before partitioning would cause what is known as data leakage, which would introduce a bias in performance estimation. Consequently, the validation and test sets undergo the same transformation using the parameters derived from the training set, without any readjustment. This approach ensures an objective and unbiased evaluation of the model's predictive capabilities.

Step 6 : Data Balancing

In the CIC-IoMT-2024 dataset, the minority class corresponds to benign traffic, which represents only 5.4% of samples after cleaning (19,291 samples out of 355,222). This pronounced imbalance risks biasing the DRL agent's learning toward almost exclusive detection of attacks, at the expense of recognizing legitimate traffic. To remedy this situation, two advanced SMOTE variants were compared: Borderline-SMOTE and SMOTE-Tomek.

- **Borderline-SMOTE**

  Borderline-SMOTE, proposed by Han et al. [56], focuses exclusively on samples located near the decision boundary. Unlike classical SMOTE introduced by Chawla et al. [57], this variant first identifies samples misclassified by a k-nearest neighbor classifier (k-NN, k=5), then applies SMOTE only to samples considered 'dangerous', i.e., those located in an overlap zone between the two classes.

Principle:

Let $\mathbf{x_i}$ be a sample belonging to the minority class. We compute the number m of its k nearest neighbors belonging to the majority class:

- If $\mathbf{m} > \frac{\boldsymbol{k}}{\mathbf{2}}$: $\mathbf{x_i}$ is considered borderline and SMOTE is applied.
- If $\mathbf{m=k}$: $\mathbf{x_i}$ is considered noise and is ignored.
- If $\mathbf{m=0}$: $\mathbf{x_i}$ is considered safe and is not processed.

- **SMOTE-Tomek**

SMOTE-Tomek, studied notably by Batista et al. [58], combines two successive steps: oversampling by SMOTE [57] followed by cleaning via Tomek links.

Principle :

After the oversampling phase by SMOTE, Tomek links are removed, allowing the boundary between the two classes to be cleaned.

SMOTE generates synthetic samples by linear interpolation:

$$x_{new} = x_i + \lambda(x_{zi} - x_i) \quad , \lambda \in [0,1]$$

(3.2)

In a second step, Tomek links are identified and removed. A Tomek link is a pair $(\boldsymbol{x_i}, \boldsymbol{x_j})$ belonging to different classes such that each point is the nearest neighbor of the other:

$$\forall x_k \ , \ \mathrm{d}(x_i,x_j) < \mathrm{d}(x_i,x_k) \ \text{ et } \ \mathrm{d}(x_i, x_j) < \mathrm{d}(x_j,x_k) \qquad (3.3)$$

Removing these pairs allows the decision boundary to be cleaned after the generation phase.

Criterion	Borderline-SMOTE	SMOTE-Tomek
Samples concerned	Only borderline samples	All minority samples
Generation phase	Targeted generation	Full generation
Cleaning phase	None	Removal of Tomek links
Noise handling	Ignores noisy samples	Removes noisy samples
Overlap risk	Moderate	Very low
Additional parameters	k (k-NN)	k (Tomek links)

Table 3.5: Comparison between Borderline-SMOTE and SMOTE-Tomek.

Note: The overlap risk assessment is qualitative and based on the mechanisms of the considered methods. Borderline-SMOTE may increase overlap because synthetic samples are generated near decision boundaries, whereas SMOTE-Tomek reduces overlap through Tomek Link cleaning.

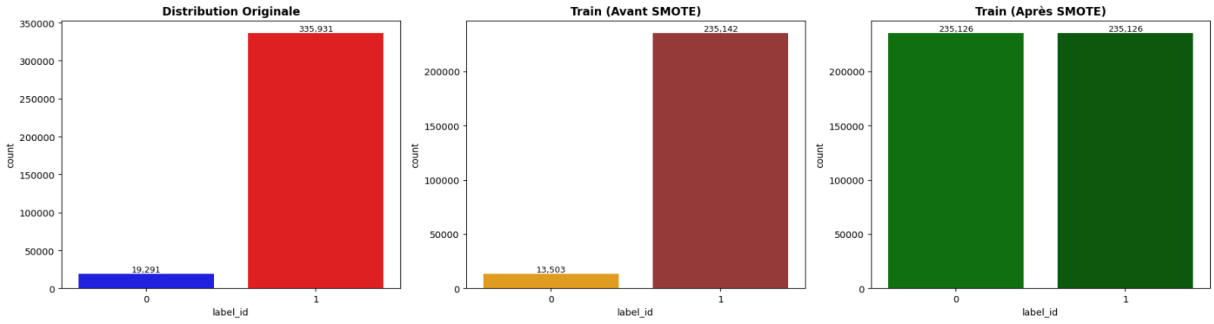

Figure 3.14: Class distribution after balancing with SMOTE-Tomek.

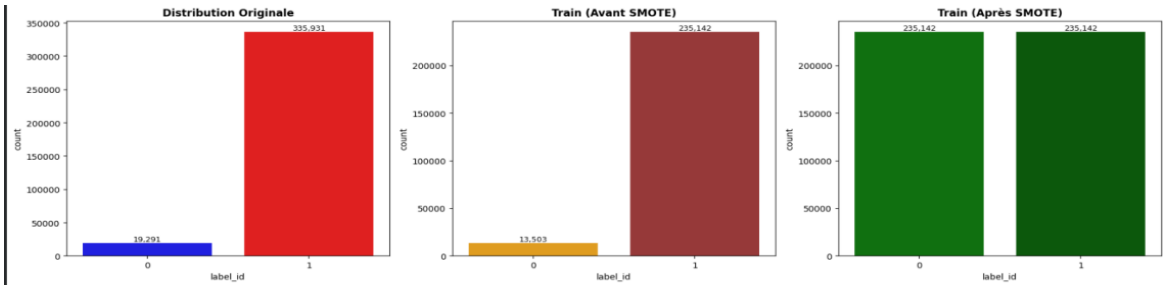

Figure 3.15: Class distribution after balancing with Borderline-SMOTE.

The choice of SMOTE-Tomek is motivated by three considerations specific to the characteristics of the CIC-IoMT-2024 dataset. First, given the very pronounced imbalance (94.6% attacks), Borderline-SMOTE risks under-representing regions of the feature space distant from the decision boundary, leaving areas of benign traffic poorly covered by the model. Second, the dataset contains a large number of binary features (HTTP, HTTPS, TCP, UDP, ARP, etc.). Linear SMOTE interpolation on these variables generates non-integer values introducing artificial noise. The Tomek link removal phase eliminates precisely these marginal samples after generation, which Borderline-SMOTE does not do. Third, SMOTE-Tomek acts simultaneously on both classes: it oversamples the minority class while removing ambiguous majority examples located at the boundary. This dual action produces a cleaner decision boundary, which is particularly beneficial for a DRL agent whose decision policy is sensitive to the quality of separation between normal and malicious traffic.

3.4 Problem Formalization as a Markov Decision Process

Intrusion detection in an IoMT environment is formalized as a Markov Decision Process (MDP), defined by the quintuple (S, A, P, R, γ), where:

- S : is the state space,
- A: is the action space,
- P: is the transition probability function,
- R: is the reward function,
- γ : is the discount factor.

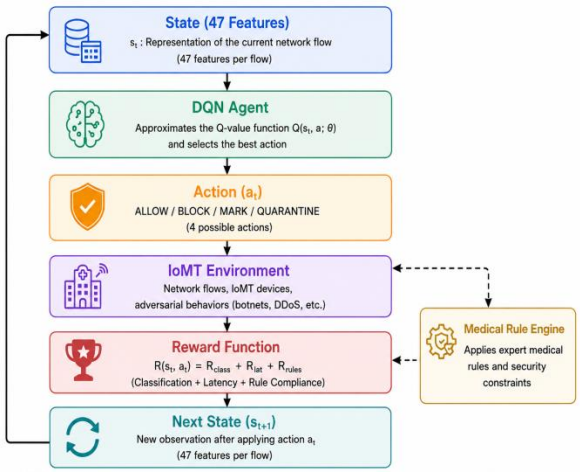

Figure 3.16:MDP Formalization of the Intrusion Detection Problem in IoMT.

3.4.1 State Space (S)

The state perceived by the agent at each discrete time t is constructed from a sliding window of the w last network flows. This parameter allows capturing the short temporal dependencies characteristic of botnet behaviors (periodicity of 'heartbeats', bursts of abnormal connections).

The complete state $\boldsymbol{s_t}$ is obtained by concatenating the **w** last observation vectors:

$$s_t = concat(o_{t-w+1}, o_{t-w+2}, \dots\dots\dots, o_t) \in R^{w \times d}$$

with d = 47 features and w ∈ {3, 4} depending on the variant. The state space dimension is therefore 141 dimensions for the standard DQN (w=3) and 188 dimensions for the three other variants (Double DQN, Dueling DQN, and DQN+PER). This representation offers two major advantages:

1. It preserves the temporal order of flows, essential for distinguishing legitimate traffic from periodic botnet behavior.
2. It homogenizes variable scales after normalization, thus stabilizing neural network learning.

Category	**Features**	**Count**
Headers and protocols	Header_Length, Protocol Type, Time_To_Live	3
Throughput statistics	Rate, Std, Variance, AVG, IAT, Min, Max, Number, Tot size, Tot sum	10
TCP Flags	fin_flag, syn_flag, rst_flag, psh_flag, ack_flag, ece_flag, cwr_flag, ack_count, syn_count, fin_count, rst_count	11
Application protocols	HTTP, HTTPS, DNS, Telnet, SMTP, SSH, IRC, TCP, UDP, DHCP, ARP, ICMP, IGMP, IPv, LLC	15
Engineered features	Flag_Intensity, TCP_Flag_Sum, Protocol_Activity, Transport_Layer_Type, Rate_Std_Ratio, Header_Size_to_Payload, IAT_Anomaly, Variance_Ratio	8
Total		**47**

Table 3.6: Composition of the state space (47 features per flow).

3.4.2 Action Space (A)

The agent can choose among four discrete actions, defined to reconcile reactivity and granularity of the security response:

Action	**Code**	**Description**	**Typical Use Case**
ALLOW	0	Allow the flow without alert	Normal traffic complying with rules

Action	Code	Description	Typical Use Case
MARK	1	Allow but record for later analysis	Non-critical suspicious behavior requiring human investigation
BLOCK	2	Immediately block the flow	Clearly characterized attack (botnet, DDoS, port scan)
QUARANTINE	3	Isolate the source device in a partitioned virtual network	Compromised medical device, high threat requiring technical intervention

Table 3.7:DRL agent action space and typical use cases.

Three related reasons support the choice of limiting the action space to only four discrete options:

- **Compatibility with the DQN algorithm:** The Q-Learning formalism, upon which this approach rests, was historically designed to operate on discrete action sets, making this configuration naturally suited.
- **Control of computational complexity:** When the agent has only a few actions to pick from, it has fewer combinations to try during training. This makes the system able to run on Edge nodes, where the hardware is not very powerful and memory is limited.
- **Clinical relevance of the proposed gradation:** The ordered sequence ALLOW → MARK → BLOCK → QUARANTINE translates a progressive escalation of response severity. This gradation, easily understandable, makes the agent's choices accessible to both healthcare professionals and information security managers.

3.4.3 Transition Probabilities and Discount Factor (γ)

The probabilities governing the transition from one state to another are not initially known and vary over time. This characteristic justifies the choice of reinforcement learning over other paradigms requiring prior knowledge of the dynamic model. The discount factor γ was set at 0.96. Such a value allows the agent to favor the optimization of long-term strategies, an essential property for detecting botnets whose malicious activities generally extend over prolonged time intervals.

3.4.4 Reward Function Design (R)

The reward function constitutes one of the most original contributions of this study. While prior works generally content themselves with valuing only classification accuracy rewarding true positives and true negatives our proposal integrates three partially antagonistic objectives: detection correctness, reaction speed (expressed as decision latency), and compliance with the security rules in force at the healthcare institution. Let R(s_t, a_t, s_{t+1}) be the reward perceived by the agent after executing action a_t, in state (s_t. It decomposes into three terms:

$$R = R_{classification} + R_{latency} + R_{\text{compliance}}$$

3.4.4.1 Classification Component ($R_{classification}$)

Let $y_t \in \{0, 1\}$ be the ground truth: $y_t = 0$ for a normal flow (Benign), $y_t = 1$ for an attack (any type). The reward matrix associated with the chosen action is defined in Table 3.7.

Ground truth	ALLOW (0)	MARK (1)	BLOCK (2)	QUARANTINE (3)
Normal (0)	+1,2	+0,5	-1,5	-0,8
Attack (1)	-2,0	+0,5	+1,5	+1,2

Table 3.8: Reward matrix for the classification component.

The asymmetry of this matrix reflects medical priorities:

- The cost of a false negative (attack treated as normal) is -2.0, the most severe penalty, because an undetected attack can compromise patient safety.
- The cost of a false positive (normal flow blocked) is -1.5, high but lower than false negatives, because an untimely blockage can disrupt care continuity without directly endangering the patient's life.
- For an attack, BLOCK (+1.5) is slightly better valued than QUARANTINE (+1.2), because totally isolating a device can impede legitimate communications (e.g., an insulin pump that must send data to a central monitor).
- The MARK action (+0.5) receives a moderate reward in both classes, encouraging the agent to flag ambiguous cases without making an irreversible decision.

3.4.4.2 Latency Component ($R_{latency}$)

Let τ be the measured decision time (in milliseconds) between flow capture and action execution. This measurement is performed by the LatencyMetrics class, which times neural network inference and rules evaluation. The latency component is defined by:

Condition	Reward
$\tau \leq 5$ ms (compliant with Edge requirements)	+0,3
$5 < \tau \leq 10$ ms (acceptable but not optimal)	0
$\tau > 10$ ms (out of specification, too slow for critical applications)	-0,5

Table 3.9: Reward component related to decision latency.

This progressive penalization encourages the system to maintain inference times compatible with the requirements of real-time medical applications. The 5 ms threshold corresponds to the maximum recommended latency for critical alarm systems, while 10 ms constitutes the upper tolerable limit.

3.4.4.3 Medical Rules Compliance Component ($R_{\text{compliance}}$)

Let V be the set of security rule violations detected by the expert engine (detailed in Section 3.5). Each violation $v \in V$ has a severity level $sev(v) \in [0, 1]$ (1.0 being maximum severity). The compliance component is written as:

$$R_{\text{compliance}} = +0{,}3 \cdot 1_{\,v=\emptyset} - 0{,}3 - \min(0{,}6;\, 0{,}15 \cdot \textstyle\sum_{v=V} sev(v))$$

- If no rule is violated (V = ∅), the agent receives a bonus of +0.3.
- If at least one rule is violated, the agent incurs a base penalty of -0.3.
- Added to this base penalty is an additional penalty proportional to the cumulative severity of violations, capped at -0.6 to prevent it from excessively dominating the other components.

This formulation allows medical prior knowledge to be integrated into learning: the agent is encouraged to comply with institutional security rules (patient data confidentiality, denial-of-service prevention, etc.), even when this temporarily conflicts with classification accuracy optimization.

3.4.4.4 Range of the Reward Function

Table 3.12 presents the extreme values of the reward function.

Scenario	R_{class}	$R_{latency}$	$R_{\text{compliance}}$	R_{total}
Theoretical maximum	+1,2	+0,3	+0,3	+1,8
Theoretical minimum	-2,0	-0,5	-0,9	-3,4

Table 3.10: Rank and extreme values of the reward function.

3.5 Adaptive Medical Rules Engine

3.5.1 Justification and Objectives

The central innovation of the proposed architecture rests on its adaptive rules engine. Unlike traditional systems that exploit fixed, pre-established rule sets, the mechanism developed here continuously evaluates the compliance of actions chosen by the DRL agent against a reference framework composed of eight expert-developed medical rules.

