

**T.R.**  
**GEBZE TECHNICAL UNIVERSITY**  
**GRADUATE SCHOOL**

**PERFORMANCE OF AI ALGORITHMS FOR RELIABLE UAV  
SWARM COMMUNICATIONS UNDER DIFFERENT JAMMING**

**ÖMER FARUK DİKMEN**

**A THESIS OF MASTER OF SCIENCE**  
**DEPARTMENT OF COMPUTER ENGINEERING**

**ADVISOR: PROF. DR. HASARİ ÇELEBİ**

**JUNE 2025**

**T.R.**  
**GEBZE TECHNICAL UNIVERSITY**  
**GRADUATE SCHOOL**

**PERFORMANCE OF AI ALGORITHMS FOR  
RELIABLE UAV SWARM COMMUNICATIONS UNDER  
DIFFERENT JAMMING**

**ÖMER FARUK DİKMEN**

**A THESIS OF MASTER OF SCIENCE**  
**DEPARTMENT OF COMPUTER ENGINEERING**

**ADVISOR: PROF. DR. HASARİ ÇELEBİ**

**JUNE 2025**

T.C.  
GEBZE TEKNİK ÜNİVERSİTESİ  
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

FARKLI SİNYAL KARIŞTIRMALARI ALTINDA  
GÜVENİLİR İHA SÜRÜ HABERLEŞMESİ İÇİN YAPAY  
ZEKA ALGORİTMALARININ PERFORMANSI

ÖMER FARUK DİKMEN

YÜKSEK LİSANS TEZİ  
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMAN: PROF. DR. HASARİ ÇELEBİ

HAZİRAN 2025

## MASTER of SCIENCE JURY APPROVAL FORM

A thesis submitted by Name, Surname, defended on 16/06/2025 before the jury formed with the 11/06/2025 date and 2025/31 numbered decision of the GTU Graduate Administration Board, has been accepted as a MASTER of SCIENCE thesis in the Department of Computer Engineering.

### JURY

MEMBER

(THESIS ADVISOR) :Prof. Dr. Hasari Çelebi

MEMBER

:Prof. Dr. Mehmet Kemal Özdemir

MEMBER

:Assist. Prof. Dr. Abuu Bakari Kihero

:

### APPROVAL

Gebze Technical University Graduate Administration Board  
...../...../..... date and ...../..... numbered decision.

SIGNATURE/SEAL



*To my beloved father, who  
is no longer with us and who  
always wanted me to pursue an  
academic path, and to my dear  
son, who, years later, inspired me  
to return to academia.*

## ABSTRACT

In this thesis, deep reinforcement learning based anti-jamming approaches have been developed and systematically evaluated in multidimensional simulation scenarios to ensure reliable and efficient communication of unmanned aerial vehicle (UAV) swarms in dynamic and threat-laden environments. In the study, a three-stage experimental structure was followed. In the first stage, the basic responses of the agents are analysed under the threat of an adaptive jammer, one of the six jammer types used in the experiments, in a simple environmental model where communication is over a single channel. In the second phase, environmental complexity was increased, expanding the action and observation space; separate jammer power per channel, TTL-based retransmission, multidimensional actions, and advanced log mechanisms were incorporated into the system. In the third and final experiment, six different algorithms—TD3, SAC, DDPG, SingleAgentTD3, SAR, and Random—were compared in a comprehensive benchmark environment against six different types of jammers: adaptive, comb, sweeping, dynamic, smart, and random.

In the study, advanced reinforcement learning architectures were configured, particularly for decisions such as channel selection, power control, relay strategy, and target determination, which require continuous action space and can be optimized simultaneously. According to the findings, the TD3 algorithm produced the most stable and high-performance results thanks to its double value structure. SingleAgentTD3 achieved significant success in decentralized scenarios, while SAC exhibited consistent behavior against certain types of jammers. DDPG, despite its rapid learning, has experienced issues with evaluation stability. The SAR agent stands out as an effective reference agent in low-complexity environments, while the Random agent has been used solely for comparison purposes.

Additionally, both centralized and decentralized architectures were evaluated within the scope of the study; the performance of agents in terms of metrics such as energy efficiency, fairness, success rate, and throughput against different jammer strategies was compared. When the results are analysed, it is seen that TD3 algorithm has the best performance in general with an average throughput of 4950, a low packet loss rate of 5% and balanced energy usage. The SAR algorithm, which does not have learning capability, attracted attention with a throughput value of 5000 and near-zero packet loss, especially in stationary environments. Although the SingleAgentTD3 algorithm showed the best transmission success with a drop rate of 3% in the smart jammer scenario, it exhibited unstable behaviour from time to time due to high critic loss values. DDPG and SAC algorithms produced unstable results especially against dynamic jammers. On the other hand, the random algorithm showed the lowest success with a throughput of about 1300 and a drop rate of over 60%, which clearly demonstrates the necessity of learning-based methods. In conclusion, the study provides an innovative foundation that contributes to the development of flexible and learning UAV systems with multidimensional decision-making capabilities, both in academic and practical applications.