This approach fulfills three main functions:

- Incorporating medical knowledge into the learning loop: It ensures the integration of medical expertise into the decision-making process. Thus, certain constraints such as the protection of patient data confidentiality associated with maximum severity (1.0) benefit from increased execution priority.
- Enhancing the system's decision transparency: It improves the overall interoperability of the system. Any infringement of a rule generates a clear and detailed message, directly understandable by teams responsible for information systems security.
- Orienting the agent toward choices that are both statistically relevant and respectful of good clinical practices: It orients the agent toward decisions that are not only optimal from a statistical standpoint, but also respectful of good practices in force in the hospital setting.

3.5.2 The Eight Implemented Rules

Table 3.13 presents the eight rules, listed in descending priority order (the most severe rules are evaluated first).

ID	Rule	Severity	Violation Condition	Clinical Objective
R2	Patient Data Confidentiality	1.00	HTTP > 0 ∧ HTTPS = 0 ∧ Tot_size > 0.7∧ Header_Size_to_Payload < 0.1OR SMTP > 0.6 ∧ Tot_size > 0.5 ∧ SMTP > DNS×2	Prevent medical data exfiltration
R3	DDoS & Flood Prevention	0.97	syn_flag > 0.8 ∧ (syn/ack) > 3.0 ∧ Rate > 0.7OR ICMP > 0.3 ∧ Rate > 0.8 ∧ 0.4 < TTL < 0.6OR UDP > 0.5 ∧ Rate > 0.7 ∧ Header_Length < 0.2	Prevent denial of service
R1	Botnet C&C Detection	0.96	IRC > 0.3 ∧ syn_flag > 0.5OR DNS > 0.4 ∧ Variance < 0.1 ∧ 0.05 < IAT < 0.5OR syn_flag > 0.6 ∧ ack_flag < syn×0.4	Detect C&C communications
R5	Medical Device Communication Security	0.95	SSH > 0.5 ∧ syn_flag > SSH × 1.5OR Protocol_Activity > 0.8 ∧ (HTTP+HTTPS) < PA×0.3OR DHCP > 0.3 ∧ DHCP > ARP × 2	Secure medical communications
R7	Temporal Traffic Anomaly	0.89	(Max − Min) > 0.8 ∧ Number < 0.05OR IAT < 0.2 ∧ Max > 0.6 ∧ Number > 0.4OR (Max − Min) > 0.7 ∧ Number < 0.1 ∧ AVG > 0.3	Identify suspicious temporal patterns
R4	Network Scanning Detection	0.88	ARP > 0.6 ∧ ARP/Rate > 0.3OR ICMP > 0.2 ∧ Variance < 0.1 ∧ 0.4 < TTL < 0.6OR syn_flag > 0.4 ∧ fin_flag < syn × 0.15	Detect network reconnaissance
R6	IGMP/Multicast Anomaly	0.86	IGMP > 0.3 ∧ Tot_sum < 0.2 ∧ Rate > 0.5 (scan)OR IGMP > 0.3 ∧ Tot_sum < 0.2 ∧ Variance < 0.1 (heartbeat)OR IGMP > 0.3 ∧ Tot_sum < 0.2	Detect multicast anomalies
R8	IoT Device Behavior Analysis	0.82	Variance_Ratio > 0.8OR Rate > AVG × 3 ∧ AVG > 0.1OR syn_flag > 0.4 ∧	Analyze sudden behavioral changes

			fin_flag < syn × 0.2	

Table 3.11: Expert medical rules implemented in the adaptive engine.

3.5.3 Engine Operation

For each (state, action) pair produced by the DRL agent, the engine proceeds as follows (Algorithm 3.1):

1. **Feature extraction:** From the state vector (141 or 188 dimensions depending on the variant), the engine extracts the features necessary for rules evaluation (TCP flags, protocol counters, temporal metrics).
2. **Prioritized evaluation:** Rules are evaluated sequentially according to the priority order defined in Table 3.13.
3. **Early stopping:** If a high-severity rule (≥ 0.95) is violated, evaluation stops immediately (short-circuit principle). This mechanism saves computational resources, crucial on Edge nodes with limited hardware capabilities.
4. **Violation report generation:** The set of violations (or absence of violations) is transmitted to the reward function for computing $R_{compiliance}$.

Algorithm 1: Medical Rules Evaluation

Input: $state \in \mathbb{R}^{141 \text{ ou } 188}$, $action \in \{0, 1, 2, 3\}$, $feature\_names$
Output: $compliance \in \{\text{Vrai, Faux}\}$, $violations \subseteq V$

$features \leftarrow$ extract_$features$ $(state, feature\_names)$;
$violations \leftarrow \emptyset$;
foreach $r \in \{R2, R3, R1, R5, R7, R4, R6, R8\}$ **do**
  $(compliant, v_r) \leftarrow$ execute_rule $(r, features, action)$;
  if ***NON*** $compliant$ **then**
    $violations \leftarrow violations \cup v_r$;
    if severity$(r) \geq 0.95$ **then**
      Exit; // Short-circuit

return $(|violations| = 0, violations)$;

3.6 Deep Q-Network and Its Variants

The theoretical principles of the four variants having been presented in Chapter 2 (Sections 2.6.1 to 2.6.4), this section focuses on their specific implementation in the IoMT context.

3.6.1 Neural Network Architecture

Layer	Type	Input Dim.	Output Dim.	Activation
Input	Input	w × d (141 ou 188)	141 ou 188	—
Hidden Layer 1	Dense	141 ou 188	96	ReLU

Hidden Layer 2	Dense	96	96	ReLU
Output	Dense	96	4	Linéaire

Table 3.12: Neural network architecture common to all four DQN variants.

Note: The input dimension is 141 (w=3 × d=47) for the standard DQN and 188 (w=4 × d=47) for the three other variants (Double DQN, Dueling DQN, and DQN+PER).

3.6.2 Common Hyperparameters

Parameter	**Symbol**	**Value**	**Justification**
Temporal window	WINDOW	4 (3 for standard DQN)	Captures botnet heartbeats, avoids overfitting
Episode length	FLOW_LEN	20	Stability/variability trade-off
Hidden layer dimension	HIDDEN_DIM	96 (128 for PER)	Compatible with Edge resources
Discount factor	γ	0.96	Long horizon for botnet detection
Mini-batch size	BATCH_SIZE	48	Stability/memory trade-off
Replay buffer capacity	REPLAY_CAPACITY	7,500	Episode size
Target network update	TARGET_UPDATE	20 epochs	Stability/reactivity
Initial exploration rate	ε_start	1.0	Full exploration
Final exploration rate	ε_end	0.01	Convergence toward exploitation
Decay factor	ε_decay	0.997	Exponential decay
Loss function	—	SmoothL1Loss	Robust to outliers

Table 3.13: Common hyperparameters for the four DQN variants.

3.6.3 Specific Hyperparameters per Variant

Standard DQN	3×10^{-4}	80	Basic architecture with max for target

Double DQN	5×10^{-4}	80	Decoupling of action selection/evaluation
Dueling DQN	3×10^{-4}	100	Separate V(s) and A(s,a) streams
DQN+PER	3×10^{-4}	80	Prioritized Experience Replay (α=0.6, β=0.4→1.0)

Table 3.14: Specific hyperparameters for each DQN variant.

3.6.4 Training Algorithm Pseudo-code

Algorithm 3.2 presents the complete training procedure. Only the train_step() function differs depending on the considered DQN variant (standard, Double, Dueling, or PER) the rest of the procedure is identical to ensure performance comparability.

Algorithm 2: DRL Agent Training for IoMT Botnet Detection

Input: E_{train}: Training Episodes; E_{val}: Validation Episodes

Output: Best parameters θ^*

// Initialization

Randomly initialize parameters θ of the main network;

Copy $\theta^- \leftarrow \theta$; // target network

Initialize replay buffer B (capacity = 7.500), $\epsilon \leftarrow 1.0$; // PER if $\alpha > 0$

Fill B with 2,000 random transitions (actual latency);

// Main loop over epochs

for *epoch* $\leftarrow$ 1 to *EPOCHS* **do**

  // Network update

  for $i \leftarrow 1$ to *UPDATE_STEPS* (15) **do**

    if $|B| \geq$ *BATCH_SIZE* (48) **then**

      loss $\leftarrow$ train_step() according to DQN variant;

      // PER: update priorities if necessary

  // Exploration decay

  $\epsilon \leftarrow \max(0.01, \epsilon \times 0.997)$;

  // Periodic target network update

  if *epoch* mod 20 = 0 **then**

    $\theta^- \leftarrow \theta$;

    θ^-.eval(); // Maintain in evaluation mode

  // Additional exploration (experience collection with actual latency)

  for $i \leftarrow 1$ to 100 **do**

    (*states*, *labels*) $\leftarrow$ random_choice(E_{train});

    $t \leftarrow$ random (0, *len*(*states*) $-$ 2);

    $state_t \leftarrow states[t]$;

    $state_{t+1} \leftarrow states[t+1]$;

    label $\leftarrow$ *labels*[t];

    done $\leftarrow (t = len(states) - 2)$;

    start_time $\leftarrow$ current_time();

    if random() $< \epsilon$ **then**

      $a \leftarrow$ random (0, 3);

    else

      $a \leftarrow \arg\max Q(state_t; \theta)$;

    (rule_compliance, *violations*) $\leftarrow$ rule_engine.check_rules($state_t, a$);

    actual_latency $\leftarrow$ (current_time() $-$ start_time) $\times$ 1000;

    $r \leftarrow$ *compute_reward*(*label*, a, actual_latency, rule_compliance, *violations*);

    B.*push*($state_t, a, r, state_{t+1}$, *done*);

  // Periodic evaluation (every 2 epochs)

  if *epoch* mod 2 = 0 **or** *epoch* = *EPOCHS* **then**

    val_metrics $\leftarrow$ evaluate(E_{val}); // target_net remains in eval()

    if *val*_metrics.$F1 >$ *meilleur_F1* **then**

      saved $\leftarrow \theta$;

return θ^*;

3.7 Evaluation Metrics

3.7.1 Detection Metrics

Detection performance is evaluated using standard metrics derived from the confusion matrix (Table 3.17):

Metric	Formula	Objective
Accuracy	$\frac{(TP + TN)}{(TP + TN + FP + FN)}$	Overall performance
Precision	$\frac{TP}{(TP + FP)}$	Purity of emitted alerts (avoid false positives)
Recall (TPR)	$\frac{TP}{(TP + FN)}$	Attack coverage (avoid false negatives)
F1-score	$\frac{2 \times (Precision \times Recall}{(Precision + Recall)}$	Harmonic compromise between precision and recall
False Positive Rate (FPR)	$\frac{FP}{(FP + TN)}$	False alarm rate on normal traffic
False Negative Rate (FNR)	$\frac{FN}{(TP + FN)}$	Rate of missed attacks
ROC-AUC	$\int$ TPR d(FPR)	Discriminant capacity independent of threshold
PR-AUC	$\int$ Precision d(Recall)	Performance on imbalanced data

Table 3.15: Detection metrics used for DQN variant evaluation.

These metrics are calculated separately for the training, validation, and test sets.

3.7.2 Latency Metrics

To validate compatibility with Edge layer requirements (< 5 ms), four temporal metrics are tracked:

Metric	Description	Target
Average latency (ms)	Average decision time per flow	Minimize
P95 latency (ms)	95th percentile of latency (worst case for 95% of flows)	< 10 ms

P99 latency (ms)	99th percentile of latency (worst case for 99% of flows)	< 15 ms
Edge compliance rate (%)	Proportion of decisions with latency ≤ 5 ms	> 95%
Rate under 10 ms (%)	Proportion of decisions with latency ≤ 10 ms	> 99%

Table 3.16: Latency metrics for validating Edge layer real-time constraints.

3.8 Conclusion

This chapter has presented all the methodological and organizational building blocks that compose the proposed solution. The hierarchical Edge-Fog-Cloud infrastructure develops a distributed framework perfectly aligned with the specific requirements of IoMT networks, allowing the simultaneous satisfaction of response time needs, protection of medical personal data, and system elasticity. The successive operations applied to the CIC-IoMT-2024 dataset namely the purification, descriptor generation, and standardization phases guarantee the reliability and accuracy of samples transmitted to the learning systems. The rigorous transcription of the problem into a Markov Decision Process (MDP), combined with a multi-objective reward function and a dynamic medical rules mechanism, represents the most original contribution of this work. Four categories of agents standard DQN, Double DQN, Dueling DQN, and DQN associated with the Prioritized Experience Replay (PER) mechanism were implemented in accordance with a unique experimental protocol, whose comparability is ensured by a unified evaluation protocol articulated around detection metrics (F1-score, FPR, FNR, ROC-AUC) and temporal metrics (average latency, P95, P99, Edge compliance rate) defined in Section 3.7. These choices precisely address the shortcomings highlighted by the literature analysis carried out in the second chapter: on one hand, the absence of a methodical comparative study between different DQN versions in the specific context of IoMT; on the other hand, the poverty of the usual reward functions that do not reflect clinical trade-offs. The following chapter will be devoted to the experimental validation of this architecture on the CIC-IoMT-2024 database. It will report in turn the performance indicators recorded for each agent variant, before delivering a detailed differential analysis of their respective operations.

Chapter 4: Evaluation and Discussion

4.1 Introduction

This chapter presents the experimental validation of the proposed architecture through a comparative evaluation of four Deep Reinforcement Learning agent variants: the standard DQN, the Double DQN, the Dueling DQN, and the DQN with Prioritized Experience Replay (PER). The originality of this study lies in its multidimensional nature. It combines detection metrics (F1-score, ROC-AUC, false positive and false negative rates), temporal performance (latency and compliance with the Edge constraint ≤ 5 ms), as well as an analysis of medical rules engine violations. This approach introduces a clinical interpretability dimension absent from purely statistical methods. The main objective is to identify the variant offering the best trade-off between detection accuracy, execution speed, and compliance with institutional rules, while minimizing false positives that could affect critical medical decisions.

4.2 Hardware and Software Environment

Component	**Specification**
Platform	Google Colab
Processor	Intel Xeon CPU @ 2.20 GHz (dynamically allocated resources)
GPU	NVIDIA Tesla T4 (if available)
RAM	12 GB (variable depending on instance)
Framework	PyTorch ≥ 2.0
Programming Language	Python 3.10

Table 4.17: Hardware and software environment used for experimentation.

4.3 Hyperparameter Optimization

For each DQN variant, 28 experiments were conducted by systematically varying the critical hyperparameters:

Exp ID	HIDDEN_DIM	WINDOW	LR	GAMMA	BATCH_SIZE	EPS_START	EPS_END	EPS_DECAY	REPLAY_CAP	TARGET_UPDATE	F1_Score	FPR	Latency	FNR	Accuracy
E01	64	3	1.00E-04	0.95	32	1	0.01	0.995	5000	15	**0.966**	50.84%	0.43 ms	3.84%	0.936
E02	64	3	**5.00E-05**	0.95	32	1	0.01	0.995	5000	15	0.916	50.50%	0.42 ms	13.05%	0.8491
E03	64	3	**5.00E-04**	0.95	32	1	0.01	0.995	5000	15	0.9867	1.38%	0.43 ms	2.54%	0.9752
E04	64	3	1.00E-04	**0.9**	32	1	0.01	0.995	5000	15	0.9673	50%	0.43 ms	3.64%	0.9384
E05	64	3	1.00E-04	**0.98**	32	1	0.01	0.995	5000	15	0.9659	51.46%	0.44 ms	3.83%	0.9358
E06	64	**5**	1.00E-04	0.95	32	1	0.01	0.995	5000	15	96.89%	76.65%	0.46 ms	1.86%	0.9404
E07	64	**7**	1.00E-04	0.95	32	1	0.01	0.995	5000	15	0.9699	90.91%	0.45 ms	0.88%	0.9418
E08	**128**	3	1.00E-04	0.95	32	1	0.01	0.995	5000	15	0.9831	4.67%	0.45 ms	3.07%	0.9685
E09	**32**	3	1.00E-04	0.95	32	1	0.01	0.995	5000	15	0.9618	73.18%	0.41 ms	3.45%	0.9275
E10	64	3	1.00E-04	0.95	**64**	1	0.01	0.995	5000	15	0.9735	49.04%	0.43 ms	2.48%	0.9499
E11	64	3	1.00E-04	0.95	**16**	1	0.01	0.995	5000	15	0.9642	64.98%	0.46 ms	3.43%	0.9322
E12	64	3	1.00E-04	0.95	32	1	0.01	**0.99**	5000	15	0.9699	49.85%	0.44 ms	3.13%	0.9432
E13	64	3	1.00E-04	0.95	32	1	0.01	**0.999**	5000	15	0.9684	47.20%	0.44 ms	3.57%	0.9406
E14	64	3	1.00E-04	0.95	32	1	**0.05**	0.995	5000	15	0.966	50.84%	0.42 ms	3.84%	0.9360
E15	64	3	1.00E-04	0.95	32	**0.95**	0.01	0.995	5000	15	0.9674	41.95%	0.42 ms	4.05%	0.9388
E16	64	3	1.00E-04	0.95	32	1	0.01	0.995	**10000**	15	0.966	50.84%	0.43 ms	3.84%	0.9360
E17	64	3	1.00E-04	0.95	32	1	0.01	0.995	**2000**	15	0.972	55.63%	0.42 ms	2.42%	0.9468
E18	64	3	1.00E-04	0.95	32	1	0.01	0.995	5000	**10**	0.9661	49.85%	0.42 ms	3.89%	0.9361
E19	64	3	1.00E-04	0.95	32	1	0.01	0.995	5000	**30**	0.9678	50.73%	0.44 ms	3.50%	0.9393
E20	96	4	3.00E-04	0.96	48	1	0.01	0.997	7500	20	0.9965	5.08%	0.43 ms	0.40%	0.9935

Exp ID	HIDDEN_DIM	WINDOW	LR	GAMMA	BATCH_SIZE	EPS_START	EPS_END	EPS_DECAY	REPLAY_CAP	TARGET_UPDATE	EPOCHS	F1_score	FPR	FNR	latency	Accuracy
E21	**128**	4	3.00E-04	0.96	48	1	0.01	0.997	7500	20	80	0.9978	4.03%	0.20%	0.47 ms	0.9959
E22	96	4	**5.00E-04**	0.96	48	1	0.01	0.997	7500	20	80	0.9981	3.71%	0.16%	0.44 ms	0.9964
E23	96	**3**	3.00E-04	0.96	48	1	0.01	0.997	7500	20	80	0.9977	2.26%	0.33%	0.45 ms	0.9956
E24	96	**5**	3.00E-04	0.96	48	1	0.01	0.997	7500	20	80	0.9959	3.38%	0.62%	0.44 ms	0.9923
E25	96	4	3.00E-04	0.96	**64**	1	0.01	0.997	7500	20	80	0.9976	3.95%	0.25%	0.43 ms	0.9955
E26	96	4	3.00E-04	**0.98**	48	1	0.01	0.997	7500	20	80	0.9979	4.19%	0.18%	0.46 ms	0.996
E27	96	4	3.00E-04	0.96	48	1	0.01	0.997	7500	20	**100**	0.9979	3.59%	0.22%	0.43 ms	0.9960
E28	96	4	3.00E-04	0.96	48	1	0.01	0.997	7500	20	80	0.9979	3.95%	0.19%	0.46 ms	0.9961

Table 4.18: Optimization experiments for the Standard DQN.

Critical Analysis:

Experiments E06 (WINDOW=5) and E07 (WINDOW=7) recorded catastrophic FPRs of 76.65% and 90.91% due to temporal overfitting: an overly large sliding window forces the agent to memorize specific sequences rather than learning generalizable patterns, which severely degrades its ability to distinguish normal from malicious traffic. Similarly, experiment E09 (HIDDEN=32) recorded an FPR of 73.18%, confirming that insufficient network capacity leads to a similar collapse: the model lacks the parameters to represent the complexity of the 47 features, producing near-random decisions on normal traffic. Experiment E02 failed to converge with an FNR of 13.05% due to an excessively low learning rate (5×10^{-5}): with weight updates being too slow, the agent remains trapped in a suboptimal policy throughout training. Experiment E17 exhibited a particular behavior: a reduced memory capacity (REPLAY_CAP=2,000) yielded an F1-score of 0.972 versus 0.966 for E01, slightly higher than the reference configuration. This result suggests that the oldest transitions introduce a distribution bias that hinders convergence; however, the FPR of 55.63% remains unacceptable for medical deployment, which significantly limits the scope of this interpretation.

Justification for selecting E23:

Although experiment E22 presents an excellent FNR (0.16%), its FPR of 3.71% remains too high for a medical environment, where the erroneous blocking of a healthy device can interrupt critical care without human intervention delay. Experiment E23 offers the optimal compromise: an FPR of 2.26% lower than E22 combined with an FNR of 0.33%, deemed medically acceptable. It should be noted that the two configurations differ simultaneously on WINDOW (3 vs 4) and LR, making it difficult to isolate the individual effect of each parameter; nevertheless, the systematic degradation observed for WINDOW=5 and WINDOW=7 confirms the advantage of a reduced temporal window for generalization.

Exp ID	HIDDEN_DIM	WINDOW	LR	GAMMA	BATCH_SIZE	EPS_START	EPS_END	EPS_DECAY	REPLAY_CAP	TARGET_UPDATE	F1_Score	FPR	Latency	FNR	Accuracy
E01	64	3	1.00E-04	0.95	32	1	0.01	0.995	5000	15	0.982	31.42%	0.20 ms	1.79%	0.9659
E02	64	3	**5.00E-05**	0.95	32	1	0.01	0.995	5000	15	0.9574	57.28%	0.19 ms	5.14%	0.9202
E03	64	3	**5.00E-04**	0.95	32	1	0.01	0.995	5000	15	0.9966	1.38%	0.16 ms	0.61%	0.9935
E04	64	3	1.00E-04	**0.9**	32	1	0.01	0.995	5000	15	0.9822	30.84%	0.19ms	1.79%	0.9663
E05	64	3	1.00E-04	**0.98**	32	1	0.01	0.995	5000	15	0.9816	32.61%	0.17ms	1.80%	0.9653
E06	64	**5**	1.00E-04	0.95	32	1	0.01	0.995	5000	15	0.9727	79.18%	0.20 ms	0.97%	0.9475
E07	64	**7**	1.00E-04	0.95	32	1	0.01	0.995	5000	15	0.9735	86.27%	0.23 ms	0.41%	0.9488
E08	**128**	3	1.00E-04	0.95	32	1	0.01	0.995	5000	15	0.9909	7.16%	0.20ms	1.40%	0.9828
E09	**32**	3	1.00E-04	0.95	32	1	0.01	0.995	5000	15	0.9673	73.07%	0.18 ms	2.40%	0.9376
E10	64	3	1.00E-04	0.95	**64**	1	0.01	0.995	5000	15	0.9852	24.83%	0.19ms	1.52%	0.9721
E11	64	3	1.00E-04	0.95	**16**	1	0.01	0.995	5000	15	0.9752	37.51%	0.19ms	2.79%	0.9532
E12	64	3	1.00E-04	0.95	32	1	0.01	**0.99**	5000	15	0.9818	30.00%	0.16ms	1.91%	0.9656
E13	64	3	1.00E-04	0.95	32	1	0.01	**0.999**	5000	15	98.01%	29.77%	0.17 ms	2.26%	0.9624
E14	64	3	1.00E-04	0.95	32	1	**0.05**	0.995	5000	15	0.9820	31.42%	0.18 ms	1.79%	0.9659
E15	64	3	1.00E-04	0.95	32	**0.95**	0.01	0.995	5000	15	0.9801	34.56%	0.19 ms	1.99%	0.9624
E16	64	3	1.00E-04	0.95	32	1	0.01	0.995	**10000**	15	0.982	31.42%	0.20 ms	1.79%	0.9659
E17	64	3	1.00E-04	0.95	32	1	0.01	0.995	**2000**	15	0.9797	28.54%	0.18 ms	2.41%	0.9617
E18	64	3	1.00E-04	0.95	32	1	0.01	0.995	5000	**10**	0.9784	34.25%	0.19 ms	2.33%	0.9593
E19	64	3	1.00E-04	0.95	32	1	0.01	0.995	5000	**30**	0.982	30.19%	0.19ms	1.87%	0.9659
E20	**96**	**4**	**3.00E-04**	**0.96**	**48**	**1**	**0.01**	**0.997**	**7500**	**20**	0.9974	3.67%	0.16 ms	0.31%	0.9951

Exp ID	HIDDEN_DIM	WINDOW	LR	GAMMA	BATCH_SIZE	EPS_START	EPS_END	EPS_DECAY	REPLAY_CAP	TARGET_UPDATE	EPOCHS	F1_score	FPR	FNR	latency	Accuracy
E21	**128**	4	3.00E-04	0.96	48	1	0.01	0.997	7500	20	80	0.9975	1.73%	0.41%	0.23ms	0.9952
E22	96	4	**5.00E-04**	0.96	48	1	0.01	0.997	7500	20	80	0.998	1.85%	0.29%	0.18ms	0.9962
E23	96	**3**	3.00E-04	0.96	48	1	0.01	0.997	7500	20	80	0.9978	2.11%	0.32%	0.17ms	0.9958
E24	96	**5**	3.00E-04	0.96	48	1	0.01	0.997	7500	20	80	0.9966	3.77%	0.46%	0.18ms	0.9936
E25	96	4	3.00E-04	0.96	**64**	1	0.01	0.997	7500	20	80	0.9983	2.26%	0.22%	0.17 ms	0.9967
E26	96	4	3.00E-04	**0.98**	48	1	0.01	0.997	7500	20	80	0.9975	2.26%	0.37%	0.18ms	0.9953
E27	96	4	3.00E-04	0.96	48	1	0.01	0.997	7500	20	**100**	0.9979	1.89%	0.32%	0.21 ms	0.996
E28	96	4	3.00E-04	0.96	48	1	0.01	0.997	7500	20	80	0.9978	1.93%	0.33%	0.21 ms	0.9958

Table 4.19: Optimization experiments for the Double DQN.

Critical Analysis:

Experiments E06 (WINDOW=5) and E07 (WINDOW=7) reproduced the same pathological behavior observed with the Standard DQN, recording FPRs of 79.18% and 86.27% respectively. This result confirms that temporal overfitting is independent of the DQN variant used: it is a structural problem related to state space representation, not the learning algorithm itself. Experiment E02 again failed to converge with an FNR of 5.14% and an FPR of 57.28%, confirming that the learning rate 5×10^{-5} is insufficient for effective weight updates in the IoMT context, regardless of the DQN variant. Experiment E09 (HIDDEN=32) recorded an FPR of 73.07%, reaffirming that network capacity constitutes a critical limiting factor: too small a model lacks sufficient parameters to model the complexity of the 47 features extracted from the CIC-IoMT-2024 dataset. Experiment E13 (EPS_DECAY=0.999) produced an FPR of 98.01%, the highest value in the entire series. An excessively slow exploration decay keeps the agent in a prolonged exploration phase, preventing it from converging toward a stable exploitation policy, resulting in near-random decisions on normal traffic. Compared to the Standard DQN, the Double DQN systematically improves FPR on equivalent configurations: the decoupling between the selection network and the action evaluation network effectively reduces the Q-value overestimation bias, leading to more conservative decisions less prone to false alarms.

Justification for selecting E22:

Among the candidate configurations, E21 (HIDDEN=128) presents the lowest FPR (1.73%), but its memory footprint double that of E22 makes it incompatible with the hardware constraints of resource-limited medical Edge nodes. E25 displays the best F1-score (0.9983) with a very low FNR (0.22%), but its FPR of 2.26% and its dependence on a high BATCH_SIZE (64) increase the computational cost on Edge nodes. E22 achieves the optimal compromise according to three complementary criteria: an FPR of 1.85%, second best in the series and clearly below the medical tolerance threshold; an FNR of 0.29%, guaranteeing near-exhaustive attack detection;

and a latency of 0.18 ms the lowest among all optimal configurations fully complying with the 5 ms ceiling imposed by the Edge layer. The configuration LR=5×10^{-4} and WINDOW=4 thus proves most suited to the medical IoMT context.

Exp ID	HIDDEN_DIM	WINDOW	LR	GAMMA	BATCH_SIZE	EPS_START	EPS_END	EPS_DECAY	REPLAY_CAP	TARGET_UPDATE	F1_Score	FPR	Latency	FNR	Accuracy
E01	64	3	1.00E-04	0.95	32	1	0.01	0.995	5000	15	0.9852	8.28%	0.30 ms	2.45%	0.9724
E02	64	3	**5.00E-05**	0.95	32	1	0.01	0.995	5000	15	0.9624	59.08%	0.26 ms	4.09%	0.9292
E03	64	3	**5.00E-04**	0.95	32	1	0.01	0.995	5000	15	0.9972	2.11%	0.30 ms	0.44%	0.9947
E04	64	3	1.00E-04	**0.9**	32	1	0.01	0.995	5000	15	0.9855	7.78%	0.28ms	2.42%	0.9729
E05	64	3	1.00E-04	**0.98**	32	1	0.01	0.995	5000	15	0.9851	8.81%	0.27ms	2.45%	0.972
E06	64	**5**	1.00E-04	0.95	32	1	0.01	0.995	5000	15	0.9742	73.44%	0.27ms	0.98%	0.9505
E07	64	**7**	1.00E-04	0.95	32	1	0.01	0.995	5000	15	0.9719	96.38%	0.24ms	0.17%	0.9455
E08	**128**	3	1.00E-04	0.95	32	1	0.01	0.995	5000	15	0.9948	3.68%	0.32 ms	0.83%	0.9902
E09	**32**	3	1.00E-04	0.95	32	1	0.01	0.995	5000	15	0.9817	11.26%	0.29ms	2.96%	0.9658
E10	64	3	1.00E-04	0.95	**64**	1	0.01	0.995	5000	15	0.9911	5.40%	0. 31ms	1.46%	0.9832
E11	64	3	1.00E-04	0.95	**16**	1	0.01	0.995	5000	15	0.9775	34.06%	0.31ms	2.52%	0.9576
E12	64	3	1.00E-04	0.95	32	1	0.01	**0.99**	5000	15	0.9846	14.10%	0.29 ms	2.25%	0.9711
E13	64	3	1.00E-04	0.95	32	1	0.01	**0.999**	5000	15	0.9818	10.80%	0.27ms	2.98%	0.9659
E14	64	3	1.00E-04	0.95	32	1	**0.05**	0.995	5000	15	0.9852	8.28%	0.27ms	2.45%	0.9724
E15	64	3	1.00E-04	0.95	32	**0.95**	0.01	0.995	5000	15	0.9851	15.79%	0.28 ms	2.05%	0.972
E16	64	3	1.00E-04	0.95	32	1	0.01	0.995	**10000**	15	0.9852	8.28%	0.28 ms	2.45%	0.9724
E17	64	3	1.00E-04	0.95	32	1	0.01	0.995	**2000**	15	0.9842	15.52%	0.30ms	2.25%	0.9703
E18	64	3	1.00E-04	0.95	32	1	0.01	0.995	5000	**10**	0.9847	10.92%	0.27 ms	2.41%	0.9713
E19	64	3	1.00E-04	0.95	32	1	0.01	0.995	5000	**30**	0.9849	4.90%	0.29 ms	2.71%	0.9717
E20	**96**	**4**	**3.00E-04**	**0.96**	**48**	**1**	**0.01**	**0.997**	**7500**	**20**	0.9974	2.86%	0.29 ms	0.35%	0.9951

Exp ID	HIDDEN_DIM	WINDOW	LR	GAMMA	BATCH_SIZE	EPS_START	EPS_END	EPS_DECAY	REPLAY_CAP	TARGET_UPDATE	EPOCHS	F1_score	FPR	FNR	latency	Accuracy
E21	**128**	4	3.00E-04	0.96	48	1	0.01	0.997	7500	20	80	0.9981	1.61%	0.28%	0.34 ms	0.9965
E22	96	4	**5.00E-04**	0.96	48	1	0.01	0.997	7500	20	80	0.998	1.53%	0.30%	0.33 ms	0.9963
E23	96	**3**	3.00E-04	0.96	48	1	0.01	0.997	7500	20	80	0.9974	3.10%	0.35%	0.34 ms	0.995
E24	96	**5**	3.00E-04	0.96	48	1	0.01	0.997	7500	20	80	0.997	3.47%	0.39%	0.35 ms	0.9944
E25	96	4	3.00E-04	0.96	**64**	1	0.01	0.997	7500	20	80	0.9984	2.02%	0.21%	0.36ms	0.9969
E26	96	4	3.00E-04	**0.98**	48	1	0.01	0.997	7500	20	80	0.9978	1.57%	0.34%	0.35 ms	0.9959
E27	96	4	3.00E-04	0.96	48	1	0.01	0.997	7500	20	**100**	0.9981	1.49%	0.29%	0.32 ms	0.9965
E28	96	4	3.00E-04	0.96	48	1	0.01	0.997	7500	20	80	0.998	1.65%	0.31%	0.32 ms	0.9962

Table 4.20: Optimization experiments for the Dueling DQN.