**Keywords: Jamming, Anti-jamming, Reinforcement Learning, Swarm UAV Communication**

## ÖZET

Bu tez çalışmasında, insansız hava aracı (İHA) sürülerinin dinamik ve tehdit içeren ortamlarda güvenilir ve verimli haberleşmesini sağlamak amacıyla, derin pekiştirmeli öğrenme (DPL) tabanlı anti-jamming yaklaşımları geliştirilmiş ve çok boyutlu simülasyon senaryolarında sistematik olarak değerlendirilmiştir. Çalışmada üç aşamalı deneysel bir yapı izlenmiştir. İlk aşamada, basit bir çevre modeli altında tek kanal ve adaptif jammer tehdidi kullanılarak ajanların temel tepkileri incelenmiştir. İkinci aşamada, çevresel karmaşıklık artırılarak aksiyon ve gözlem uzayı genişletilmiş; kanal başına ayrı jammer gücü, TTL tabanlı yeniden iletim, çok boyutlu aksiyonlar ve gelişmiş log mekanizmaları sisteme dahil edilmiştir. Üçüncü ve son deneyde ise TD3, SAC, DDPG, SingleAgentTD3, SAR ve Random olmak üzere altı farklı algoritma; adaptive, comb, sweeping, dynamic, smart ve random olmak üzere altı farklı jammer tipine karşı kapsamlı bir benchmark ortamında karşılaştırılmıştır.

Çalışmada, özellikle sürekli aksiyon uzayı gerektiren kanal seçimi, güç kontrolü, relay stratejisi ve hedef belirleme gibi kararların aynı anda optimize edilebildiği gelişmiş pekiştirmeli öğrenme mimarileri yapılandırılmıştır. Elde edilen bulgulara göre, TD3 algoritması çift değerleme yapısı sayesinde en istikrarlı ve yüksek performanslı sonuçları üretmiştir. SingleAgentTD3, merkezi olmayan senaryolarda önemli başarılar elde ederken; SAC belirli jammer tiplerine karşı tutarlı davranış sergilemiştir. DDPG ise hızlı öğrenmesine rağmen değerlendirme istikrarında sorunlar yaşamıştır. SAR ajanı, düşük karmaşıklık ortamlarda etkili bir referans ajan olarak öne çıkarken; Random ajan yalnızca karşılaştırma amacıyla kullanılmıştır.

Ayrıca çalışma kapsamında hem merkezi hem merkezi olmayan mimariler değerlendirilmiş; farklı jammer stratejilerine karşı ajanların enerji verimliliği, adalet, başarı oranı ve throughput gibi metriklerdeki performansları karşılaştırılmıştır. Sonuçlar incelendiğinde, TD3 algoritmasının ortalama 4950 throughput, %5'lik düşük bir paket kaybı oranı ve dengeli enerji kullanımı ile genel olarak en başarılı performansı sergilediği görülmektedir. Öğrenme yeteneği bulunmayan SAR algoritması ise özellikle sabit ortamlarda 5000'e ulaşan throughput değeri ve sıfıra yakın paket kaybı ile dikkat çekmiştir. SingleAgentTD3 algoritması, smart jammer senaryosunda %3'lük drop rate ile en iyi iletim başarısını göstermiş olsa da, yüksek critic loss değerleri nedeniyle zaman zaman kararsız davranışlar sergilemiştir. DDPG ve SAC algoritmaları ise özellikle dinamik jammer türleri karşısında istikrarsız sonuçlar üretmiştir. Öte yandan, random algoritması yaklaşık 1300 throughput ve %60'ın üzerinde drop rate ile en düşük başarıyı göstermiş; bu da öğrenmeye dayalı yöntemlerin ne denli gerekli olduğunu açık bir şekilde ortaya koymuştur. Sonuç olarak, çalışma hem akademik hem de pratik uygulamalar açısından, çok boyutlu karar verme yeteneğine sahip, esnek ve öğrenen UAV sistemlerinin geliştirilmesine katkı sağlayan yenilikçi bir temel sunmaktadır.