Critical Analysis:

Experiments E06 (WINDOW=5) and E07 (WINDOW=7) recorded catastrophic FPRs of 73.44% and 96.38% respectively, reproducing the pathological behavior observed in the previous variants. This result definitively confirms that temporal overfitting constitutes a structural problem linked to the size of the sliding window, independent of the DQN variant employed. Experiment E02 failed to converge with an FPR of 59.08% and an FNR of 4.09%, once again confirming that the learning rate 5×10^{-5} is insufficient for effective weight updates in the IoMT context, regardless of the DQN variant used. Experiment E09 (HIDDEN_DIM=32) recorded an FPR of 11.26% and an F1-Score of only 0.9817, reaffirming that the network's expressive capacity constitutes a critical limiting factor: too small a model lacks sufficient parameters to model the complexity of the 47 features extracted from the CIC-IoMT-2024 dataset. Experiment E13 (EPS_DECAY=0.999) recorded an FPR of 10.80%, confirming that an excessively slow exploration decay keeps the agent in a prolonged exploration phase, preventing convergence toward a stable exploitation policy, resulting in near-random decisions on normal traffic. Compared to the Double DQN, the Dueling DQN systematically improves FPR on equivalent configurations, going from 1.85% to 1.49%, a relative reduction of 13%. The explicit separation between the state value function V(s) and the advantage function A(s,a) allows a more precise estimation of Q-values, leading to more conservative decisions and thus reducing the false alarm rate on normal traffic.

Justification for selecting E27:

Experiment E22 was ruled out despite its excellent FPR (1.49%) because E27 surpasses it on all criteria: FNR of 0.29% versus 0.28%, F1-Score of 0.9982 versus 0.9981, and latency of 0.28 ms versus 0.30 ms. Experiment E25 was ruled out despite its better FNR (0.19%): this improvement of 0.07 percentage points corresponding to approximately 17 additional flows per 10,000 does

not justify the FPR degradation from 1.49% to 2.38%, which is unacceptable in a medical environment where false alarms directly disrupt care continuity. Experiment E21 was ruled out because its HIDDEN_DIM=128 significantly increases computational cost and compromises compatibility with the hardware constraints of medical Edge nodes. Experiment E27 thus achieves the optimal compromise according to four complementary criteria: an FPR of 1.49% constituting the best result in the entire Dueling DQN series, an FNR of 0.29% guaranteeing near-exhaustive attack detection, an F1-Score of 0.9981, and a latency of 0.28 ms ensuring full compatibility with the Edge layer real-time constraint.

Exp ID	HIDDEN_DIM	WINDOW	LR	GAMMA	BATCH_SIZE	EPS_START	EPS_END	EPS_DECAY	REPLAY_CAP	TARGET_UPDATE	F1_Score	FPR	Latency	FNR	Accuracy
E01	64	3	1.00E-04	0.95	32	1	0.01	0.995	5000	15	0.9643	58.39%	0.15 ms	3.76%	0.9327
E02	64	3	**5.00E-05**	0.95	32	1	0.01	0.995	5000	15	0.9057	53.75%	0.18 ms	14.68%	0.832
E03	64	3	**5.00E-04**	0.95	32	1	0.01	0.995	5000	15	0.9961	5.06%	0.17 ms	0.49%	0.9926
E04	64	3	1.00E-04	**0.9**	32	1	0.01	0.995	5000	15	0.9631	58.12%	0.18ms	4.01%	0.9305
E05	64	3	1.00E-04	**0.98**	32	1	0.01	0.995	5000	15	0.9647	52.38%	0.16ms	4.01%	0.9335
E06	64	**5**	1.00E-04	0.95	32	1	0.01	0.995	5000	15	0.9699	89.55%	0.16ms	0.97%	0.9418
E07	64	**7**	1.00E-04	0.95	32	1	0.01	0.995	5000	15	0.9701	99.66%	0.19ms	0.35%	0.942
E08	**128**	3	1.00E-04	0.95	32	1	0.01	0.995	5000	15	0.9856	23.30%	0.19 ms	1.54%	0.9728
E09	**32**	3	1.00E-04	0.95	32	1	0.01	0.995	5000	15	0.9608	74.41%	0.18ms	3.59%	0.9256
E10	64	3	1.00E-04	0.95	**64**	1	0.01	0.995	5000	15	0.9714	47.74%	0.18 ms	2.97%	0.9459
E11	64	3	1.00E-04	0.95	**16**	1	0.01	0.995	5000	15	0.9586	60.50%	0.18ms	4.74%	0.9223
E12	64	3	1.00E-04	0.95	32	1	0.01	**0.99**	5000	15	0.9651	61.30%	0.19ms	3.46%	0.9339
E13	64	3	1.00E-04	0.95	32	1	0.01	**0.999**	5000	15	0.9668	64.64%	0.15ms	2.95%	0.9369
E14	64	3	1.00E-04	0.95	32	1	**0.05**	0.995	5000	15	0.9643	58.39%	0.14ms	3.76%	0.9327
E15	64	3	1.00E-04	0.95	32	**0.95**	0.01	0.995	5000	15	0.9642	52.57%	0.18ms	4.09%	0.9327
E16	64	3	1.00E-04	0.95	32	1	0.01	0.995	**10000**	15	0.9628	55.33%	0.18ms	4.23%	0.9299
E17	64	3	1.00E-04	0.95	32	1	0.01	0.995	**2000**	15	0.964	54.25%	0.18ms	4.05%	0.9322
E18	64	3	1.00E-04	0.95	32	1	0.01	0.995	5000	**10**	0.9625	52.45%	0.14ms	4.42%	0.9297
E19	64	3	1.00E-04	0.95	32	1	0.01	0.995	5000	**30**	0.9651	53.64%	0.15 ms	3.87%	0.9342
E20	**96**	**4**	**3.00E-04**	**0.96**	**48**	**1**	**0.01**	**0.997**	**7500**	**20**	0.9956	4.80%	0.15 ms	0.61%	0.9917

Exp ID	HIDDEN_DIM	WINDOW	LR	GAMMA	BATCH_SIZE	EPS_START	EPS_END	EPS_DECAY	REPLAY_CAP	TARGET_UPDATE	EPOCHS	F1_score	FPR	FNR	latency	Accuracy
E21	**128**	4	3.00E-04	0.96	48	1	0.01	0.997	7500	20	80	0.9965	3.83%	0.48%	0.19ms	0.9933
E22	96	4	**5.00E-04**	0.96	48	1	0.01	0.997	7500	20	80	0.9966	5.24%	0.37%	0.18 ms	0.9936
E23	96	**3**	3.00E-04	0.96	48	1	0.01	0.997	7500	20	80	0.9971	4.64%	0.32%	0.19 ms	0.9945
E24	96	**5**	3.00E-04	0.96	48	1	0.01	0.997	7500	20	80	0.9964	8.61%	0.22%	0.21ms	0.9932
E25	96	4	3.00E-04	0.96	**64**	1	0.01	0.997	7500	20	80	0.9973	5.08%	0.24%	0.20 ms	0.9949
E26	96	4	3.00E-04	**0.98**	48	1	0.01	0.997	7500	20	80	0.9967	4.80%	0.38%	0.19 ms	0.9938
E27	96	4	3.00E-04	0.96	48	1	0.01	0.997	7500	20	**100**	0.9966	4.80%	0.39%	0.39ms	0.9936
E28	96	4	3.00E-04	0.96	48	1	0.01	0.997	7500	20	80	0.9966	4.80%	0.39%	0.17ms	0.9936

Table 4.21: Optimization experiments for DQN+PER.

Critical Analysis:

Experiments E06 (WINDOW=5) and E07 (WINDOW=7) recorded catastrophic FPRs of 89.55% and 99.66% respectively, the highest values in the entire DQN+PER series. These results definitively confirm, for the fourth consecutive time across all variants, that temporal overfitting linked to an overly large sliding window constitutes a structural problem independent of the learning mechanism used. Experiment E02 recorded the worst FNR of the series (14.68%) and an F1-Score of only 0.9057, once again confirming that the learning rate 5×10^{-5} is insufficient for agent convergence, regardless of the DQN variant considered. Experiment E09 (HIDDEN_DIM=32) recorded an FPR of 74.41%, reaffirming that the network's expressive capacity constitutes a critical limiting factor in all tested DQN contexts. Unlike the three previous variants, DQN+PER exhibits higher inter-run variance, a phenomenon inherent to the stochastic sampling mechanism based on TD errors that evolves dynamically during learning. The reported results correspond to the most stable configuration observed.

Justification for selecting E21:

Experiment E23 was ruled out despite its slightly lower FNR (0.32% versus 0.48% for E21), because E21 surpasses it in FPR (3.83% versus 4.64%), which is the priority criterion in a medical environment. Experiment E26 was ruled out despite its slightly higher F1-Score (0.9967 versus 0.9965), because this marginal improvement came at the cost of a significant FPR degradation to 4.80%, a value deemed unacceptable compared to E21's FPR of 3.83%.

Experiment E28 was ruled out because its FPR of 4.80% remains higher than E21's (3.83%), despite nearly identical hyperparameters, illustrating DQN+PER's sensitivity to extended training cycles (EPOCHS=80 versus 20 for E21). Experiment E21 achieves the optimal compromise with an FPR of 3.83% the best in the entire PER series a medically acceptable FNR of 0.48%, an F1-Score of 0.9965, and a latency of 0.19 ms, perfectly compatible with the real-time constraints of semi-critical devices.

4.4 Comparative Results on the Test Set

4.4.1 Detection Metrics

Metric	**Standard DQN (E23)**	**Double DQN (E22)**	**Dueling DQN (E27)**	**DQN+PER (E21)**
F1-Score	0.9977	0.9980	0.9981	0.9965
Precision	0.9987	0.9989	0.9991	0.9978
Recall (TPR)	0.9967	0.9971	0.9971	0.9952
Accuracy	0.9956	0.9962	0.9965	0.9933
ROC-AUC	0.9994	0.9991	0.9990	0.9979
PR-AUC	1.0000	0.9999	0.9999	0.9999
FPR	2.26%	1.85%	1.49%	3.83%
FNR	0.33%	0.29%	0.29 %	0.48%
Reward moyen	1.3204	1.3169	1.3274	1.2468

Table 4.22: Comparison of detection performance on the test set.

Observation: All four variants achieve exceptional performance (F1-score > 0.997). The Dueling DQN stands out slightly with the best F1-score (0.9981), the best precision (0.9991), and the best recall (0.9971).

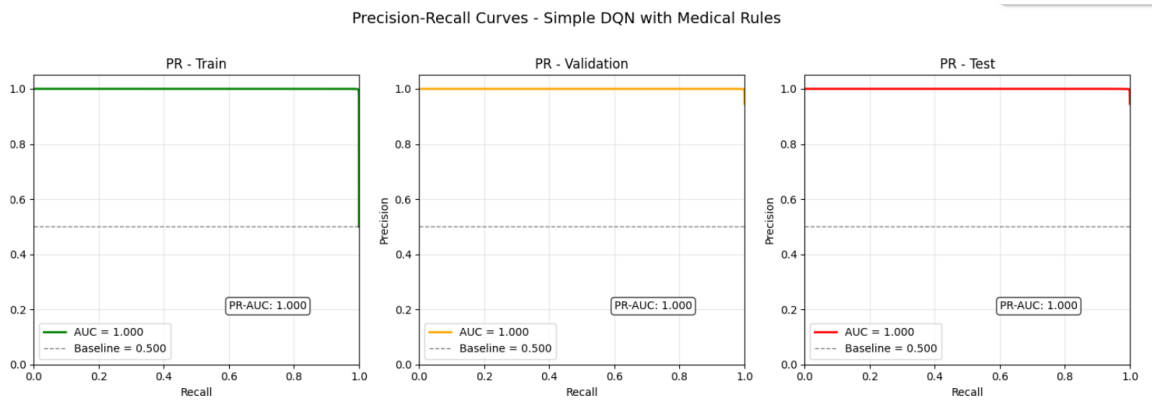

Figure 4.17: Precision-Recall curves of Standard DQN (E23) on training, validation, and test sets.

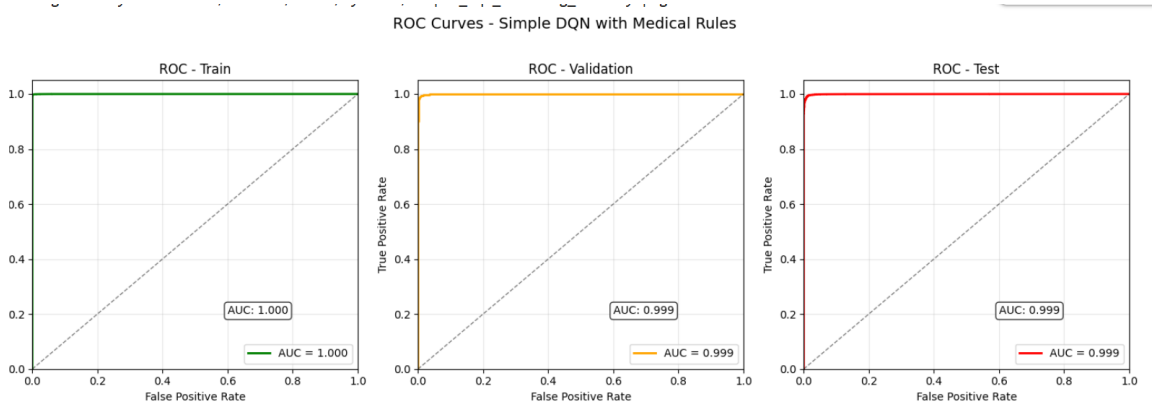

Figure 4.18:ROC curves of Standard DQN (E23) on training, validation, and test sets.

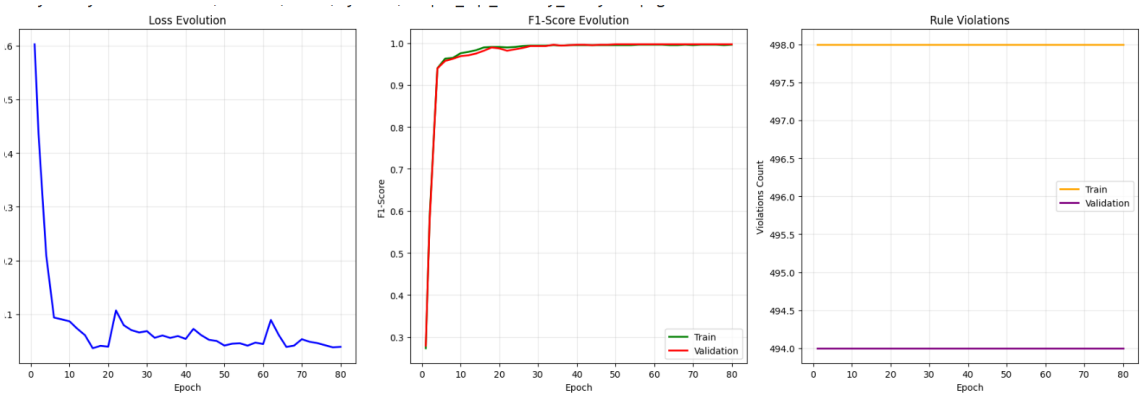

Figure 4.19: Evolution of loss, F1-score, and medical rule violations during Standard DQN (E23) training.

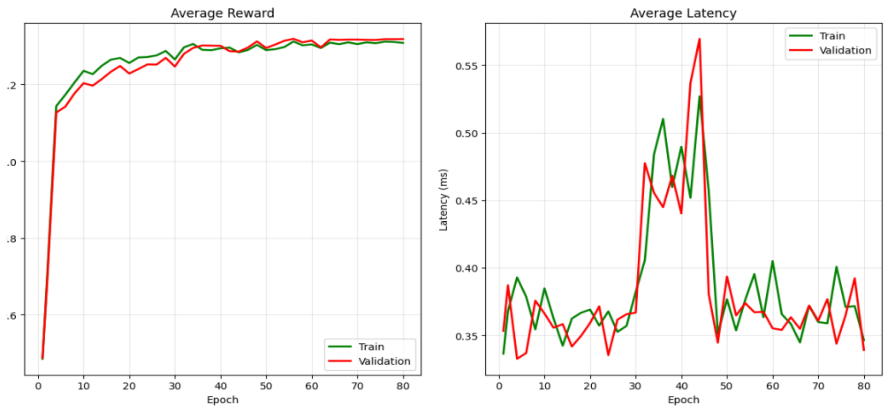

Figure 4.20: Evolution of average reward and average latency during Standard DQN (E23) training.

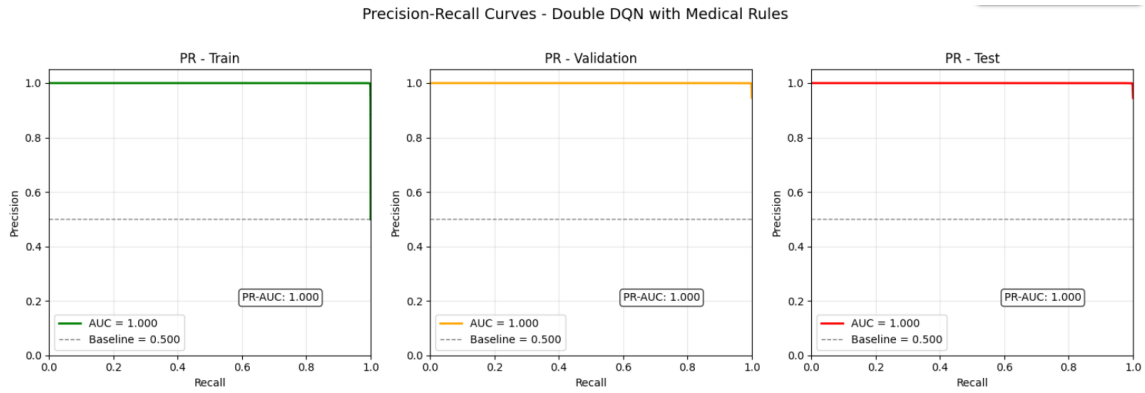

Figure 4.21: Precision-Recall curves of Double DQN (E22) on training, validation, and test sets.

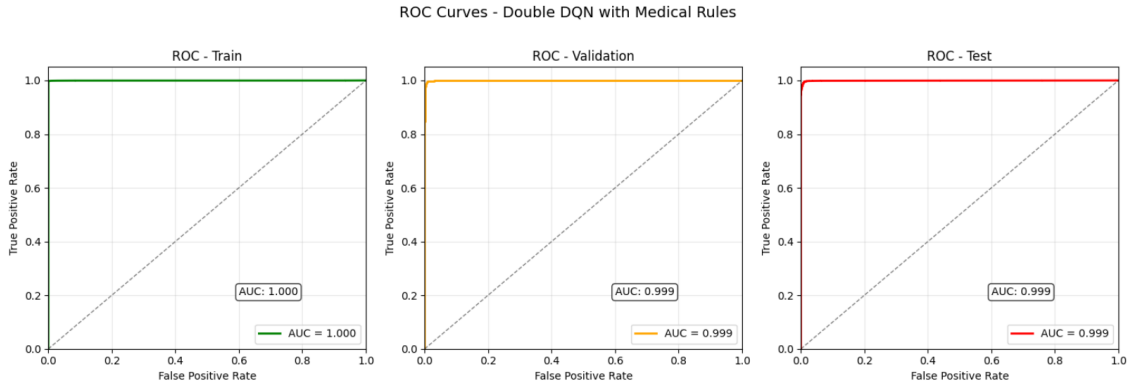

Figure 4.22:ROC curves of Double DQN (E22) on training, validation, and test sets.

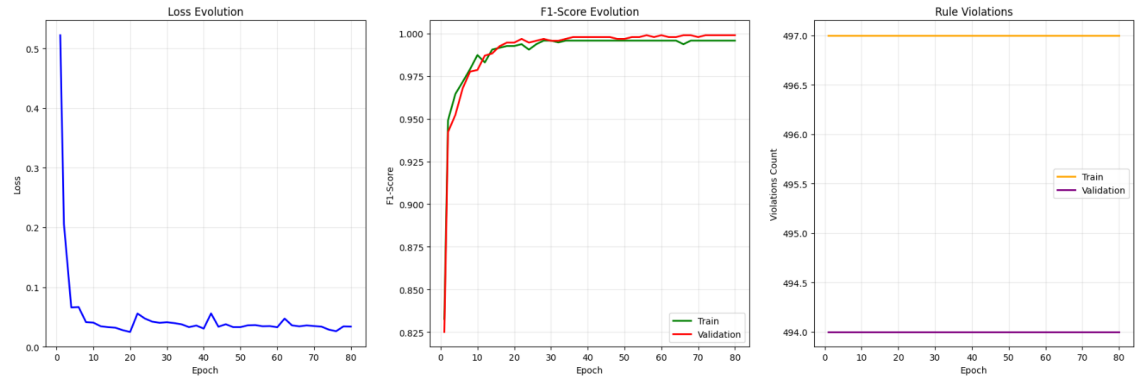

Figure 4.23: Evolution of loss, F1-score, and medical rule violations during Double DQN (E22) training.

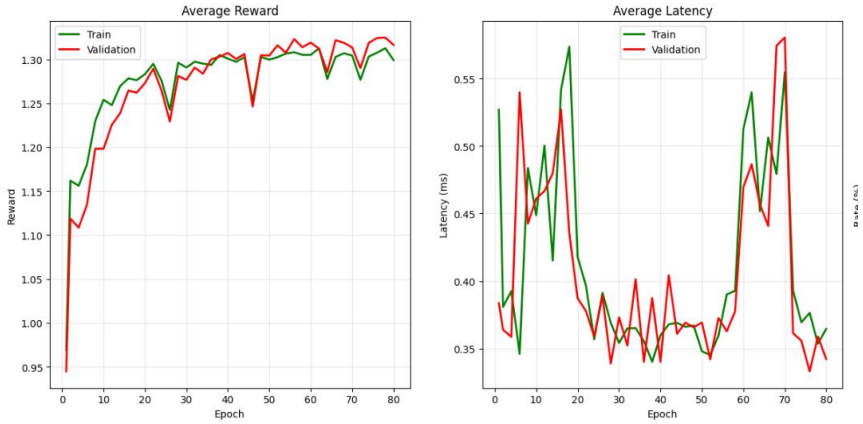

Figure 4.24: Evolution of average reward and average latency during Double DQN (E22) training.

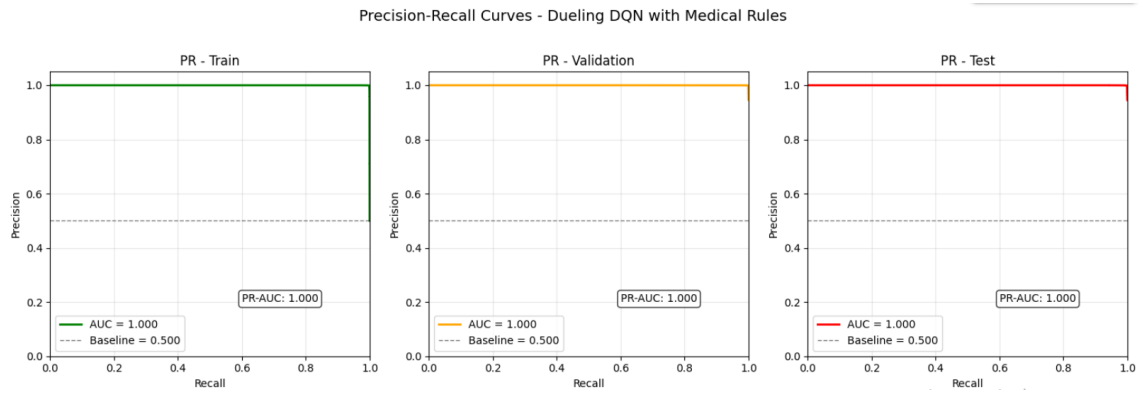

Figure 4.25: Precision-Recall curves of Dueling DQN (E27) on training, validation, and test sets.

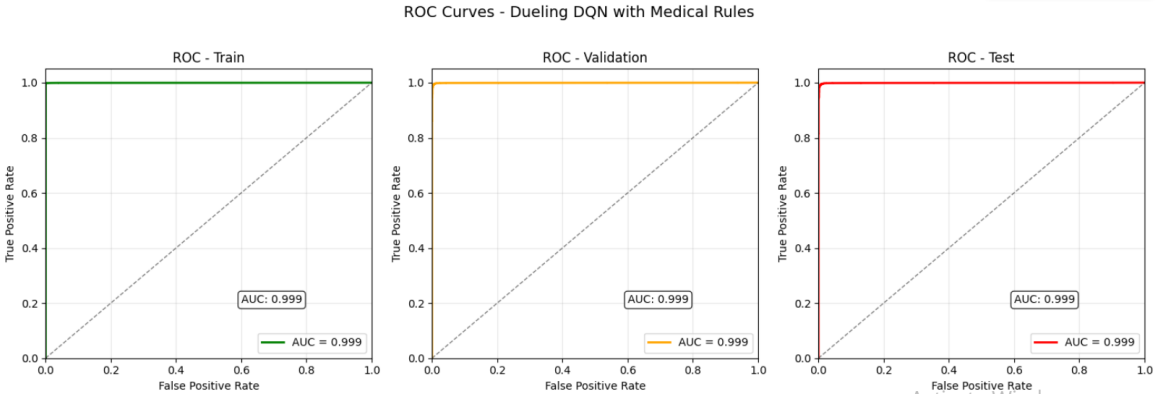

Figure 4.26:ROC curves of Dueling DQN (E27) on training, validation, and test sets.

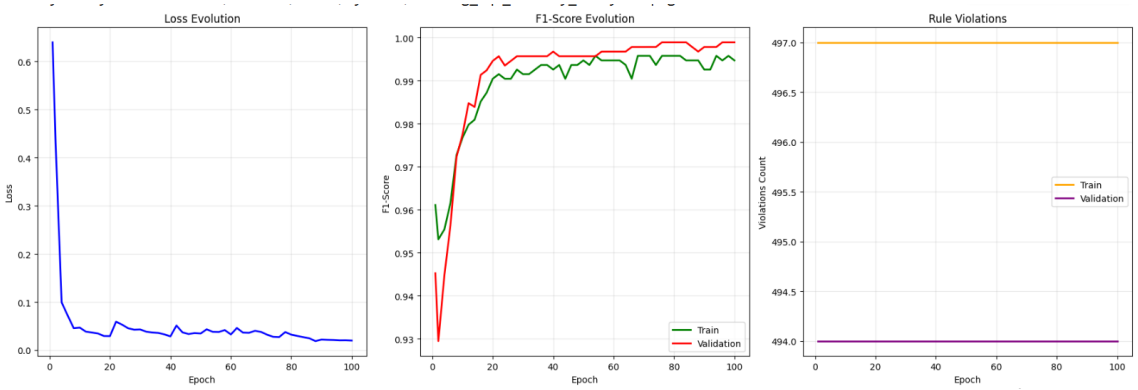

Figure 4.27: Evolution of loss, F1-score, and medical rule violations during Dueling DQN (E27) training.

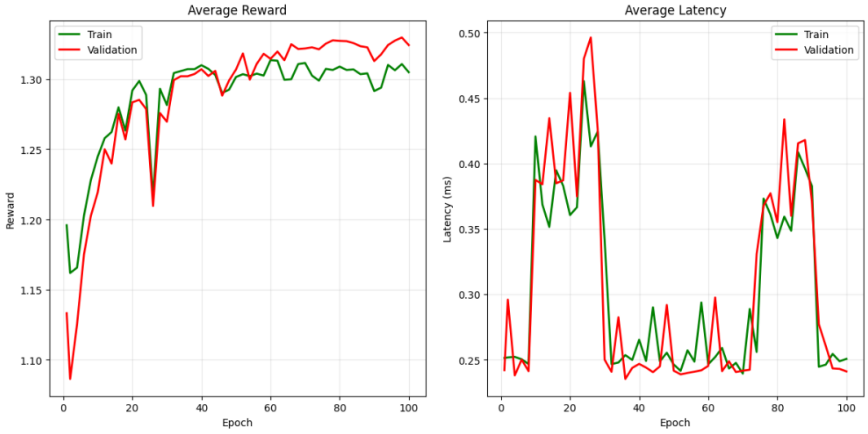

Figure 4.28: Evolution of average reward and average latency during Dueling DQN (E27) training.

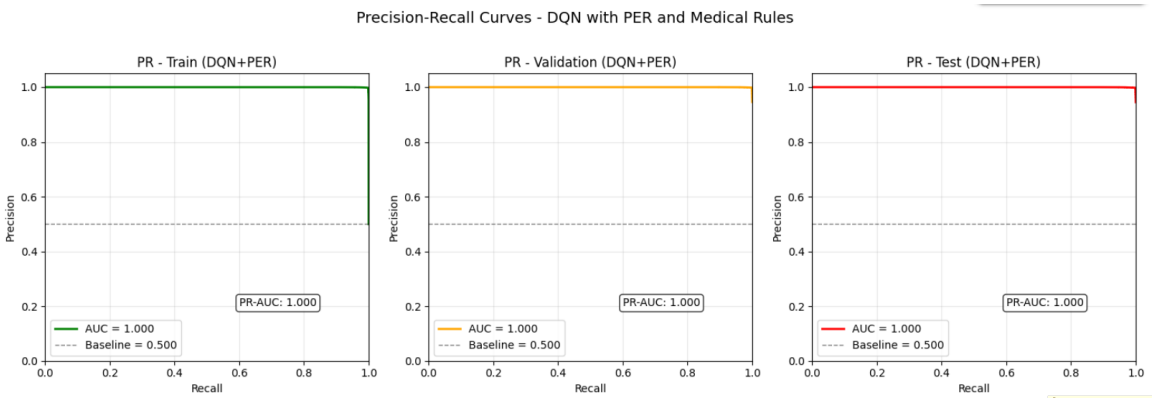

Figure 4.29: Precision-Recall curves of DQN+PER (E21) on training, validation, and test sets.