**Anahtar Kelimeler:** Sinyal Karıştırma, Sinyal Karıştırma Önleme, Pekiştirmeli Öğrenme, Sürü Halinde İHA Haberleşmesi

## ACKNOWLEDGEMENTS

This study is supported by TÜBİTAK BİLGEM (The Scientific and Technological Research Council of Türkiye Informatics and Information Security Research Center).

I would also like to express my sincere gratitude to my advisor, Prof. Dr. Hasari Çelebi, for his valuable guidance, continuous support, and insightful feedback throughout the course of this research.



# TABLE OF CONTENTS

|                                                      | <b><u>Page</u></b> |
|------------------------------------------------------|--------------------|
| ABSTRACT                                             | vi                 |
| ÖZET                                                 | vii                |
| ACKNOWLEDGEMENTS                                     | viii               |
| TABLE OF CONTENTS                                    | ix                 |
| LIST OF SYMBOLS AND ABBREVIATIONS                    | xi                 |
| LIST OF FIGURES                                      | xii                |
| LIST OF TABLES                                       | xiv                |
| <br>                                                 |                    |
| 1. INTRODUCTION                                      | 1                  |
| 1.1. History                                         | 2                  |
| 1.2. Radio Systems                                   | 3                  |
| 1.3. Standards                                       | 4                  |
| 1.4. Cellular Systems                                | 5                  |
| 1.5. Satellite Systems                               | 6                  |
| 1.6. Electromagnetic Spectrum                        | 8                  |
| 1.7. Jammer                                          | 9                  |
| 1.8. UAV                                             | 10                 |
| 1.9. Anti-jamming Techniques                         | 12                 |
| 1.9.1. Spread Spectrum Techniques                    | 12                 |
| 1.9.2. MIMO Based Jamming Mitigation                 | 12                 |
| 1.9.3. Coding Methods                                | 13                 |
| 1.9.4. Power Control and Rate Adaptation             | 13                 |
| 1.9.5. Adaptability at the Protocol Level            | 13                 |
| 1.9.6. Network Layer Routing Strategies              | 13                 |
| 1.9.7. Jamming Detection and Machine Learning        | 13                 |
| 1.9.8. Adaptive Strategies for UAV Systems           | 14                 |
| 1.10. Approaches in Literature                       | 14                 |
| 1.11. Artificial Intelligence in Spectrum Management | 15                 |
| 1.12. DRL                                            | 19                 |
| 2. PURPOSE OF THE STUDY                              | 22                 |
| 2.1. Increasing Strategic Importance of UAV Systems  | 22                 |
| 2.2. Wireless Communication in UAV Systems           | 22                 |
| 2.3. Two-way Dependency                              | 23                 |
| 2.4. The Role of DL and RL                           | 23                 |
| 2.5. DRL and Multidimensional Decision Problems      | 24                 |
| 2.6. Contribution of the Study                       | 25                 |

|                                                 |     |
|-------------------------------------------------|-----|
| 3. EXPERIMENT                                   | 27  |
| 3.1. Problem Statement                          | 27  |
| 3.2. Environment                                | 29  |
| 3.2.1. Observation Space                        | 31  |
| 3.2.2. Action Space                             | 32  |
| 3.2.3. Channel and SINR Calculation             | 34  |
| 3.2.4. Energy Model                             | 35  |
| 3.2.5. Jammer Types                             | 35  |
| 3.3. Methodology                                | 36  |
| 3.3.1. Problem Statement and Objective Function | 36  |
| 3.3.2. Timing and Transmission Order            | 38  |
| 3.3.3. Target Selection and Transmission Model  | 38  |
| 3.3.4. Transmission Type                        | 38  |
| 3.3.5. Learning Parameters and Algorithm        | 38  |
| 3.3.6. DDPG                                     | 39  |
| 3.3.7. TD3                                      | 42  |
| 3.3.8. SAC                                      | 45  |
| 3.3.9. SingleAgentTD3                           | 48  |
| 3.3.10. SAR                                     | 51  |
| 3.3.11. Random                                  | 53  |
| 3.4. Jamming System                             | 54  |
| 3.5. Experiment 1                               | 56  |
| 3.6. Experiment 2                               | 62  |
| 3.7. Experiment 3                               | 64  |
| 4. RESULTS                                      | 66  |
| 4.1. Experiment 1                               | 66  |
| 4.2. Experiment 2                               | 71  |
| 4.3. Experiment 3                               | 75  |
| 5. CONCLUSION                                   | 99  |
| REFERENCES                                      | 102 |
| BIOGRAPHY                                       | 110 |
| PUBLICATIONS AND PRESENTATIONS FROM THE THESIS  | 111 |

## LIST OF SYMBOLS AND ABBREVIATIONS

|           |                                             |
|-----------|---------------------------------------------|
| GHz       | : Gigahertz                                 |
| MHz       | : Megahertz                                 |
| sec       | : second                                    |
| ALOHA NET | : Aloha Network                             |
| DARPA     | : Defense Advanced Research Projects Agency |
| US        | : United States                             |
| USA       | : United States of America                  |
| FCC       | : Federal Communications Commission         |
| ISM       | : Industrial, Scientific and Medical(Bands) |
| Wi-Fi     | : Wireless Fidelity                         |
| AT&T      | : American Telephone and Telegraph          |
| AM        | : Amplitude Modulation                      |
| FM        | : Frequency Modulation                      |
| AMPS      | : Advanced Mobile Phone System              |
| LTE       | : Long Term Evolution                       |
| GEO       | : Geostationary Orbit                       |
| LEO       | : Low Earth Orbit                           |
| MAC       | : Media Access Control                      |
| MANET     | : Mobile Ad hoc Network                     |
| UAV       | : Unmanned Aerial Vehicle                   |
| DDPG      | : Deep Deterministic Policy Gradient        |
| TD3       | : Twin Delayed Three                        |
| SAC       | : Soft Actor Critic                         |
| SAR       | : Simple Action Rule Based                  |
| BER       | : Bit Error Rate                            |
| SINR      | : Signal to Interference Plus Noise Ratio   |
| OU        | : Ornstein Uhlenbeck                        |
| TD        | : Temporal Difference                       |
| TTL       | : Time To Live                              |
| ReLU      | : Rectified Linear Unit                     |
| N/A       | : Not Applicable                            |

## LIST OF FIGURES

|                                                                             | <u>Page</u> |
|-----------------------------------------------------------------------------|-------------|
| <b>Figure 3.1:</b> UAV Decision Parameters.                                 | 28          |
| <b>Figure 3.2:</b> Environment.                                             | 30          |
| <b>Figure 3.3:</b> Rewarding Mechanism.                                     | 37          |
| <b>Figure 3.4:</b> EDDPG Algorithm Architecture Used in the Experiments.    | 41          |
| <b>Figure 3.5:</b> ETD3 Learning Architecture.                              | 42          |
| <b>Figure 3.6:</b> ESAC Architecture in the Simulation.                     | 46          |
| <b>Figure 3.7:</b> Single TD3 Architecture.                                 | 50          |
| <b>Figure 3.8:</b> Algorithm Flowchart.                                     | 52          |
| <b>Figure 3.9:</b> Random Action Generation Flow.                           | 54          |
| <b>Figure 3.10:</b> Jamming Techniques Used in Experiments.                 | 55          |
| <b>Figure 3.11:</b> Deep Neural Network Used in Experiment 1.               | 59          |
| <b>Figure 3.12:</b> Deep Neural Network Structure for Experiment 2.         | 64          |
| <b>Figure 4.1:</b> Loss Over Episodes.                                      | 67          |
| <b>Figure 4.2:</b> Reward Over Episodes.                                    | 68          |
| <b>Figure 4.3:</b> Throughput Over Episodes.                                | 69          |
| <b>Figure 4.4:</b> Average Energy Consumption.                              | 69          |
| <b>Figure 4.5:</b> SINR Heatmap Distribution of SAR at Some Time Steps.     | 70          |
| <b>Figure 4.6:</b> Algorithm Radar Comparison.                              | 71          |
| <b>Figure 4.7:</b> Losses Over Episodes.                                    | 72          |
| <b>Figure 4.8:</b> Throughput Over Episodes.                                | 73          |
| <b>Figure 4.9:</b> Radar Chart (Normalised).                                | 73          |
| <b>Figure 4.10:</b> Radar Chart (Actual Values).                            | 74          |
| <b>Figure 4.11:</b> Distribution of Transmit Power for SAR (Watt).          | 75          |
| <b>Figure 4.12:</b> Throughput Per Episodes Under Adaptive Jamming Type.    | 76          |
| <b>Figure 4.13:</b> Actor Loss Under Adaptive Jamming Type.                 | 77          |
| <b>Figure 4.14:</b> Drop Rate Under Adaptive Jamming.                       | 77          |
| <b>Figure 4.15:</b> Throughput Under Comb Jamming.                          | 79          |
| <b>Figure 4.16:</b> Critic Loss Under Comb Jamming.                         | 79          |
| <b>Figure 4.17:</b> Drop Rate Under Comb Jamming.                           | 80          |
| <b>Figure 4.18:</b> Actor Loss Under Comb Jamming.                          | 80          |
| <b>Figure 4.19:</b> Throughput Per Episodes Under Dynamic Jamming.          | 81          |
| <b>Figure 4.20:</b> Drop Rate Per Episodes Under Dynamic Jamming.           | 82          |
| <b>Figure 4.21:</b> Throughput Per Episodes Under Smart Jamming.            | 83          |
| <b>Figure 4.22:</b> Drop Rate Per Episodes Under Smart Jamming.             | 83          |
| <b>Figure 4.23:</b> Throughput Per Episodes Under Sweeping Jamming.         | 84          |
| <b>Figure 4.24:</b> Critic Loss Per Episodes Under Sweeping Jamming.        | 84          |
| <b>Figure 4.25:</b> Throughput Per Episodes Under Random Jamming.           | 86          |
| <b>Figure 4.26:</b> Actor Loss Performance of DDPG Under Different Jamming. | 87          |
| <b>Figure 4.27:</b> Critic1 Loss Performance of DDPG.                       | 87          |
| <b>Figure 4.28:</b> Critic2 Loss Performance of DDPG.                       | 88          |
| <b>Figure 4.29:</b> Drop Rate of DDPG Under Different Jamming Types.        | 88          |