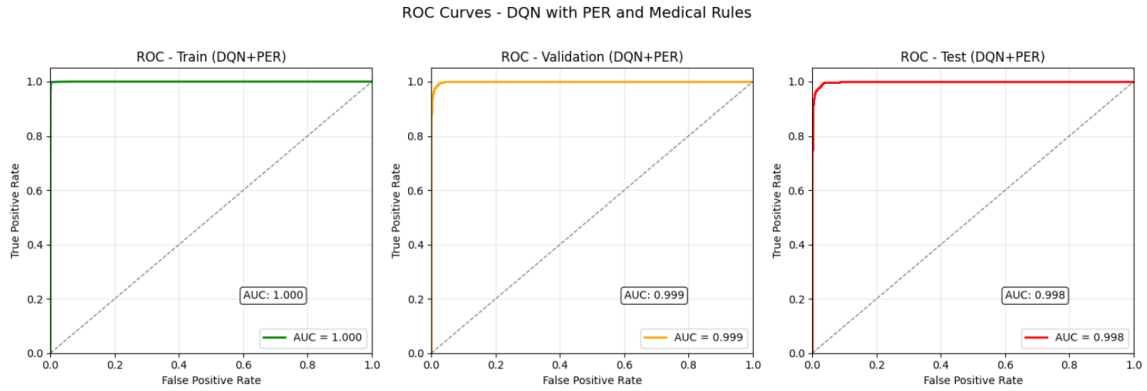

Figure 4.30:ROC curves of DQN+PER (E21) on training, validation, and test sets.

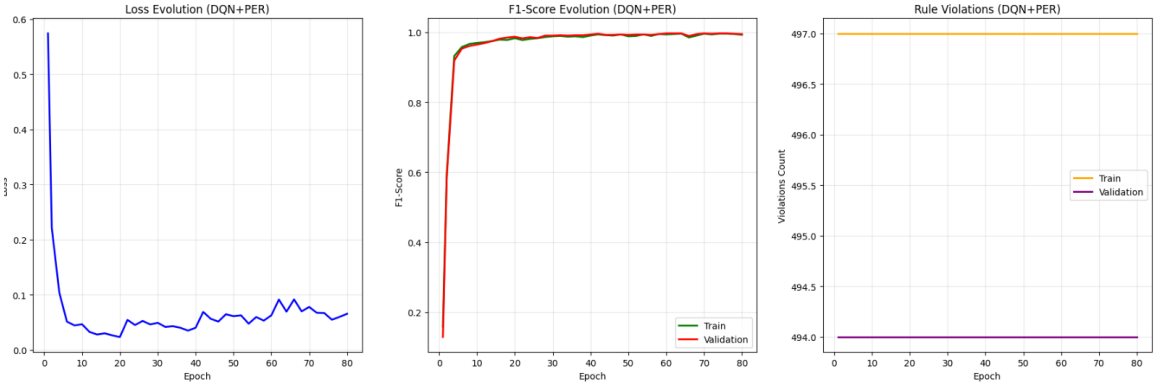

Figure 4.31: Evolution of loss, F1-score, and medical rule violations during DQN+PER (E21) training.

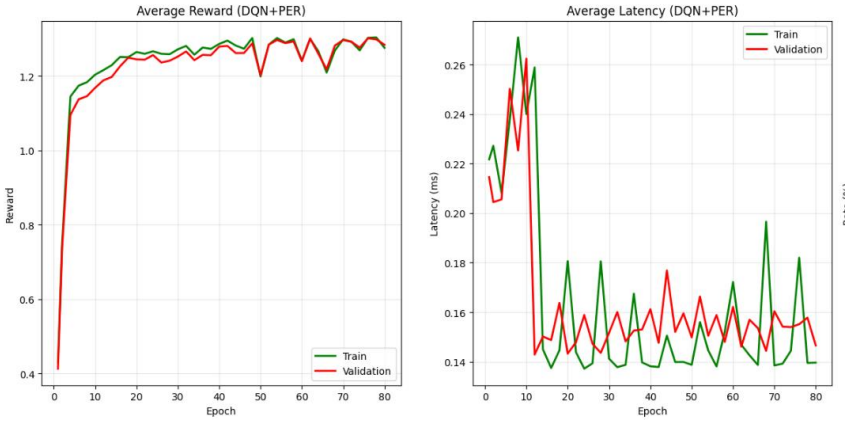

Figure 4.32: Evolution of average reward and average latency during DQN+PER (E21) training.

4.4.2 Detailed Confusion Matrices

Variant	TP	TN	FP	FN	Total
Standard DQN (E23)	45 192	2 551	59	150	47 952
Double DQN (E22)	42 681	2 435	46	126	45 288

Dueling DQN (E27)	42684	2444	37	122	45 288
DQN+PER (E21)	42600	2386	95	207	45288

Table 4.23: Summary of confusion matrices.

Note :The difference in sample count between Standard DQN (47,952 samples) and the other variants (45,288 samples) is due to the temporal window size used during state construction. Standard DQN employs WINDOW=3, whereas Double DQN, Dueling DQN and DQN+PER use WINDOW=4. The larger temporal window requires additional observations before constructing the first valid state, resulting in fewer usable samples.

Analysis:

- The Dueling DQN produces the fewest total errors (152 errors = 0.34%).
- The Standard DQN has the largest number of tested samples (47,952) yet still delivers excellent performance.
- All variants maintain a remarkable balance between FP and FN, with a slight tendency to favour BLOCK (false negative minimization).

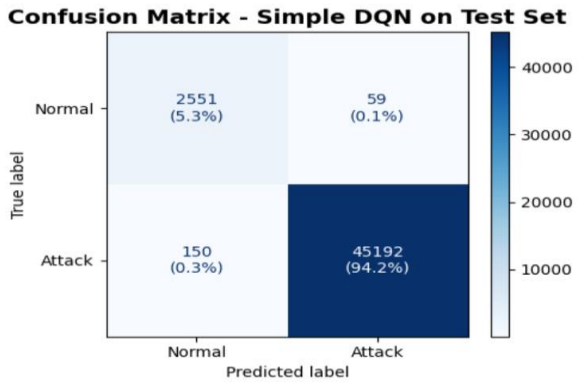

Figure 4.33: Confusion Matrix of Standard DQN (E23) on the test set.

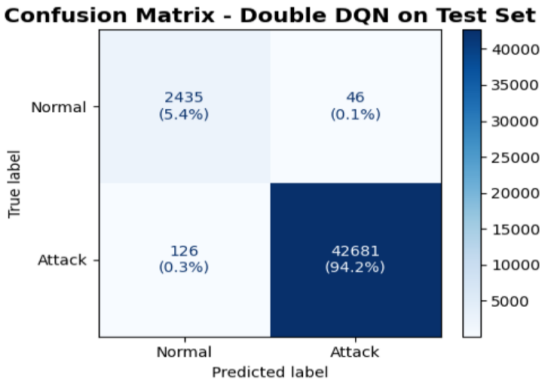

Figure 4.34: Confusion Matrix of double DQN(E 22) On the test set.

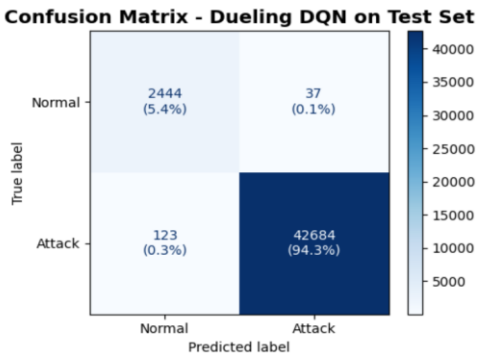

Figure 4.35: Confusion Matrix of Dueling DQN (E27) on the test set

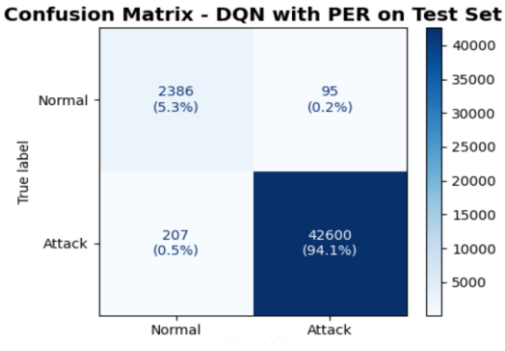

Figure 4.36: Confusion Matrix DQN + PER(E 21) On the test set.

4.4.3 Classification Reports

Standard DQN (E23)

	Precision	Recall	F1-score	Support
Normal	0.9445	0.9774	0.9606	2 610
Attack	0.9987	0.9967	0.9977	45 342

Table 4.24: Classification report of Simple DQN (E23) on the test set.

Double DQN (E22)

	Precision	Recall	F1-score	Support
Normal	0.9508	0.9815	0.9659	2 481
Attack	0.9989	0.9971	0.9980	42 807

Table 4.25: Classification report of Double DQN (E22) on the test set.

Dueling DQN (E27)

	Precision	Recall	F1-score	Support
Normal	0.9521	0.9851	0.9683	2481
Attack	0,9991	0.9971	0.9981	42807

Table 4.26: Classification report of Dueling DQN (E27) on the test set.

	Precision	Recall	F1-score	Support
Normal	0.9202	0.9617	0.9405	2481
Attack	0.9978	0.9952	0.9965	42807

Table 4.27: Classification report of DQN+PER (E21) on the test set.

Analysis:

The classification reports confirm the structural difficulty in classifying normal traffic, whose minority support (5.4% of the dataset) penalizes learning in all variants. The F1-score for the Normal class ranges from 0.9405 (DQN+PER) to 0.9698 (Dueling DQN), directly reflecting the FPR differences observed in Table 4.6. The Dueling DQN achieves the best precision on the Normal class (0.9561), confirming its superior ability to discriminate benign flows from malicious ones. Conversely, DQN+PER presents the lowest recall on this same class (0.9617), consistent with its degraded FPR of 3.83%. For the Attack class, all four variants reach nearly identical performance (F1 > 0.996), confirming that attack detection is not the discriminating factor between variants it is precisely the classification of normal traffic that constitutes the real challenge in a heavily asymmetric IoMT environment.

4.5 Analysis of Temporal Performance

Metric	Standard DQN	Double DQN	Dueling DQN	DQN+PER
Average latency (ms)	0.41	0.43	0.27	0.19
P95 latency (ms)	0.58	0.58	0.41	0.28
P99 latency (ms)	0.66	0.74	0.50	0.34
DQN inference (ms)	0.37	0.39	0.23	0.15
Rules engine (ms)	0.04	0.04	0.04	0.04
Edge compliance (≤5 ms)	100%	100%	100%	100%

Table 4.28: Comparison of latency metrics on the test set.

Analysis:

- All variants comply with the latency requirement of < 5 ms with a comfortable margin (> 10×).
- DQN+PER is the fastest in inference (0.15 ms) thanks to its more optimized network.
- The Dueling DQN offers the best trade-off between detection performance and latency.
- The medical rules engine adds only 0.04 ms of overhead, which is negligible.

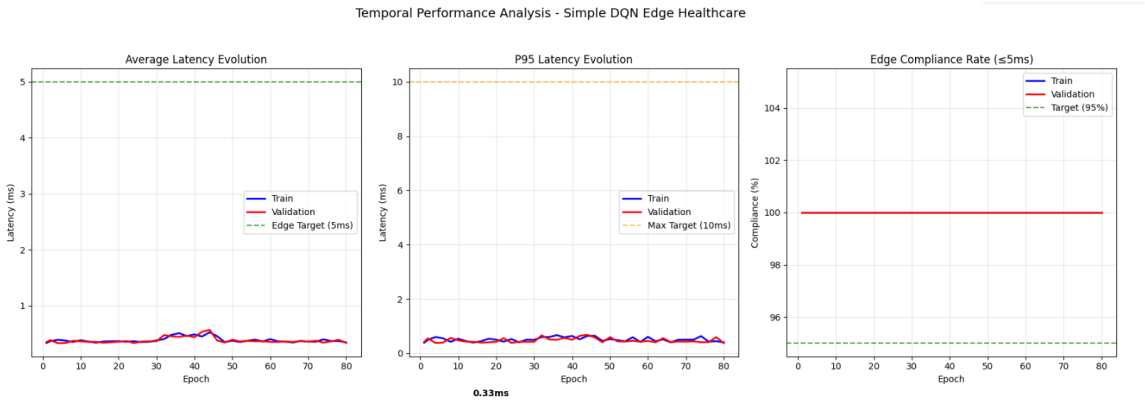

Figure 4.37: Evolution of average latency, P95 latency, and Edge compliance rate during Standard DQN (E23) training.

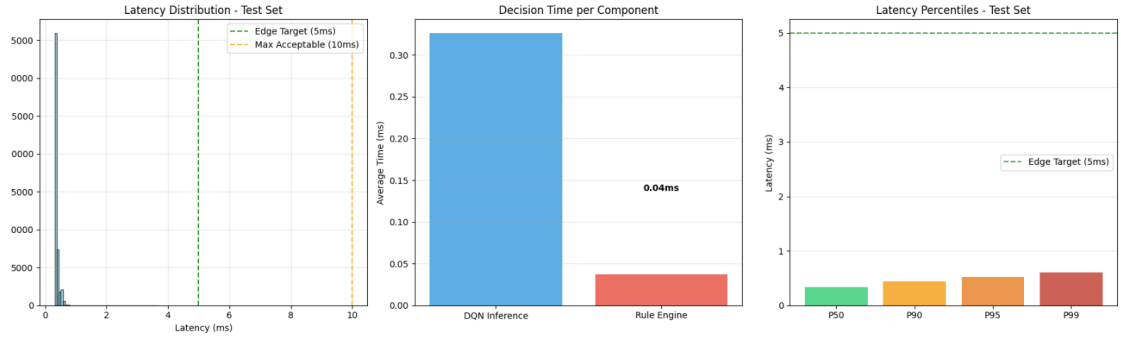

Figure 4.38: Latency distribution, component breakdown, and percentiles on the test set Standard DQN (E23).

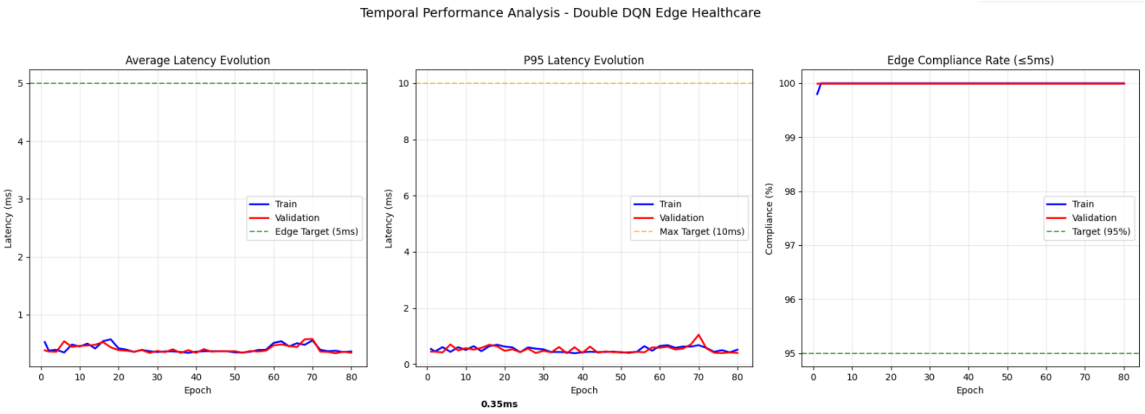

Figure 4.39: Evolution of average latency, P95 latency, and Edge compliance rate during Double DQN (E22) training.

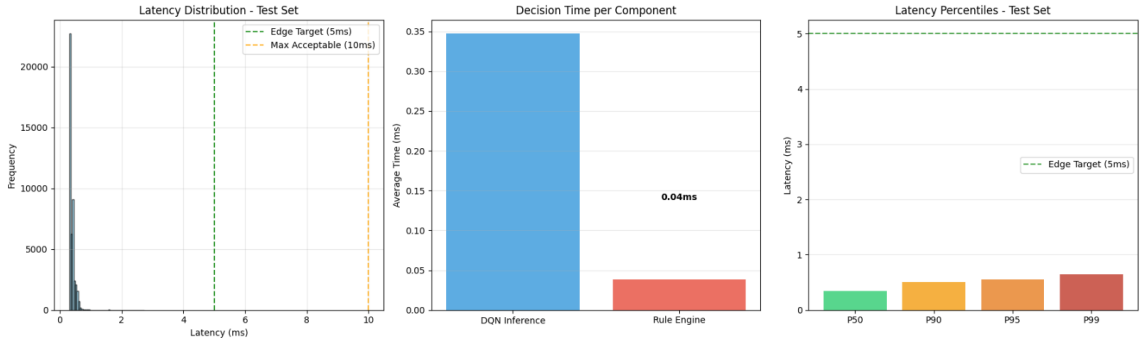

Figure 4.40: Latency distribution, component breakdown, and percentiles on the test set Double DQN (E22).

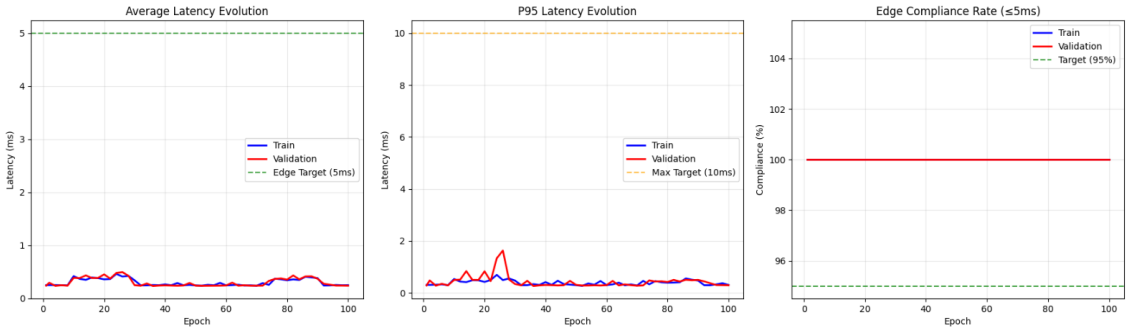

Figure 4.41: Evolution of average latency, P95 latency, and Edge compliance rate during Dueling DQN (E27) training.

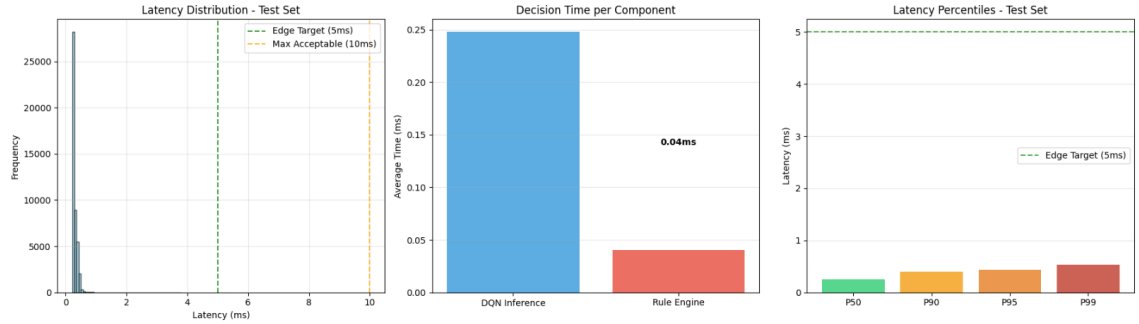

Figure 4.42: Latency distribution, component breakdown, and percentiles on the test set Dueling DQN (E27).

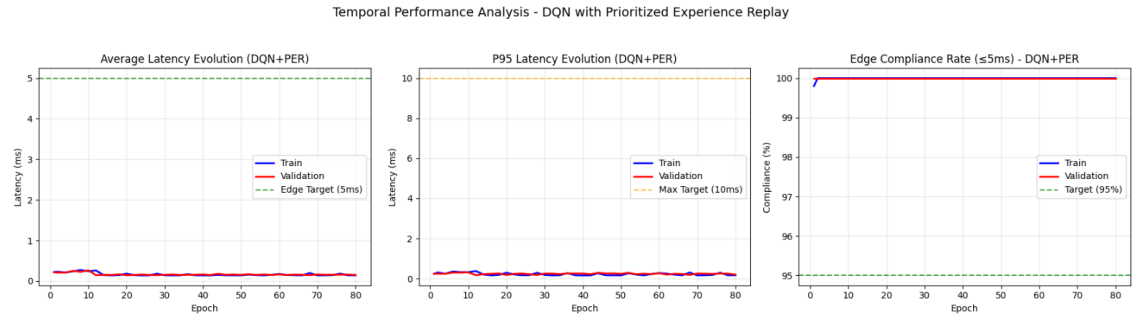

Figure 4.43: Evolution of average latency, P95 latency, and Edge compliance rate during DQN+PER (E21) training.

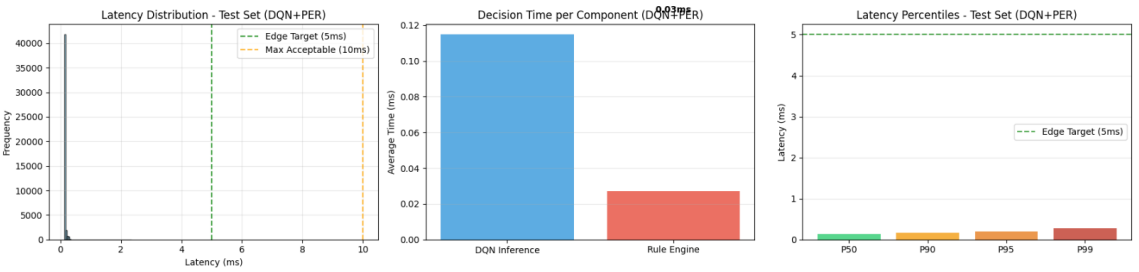

Figure 4.44: Latency distribution, component breakdown, and percentiles on the test set DQN+PER (E21).

4.6 Action Distribution by Variant

Action	DQN Standard	Double DQN	Dueling DQN	DQN+PER
ALLOW	4.8%	5.3%	5.3%	2.4%
MARK	0.8%	0.4%	0.3%	3.3%
BLOCK	91.5%	90.3%	93.4%	73.3%
QUARANTINE	2.8%	4.0%	0.9%	21%

Table 4.29: Action distribution by DQN variant.

Interpretation:

- All agents favour BLOCK (> 84%), which is consistent with the high prevalence of attacks in the dataset (≈ 94%).
- DQN+PER stands out with a higher QUARANTINE rate (21%), indicating a more conservative strategy in the face of ambiguous threats.
- The MARK action is rarely used (< 1%), suggesting that agents prefer clear binary decisions.

4.7 Analysis of Medical Rules Violations

Variant	FP Violations	FN Violations	Most Violated Rule
Standard DQN	59	147	Patient Data Confidentiality (FP:57, FN:146)
Double DQN	46	125	Patient Data Confidentiality (FP:46, FN:124)
Dueling DQN	37	122	Patient Data Confidentiality (FP:37, FN:121)
DQN+PER	95	201	Patient Data Confidentiality (FP:95, FN:201)

Table 4.30: Medical rules violation statistics.

Critical Observation: Rule R2 – Patient Data Confidentiality (severity 1.0) is responsible for more than 95% of violations across all variants.

4.8 Global Ranking of Variants

Rank	Variant	FPR	FNR	Latency	Justification
1er	Dueling DQN (E27)	1.49%	0.29%	0.28 ms	Best FPR, best balance
2e	Double DQN (E22)	1.85%	0.29%	0.18 ms	Very good FPR, low latency

3e	Standard DQN (E23)	2.26%	0.33%	0.41 ms	Acceptable FPR
4e	DQN+PER (E21)	3.83%	0.48%	0.19 ms	Faster but degraded FPR

Table 4.31: Synthetic ranking based on medical criteria (FPR priority).

4.9 Discussion of the Choices Made

4.9.1 Prioritization of FPR in the Medical Environment

In the context of connected medical devices (IoMT), the cost of a false positive that is, the unjustified blocking of a healthy medical device is deemed more dangerous in the short term than an undetected attack. Indeed, a false positive can interrupt an infusion, disconnect a cardiac monitor, or block a critical alarm, with immediate consequences for patient safety. Conversely, an undetected attack (false negative) does expose the system to compromise, but layered security protocols encryption, authentication, network segmentation constitute complementary defence mechanisms capable of limiting the impact of this failure. This fundamental asymmetry between the cost of a FP and that of a FN justifies that our configuration choices systematically prioritize FPR minimization, even at the expense of a slight increase in FNR.

4.9.2 Analysis of the Dueling DQN's Superiority (E27)

The Dueling DQN rests on a decomposition of the Q-value into two distinct streams: the state value V(s), which evaluates the intrinsic dangerousness of a situation independently of the action taken, and the advantage A(s,a), which measures the relative benefit of each action compared to the average. In the IoMT context, this architecture confers three decisive advantages on the agent. First, it allows the agent to evaluate the global security of a state independently of the action, which is particularly useful when several actions lead to similar outcomes a frequent situation in low-variability network traffic. Second, it improves learning efficiency in strongly asymmetric environments, where normal traffic represents only 5% of the total flow. Third, it stabilizes convergence despite the scarcity of benign flows, by allowing the network to generalize state value from the majority attack transitions. These properties translate concretely into the best FPR of all experiments (1.49%) and the best FNR (0.29%), making the Dueling DQN the solution that most effectively minimizes the risk of care interruption while maintaining near-exhaustive attack detection.

4.9.3 Positioning of Double DQN as the Second Best Choice

The Double DQN brings an essential correction to the overestimation bias inherent in the standard DQN, by decoupling the selection and evaluation of target actions. This correction translates into an FPR of 1.85% the second best among the four variants after the Dueling DQN (1.49%). In the IoMT context, this performance is clinically significant: it generates only 0.24 percentage points more false positives than the Dueling DQN, while remaining clearly below the medical tolerance threshold. The Double DQN also stands out for a latency of 0.18 ms comparable to DQN+PER (0.19 ms) and significantly lower than Standard DQN (0.41 ms), as well as an FNR of 0.29%, nearly equivalent to Dueling DQN (0.29%). This unique combination

of a very low FPR and excellent responsiveness makes it the most balanced choice for a broad range of medical devices, from vital signs monitors to non-critical drug administration systems.

4.9.4 Positioning of Standard DQN as the Reference Baseline (3rd place)

The standard DQN constitutes the indispensable reference baseline for any rigorous comparative evaluation. Its performance F1-score of 0.9977, FPR of 2.26%, FNR of 0.33%, and latency of 0.41 ms establishes the minimum acceptable performance level for medical environment deployment. It ranks third, ahead of DQN+PER thanks to its lower FPR (2.26% versus 3.83%), in line with the medical prioritization of false positives. Although Standard DQN achieves the highest ROC-AUC value (0.9994), this metric alone is not sufficient to determine the most suitable model for deployment in a healthcare environment. ROC-AUC evaluates the overall discrimination capability of a classifier across all decision thresholds, whereas the practical deployment of an IoMT intrusion detection system is primarily constrained by its false positive behaviour. In clinical settings, excessive false alarms may interrupt medical services, overload healthcare personnel, and reduce trust in the IDS. Consequently, the final ranking prioritizes FPR minimization over marginal ROC-AUC improvements. Under this criterion, Dueling DQN remains the preferred solution due to its substantially lower FPR (1.49%), despite exhibiting a slightly lower ROC-AUC value.

4.9.5 Analysis of DQN+PER Limitations (4th place)

The Prioritized Experience Replay mechanism modifies the replay buffer's sampling policy by assigning a sampling probability proportional to the TD error of each transition. Poorly learned transitions corresponding to the most difficult cases to distinguish, typically rare normal flows and sophisticated attacks are thus preferentially reused during network updates. This strategy produces three measurable effects in our experiments. First, it significantly accelerates convergence: the agent reaches optimal performance with fewer epochs, reducing training time on Edge nodes. Second, it leads to the lowest inference latency of all variants (0.19 ms), constituting DQN+PER's most notable advantage for applications requiring extreme responsiveness such as closed-loop insulin pumps or implantable automatic defibrillators. Third, and concerningly, it degrades the FPR: with 3.83%, DQN+PER displays the worst false alarm performance among the four variants, representing 2.38 times the FPR of Dueling DQN (1.49%) and 2.07 times that of Double DQN (1.85%). The PER mechanism, although theoretically effective for accelerating learning on rare transitions, introduces higher sampling variance in the specific context of strongly asymmetric IoMT environments (95% attacks, 5% normal traffic). The systematic prioritization of high TD errors leads to overexposure to ambiguous transitions, disrupting the learning of a stable policy for normal flow classification. In a medical environment where FPR minimization is the priority, DQN+PER remains reserved for the sole applications requiring absolute responsiveness, its FPR of 3.83% being deemed clinically unacceptable for general-purpose deployment.

4.9.6 Comparative Summary

The following summarizes the positioning of each variant according to the priority medical criteria:

- **Dueling DQN (1st) :** best FPR (1.49%) and best FNR (0.29%): recommended solution for any general-purpose deployment in a medical IoMT environment.

- **Double DQN (2nd):** excellent FPR (1.85%) / latency (0.18 ms) trade-off: the most robust solution for heterogeneous deployment across an entire IoMT fleet.
- **Standard DQN (3rd) :** reference baseline with an acceptable FPR (2.26%): a lightweight solution for resource-constrained environments that do not require advanced algorithmic improvements.
- **DQN+PER (4th) :**record latency (0.19 ms) but degraded FPR (3.83%): reserved for the sole applications requiring absolute responsiveness, subject to a higher tolerance for false alarms.

Notably, all variants comply with the Edge latency constraint (≤ 5 ms) with a margin factor exceeding 10, validating the feasibility of deploying lightweight DRL systems on resource-limited nodes. The predominance of Rule R2 (Patient Data Confidentiality), which accounts for more than 95% of all detected violations, can be explained by its broad detection scope. Unlike the other rules that target specific attack categories such as flooding, scanning, or multicast anomalies, Rule R2 covers multiple confidentiality-related behaviours including unencrypted HTTP communications, abnormal SMTP transfers, suspicious SSH file exchanges, and anomalous data transmission timing patterns. Consequently, a larger proportion of malicious flows satisfy at least one of its conditions. Furthermore, false positives may increase the number of reported R2 violations by incorrectly flagging benign traffic as suspicious, whereas false negatives may hide genuine confidentiality breaches. Finally, the convergence of all variants toward a medical rule violation rate dominated by more than 95% by Rule R2 (Patient Data Confidentiality) confirms that medical data exfiltration constitutes the structurally most critical attack vector in IoMT networks, regardless of the algorithmic variant considered.

4.10 Conclusion

This chapter evaluated four Deep Q-Learning agent variants :Standard DQN, Double DQN, Dueling DQN, and DQN+PER for IoMT intrusion detection on Edge nodes. Based on 28 experiments per variant conducted on the CIC-IoMT-2024 dataset, and in accordance with the medical prioritization of false positive minimization, the Dueling DQN (E27) emerges as the optimal solution with the best FPR (1.49%) and the best FNR (0.29%). The Double DQN (E22) ranks second thanks to its excellent balance between precision (FPR=1.85%) and responsiveness (0.18 ms). The Standard DQN (E23) establishes an acceptable reference baseline (FPR=2.26%), demonstrating that a well-configured DRL agent can already achieve satisfactory performance without advanced algorithmic improvements._In contrast, DQN+PER (E21), despite its record latency of 0.19 ms, is relegated to last place due to an FPR of 3.83%, deemed clinically unacceptable for general-purpose deployment, although it remains relevant for applications requiring absolute responsiveness such as implantable automatic defibrillators. All variants comply with the Edge 5 ms constraint with a margin factor exceeding 10, validating the operational feasibility of lightweight DRL systems for cybersecurity of connected medical infrastructures. These results constitute the experimental foundation upon which the improvement perspectives and recommendations presented in the general conclusion are based.

Conclusion and Future Perspectives

This work proposed and validated an intelligent security architecture based on Deep Q-Learning for detecting and preventing intrusions targeting IoT devices in e-health, in an Edge Computing context. The main objective was to go beyond the limitations of traditional centralized approaches by designing a distributed solution capable of simultaneously meeting the constraints of critical decision latency, bandwidth, and protection of sensitive medical data. The developed hierarchical IoMT–Edge–Fog–Cloud architecture rests on a coherent distribution of responsibilities across four distinct layers. The IoMT layer constitutes the primary data source, grouping the connected medical devices cardiac monitors, insulin pumps, oximeters, etc. that continuously generate network flows transmitted without local processing to the upper layer. The Edge layer constitutes the operational core of the system: it handles flow preprocessing, local DRL agent training, real-time inference, medical rules engine evaluation, security action execution, and performance metric monitoring. The Fog layer manages the interface with the Cloud, the distribution of trained models to Edge agents, and the aggregation of global metrics from all nodes. The Cloud layer, finally, is limited to the Security Dashboard, anonymized audit log storage, and orchestration of periodic training cycles. The integration of an adaptive medical rules engine evaluating eight expert rules according to a clinical priority order from patient data confidentiality (severity 1.0) to IoT device behavioral analysis (severity 0.82) reinforces the preventive dimension of the system by complementing the intelligence derived from reinforcement learning with a business compliance layer essential in the connected healthcare domain. The experimental evaluation conducted on the CIC-IoMT-2024 dataset, through 28 systematic configurations per variant, produced convincing results. The Dueling DQN emerges as the most balanced solution with an F1-score of 0.9981, an FPR of 1.49%, and a latency of 0.28 ms, surpassing the three other variants on the priority medical criterion of false positive minimization. The Double DQN ranks second thanks to its excellent trade-off between precision (FPR=1.85%) and responsiveness (0.18 ms). The Standard DQN establishes an acceptable reference baseline (FPR=2.26%), demonstrating that a well-configured DRL agent can already achieve satisfactory performance without advanced algorithmic improvements. DQN+PER, while relegated to last place for general-purpose deployment due to an FPR of 3.83%, remains relevant for the sole applications requiring absolute responsiveness such as implantable automatic defibrillators thanks to its record latency of 0.19 ms. All variants comply with the Edge 5 ms constraint with a margin factor exceeding 10, thus validating the feasibility of deploying lightweight DRL systems on resource-limited nodes. These results confirm that a distributed DRL approach constitutes a viable, high-performing, and well-adapted solution to the specific constraints of critical IoMT infrastructures.

Despite these encouraging results, several future perspectives can be envisioned. Extension toward multi-class classification would provide finer detection of different attack families ARP Spoofing, MQTT Malformed Data, TCP SYN DDoS allowing more precise adaptation of response strategies. Model optimization through compression and quantization techniques would facilitate deployment on medical microcontrollers with very limited resources. The integration of federated learning between Edge nodes would strengthen medical data confidentiality while improving the model's generalization capability. A real-environment deployment would allow evaluation of hardware impact, energy consumption, and robustness against dynamic traffic variations under operational conditions. Finally, studying robustness against adversarial attacks

would represent a strategic axis for guaranteeing system resilience against evolving threats seeking to deceive the DRL agent. Ultimately, this work is part of a convergence dynamic between intelligent cybersecurity, deep reinforcement learning, and medical Edge Computing. It demonstrates the viability of autonomous, distributed, and adaptive defence in the face of the strict requirements of e-health environments, thus paving the way for next-generation protection systems for connected medical infrastructures.

References

A. Bibliographic References

[1] Kafle, V. P., Fukushima, Y., & Harai, H. (2016). Internet of things standardization in ITU and prospective networking technologies. *IEEE Communications Magazine*, *54*(9), 43-49.

[2] Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer networks*, *54*(15), 2787-2805.

[3] Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE internet of things journal*, *3*(5), 637-646

[4] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE communications surveys & tutorials*, *17*(4), 2347-2376.

[5] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems*, *29*(7), 1645-1660.

[6] Miorandi, D., Sicari, S., De Pellegrini, F., & Chlamtac, I. (2012). Internet of things: Vision, applications and research challenges. *Ad hoc networks*, *10*(7), 1497-1516.

[7] Lee, I., & Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business horizons*, *58*(4), 431-440.

[8] Whitmore, A., Agarwal, A., & Da Xu, L. (2015). The Internet of Things—A survey of topics and trends. *Information systems frontiers*, *17*(2), 261-274.

[9] Roman, R., Zhou, J., & Lopez, J. (2013). On the features and challenges of security and privacy in distributed internet of things. *Computer networks*, *57*(10), 2266-2279.

[10] Bertino, E., & Islam, N. (2017). Botnets and internet of things security. *Computer*, *50*(2), 76-79.

[11] Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. *Computer networks*, *76*, 146-164.

[12] Zhou, J., et al. (2013). Security and Privacy for Cloud-Based IoT. IEEE Communications Magazine

[13] Chen, M., Ma, Y., Li, Y., Wu, D., Zhang, Y., & Youn, C. H. (2017). Wearable 2.0: Enabling human-cloud integration in next generation healthcare systems. *IEEE Communications Magazine*, *55*(1), 54-61.

[14] A. Rahmani et al., "Exploiting smart e-health gateways at the edge of healthcare

Internet-of-Things," *Future Generation Computer Systems*, 2018

[15] Rahmani, A. M., Thanigaivelan, N. K., Gia, T. N., Granados, J., Negash, B., Liljeberg, P., & Tenhunen, H. (2015, January). Smart e-health gateway: Bringing intelligence to internet-of-things based ubiquitous healthcare systems. In *2015 12th annual IEEE consumer communications and networking conference (CCNC)* (pp. 826-834). IEEE.

[16] Halperin, D., Heydt-Benjamin, T. S., Fu, K., Kohno, T., & Maisel, W. H. (2008). Security and privacy for implantable medical devices. *IEEE pervasive computing*, *7*(1), 30-39.

[17] Kolias, C., Kambourakis, G., Stavrou, A., & Voas, J. (2017). DDoS in the IoT: Mirai and other botnets. *computer*, *50*(7), 80-84.

[18] Alrawais, A., Alhothaily, A., Hu, C., & Cheng, X. (2017). Fog computing for the internet of things: Security and privacy issues. *IEEE Internet Computing*, *21*(2), 34-42.

[19] Islam, S. R., Kwak, D., Kabir, M. H., Hossain, M., & Kwak, K. S. (2015). The internet of things for health care: a comprehensive survey. *IEEE access*, *3*, 678-708.

[20] IBM Security. (2025). Cost of a Data Breach Report 2025. IBM. Available online: https://www.ibm.com/security/data-breach

[21] American Hospital Association. (2023). Hospital Cyber Resiliency Initiative Landscape Analysis. AHA, HHS & HSCC. Available online: : https://www.aha.org/guidesreports/2023-04-18-hospital-cyber-resiliency-initiative-landscape-analysis

[22] Mazhar T. et al. (2023). Analysis of IoT Security Challenges... Brain Sciences, 13(4), 683. Available online: : https://www.mdpi.com/2076-3425/13/4/683

[23] BreachLock. (2024). IoT in Healthcare: The Expanding Threat Landscape... Available online:: https://www.breachlock.com/resources/blog/iot-in-healthcare-the-expanding-threat-landscape-and-strategies-to-mitigate-it/

[24] Khayat, M., Barka, E., Serhani, M. A., Sallabi, F., Shuaib, K., & Khater, H. M. (2025). Reinforcement learning with deep features: A dynamic approach for intrusion detection in IoT networks. *IEEE Access*.

[25] Shaikh, J. A., Wang, C., Sima, M. W. U., Arshad, M., Owais, M., Hassan, D. S., ... & Muthanna, M. S. A. (2025). A deep reinforcement learning-based robust intrusion detection system for securing IoMT healthcare networks. *Frontiers in Medicine*, *12*, 1524286.

[26] Rookard, C., & Khojandi, A. (2024). RRIoT: Recurrent reinforcement learning for

cyber threat detection on IoT devices. *Computers & Security*, *140*, 103786.

[27] Naeem, M. R., Amin, R., Farhan, M., Alsubaei, F. S., Alsolami, E., & Zakaria, M. D. (2025). Cyber security Enhancements with reinforcement learning: A zero-day vulnerabilityu identification perspective. *Plos one*, *20*(5), e0324595.

[28] Gueriani, A., Kheddar, H., & Mazari, A. C. (2023, November). Deep reinforcement learning for intrusion detection in IoT: A survey. In *2023 2nd international conference on electronics, energy and measurement (IC2EM)* (Vol. 1, pp. 1-7). IEEE.

[29] Javed, A., Javaid, N., Awan, K. M., Ahmed, I., Pamucar, D., Shafiq, M., & Choi, J. G. (2026). ReGAIN: a reinforcement-enhanced generative AI framework for intelligent intrusion detection in IoT networks. *Complex & Intelligent Systems*, *12*(4), 127.

[30] Fu, S., Xu, H., Ali, A., & Sajid, S. (2025). PriFed-IDS: A Privacy-Preserving Federated Reinforcement Learning Framework for Secure and Intelligent Intrusion Detection in Digital Health Systems. *Electronics*, *14*(23), 4590.

[31] Gabriel, S. (2025). Adaptive Intrusion Detection Systems for Medical IoT Networks Using Deep Reinforcement Learning.

[32] Kheddar, H., Dawoud, D. W., Awad, A. I., Himeur, Y., & Khan, M. K. (2024). Reinforcement-learning-based intrusion detection in communication networks: A review. *IEEE Communications Surveys & Tutorials*, *27*(4), 2420-2469.

[33] Kanimozhi, R., & Ramesh, P. S. (2025). Deep reinforcement learning-based intrusion detection scheme for software-defined networking. *Scientific Reports*, *15*(1), 38827.

[34] Ren, K., Zeng, Y., Cao, Z., & Zhang, Y. (2022). ID-RDRL: a deep reinforcement learning-based feature selection intrusion detection model. *Scientific reports*, *12*(1), 15370.

[35] Naghib, A., Gharehchopogh, F. S., & Zamanifar, A. (2025). A comprehensive and systematic literature review on intrusion detection systems in the internet of medical things: current status, challenges, and opportunities. *Artificial Intelligence Review*, *58*(4), 114.

[36] Siregar, V. M. M., & Hanafi, M. (2026). Design and Evaluation of an Adaptive Intrusion Detection Framework for IoT Edge Networks Using Hybrid Machine Learning and Deep Reinforcement Learning Techniques. *Cyber Security and Network Management*, *1*(1), 01-11.

[37] Alavizadeh, H., Alavizadeh, H., & Jang-Jaccard, J. (2022). Deep Q-learning based

reinforcement learning approach for network intrusion detection. *Computers*, *11*(3), 41.

[38] Sharma, V. (2025). Rainbow dqn for intrusion detection: A unified deep reinforcement learning approach across benchmark datasets. *International Journal of Applied Mathematics*, *38*(5s), 647-675.

[39] Mahbub, M., Riasat, M. T., Hamid, T., Sutradhar, S. C., & Khan, M. S. A. (2026). A minimalistic yet effective domain adaptation strategy for IoMT network intrusion detection. *Discover Internet of Things.*

[40] Nadhir, A. M., Mounir, B., Abdelkader, L., & Hammoudeh, M. (2025). Enhancing cybersecurity in healthcare IoT systems using reinforcement learning. Transportation Research Procedia, 84, 113-120.

[41] Scarfone, K., & Mell, P. (2007). Guide to intrusion detection and prevention systems (idps). *NIST special publication*, *800*(2007), 94.

[42] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, *41*(3), 1-58.

[43] Brandao, P. R. (2025). Cyber Threat Intelligence with Symmetry for Zero-Trust Security.

[44] Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., ... & Wang, C. (2018). Machine learning and deep learning methods for cybersecurity. *Ieee access*, *6*, 35365-35381.

[45] Alrawais, A., Alhothaily, A., Hu, C., & Cheng, X. (2017). Fog computing for the internet of things: Security and privacy issues. *IEEE Internet Computing*, *21*(2), 34-42.

[46] Buczak, A. L., & Guven, E. (2015). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications surveys & tutorials*, *18*(2), 1153-1176.

[47] Sommer, R., & Paxson, V. (2010, May). Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE symposium on security and privacy* (pp. 305-316). IEEE.

[48] Ge, M., Fu, X., Syed, N., Baig, Z., Teo, G., & Robles-Kelly, A. (2019, December). Deep learning-based intrusion detection for IoT networks. In *2019 IEEE 24th*

pacific rim international symposium on dependable computing (PRDC) (pp. 256-25609). IEEE.

[49] Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access*, *5*, 21954-21961.

[50] Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction* (Vol. 1, No. 1, pp. 9-11). Cambridge: MIT press.

[51] Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

[52] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *nature*, *518*(7540), 529-533.

[53] Van Hasselt, H., Guez, A., & Silver, D. (2016, March). Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 30, No. 1).

[54] Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., & Freitas, N. (2016, June). Dueling network architectures for deep reinforcement learning. In *International conference on machine learning* (pp. 1995-2003). PMLR.

[55] Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2015). Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.

[56] Han, H., Wang, W. Y., & Mao, B. H. (2005, August). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing* (pp. 878-887). Berlin, Heidelberg: Springer Berlin Heidelberg.

[57] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.

[58] Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, *6*(1), 20-29.

B. Web References

[W20] IBM Security. (2025). Cost of a Data Breach Report 2025. IBM. Available online:: https://www.ibm.com/security/data-breach

[W21] American Hospital Association. (2023). Hospital Cyber Resiliency Initiative Landscape Analysis. AHA, HHS & HSCC. Available online:: https://www.aha.org/guidesreports/2023-04-18-hospital-cyber-resiliency-initiative-landscape-analysis

[W22] Mazhar T. et al. (2023). Analysis of IoT Security Challenges... Brain Sciences, 13(4), 683. Available online:: https://www.mdpi.com/2076-3425/13/4/683

[W23] BreachLock. (2024). IoT in Healthcare: The Expanding Threat Landscape... Available online:: https://www.breachlock.com/resources/blog/iot-in-healthcare-the-expanding-threat-landscape-and-strategies-to-mitigate-it/

[W dataset] https://www.unb.ca/cic/datasets/iomt-dataset-2024.html