|                                                                            | <b><u>Page</u></b> |
|----------------------------------------------------------------------------|--------------------|
| <b>Figure 4.30:</b> Throughput of DDPG Under Different Jamming Types.      | 89                 |
| <b>Figure 4.31:</b> Actor Loss of TD3 Under Different Jamming Types.       | 90                 |
| <b>Figure 4.32:</b> Critic1 Loss of TD3 Under Different Jamming Types.     | 90                 |
| <b>Figure 4.33:</b> Critic2 Loss of TD3 Under Different Jamming Types.     | 91                 |
| <b>Figure 4.34:</b> Drop Rate of TD3 Under Different Jamming Types.        | 91                 |
| <b>Figure 4.35:</b> Throughput of TD3 Under Different Jamming Types.       | 92                 |
| <b>Figure 4.36:</b> Actor Loss of SAC Under Different Jamming Types.       | 92                 |
| <b>Figure 4.37:</b> Critic1 Loss of SAC Under Different Jamming Types.     | 93                 |
| <b>Figure 4.38:</b> Drop Rate of SAC Under Different Jamming Types.        | 93                 |
| <b>Figure 4.39:</b> Throughput of SAC Under Different Jamming Types.       | 94                 |
| <b>Figure 4.40:</b> Throughput of SingleAgentTD3 Under Different Jamming.  | 95                 |
| <b>Figure 4.41:</b> Drop Rate of SAR Under Different Jamming.              | 96                 |
| <b>Figure 4.42:</b> Drop Rate of Random Algorithm Under Different Jamming. | 97                 |



## LIST OF TABLES

|                                                                  | <b><u>Page</u></b> |
|------------------------------------------------------------------|--------------------|
| <b>Table 3.1:</b> The Learning Parameters                        | 39                 |
| <b>Table 4.1:</b> Performance Comparison Under Adaptive Jamming. | 76                 |
| <b>Table 4.2:</b> Performance Comparison Under Comb Jamming.     | 78                 |
| <b>Table 4.3:</b> Success Ranking of Algorithms                  | 86                 |



# 1. INTRODUCTION

Today, wireless network systems have gained an indispensable place in both civilian and military applications. A wide range of applications ranging from smart cities to healthcare services, from industrial automation to transport infrastructures clearly demonstrates the critical role of wireless communication technologies. In the military field, the importance of reliable, flexible and mobile communication capabilities is increasing; especially the coordination of units in the field of duty, information sharing and command and control functions rely heavily on wireless networks. In this context, Unmanned Aerial Vehicles (UAVs), which have become widespread in both civilian and military areas in recent years, have made communication networks even more powerful in terms of mobility, flexibility and coverage. The mission success of UAVs largely depends on the reliability and continuity of these wireless networks.

The basic building block of wireless communication is the electromagnetic spectrum, a limited and precious resource. The efficient use of this spectrum has become a major engineering challenge in terms of managing the increasingly crowded wireless traffic. Especially the dynamic and shared nature of the spectrum brings with it various sources of interference that threaten the quality of communication. This interference can sometimes be caused by natural congestion, i.e. traffic-related interference caused by excessive use of the same frequency resources, or it can be caused by malicious external attacks, such as jammers.

Jammers, especially in the military field or in situations where critical infrastructures are targeted, can disrupt communication and cause systems to be disabled. However, it is very difficult to provide effective communication in such a complex and ever-changing spectrum environment with classical, static methods. Therefore, environment-sensitive, flexible and agile approaches have started to be developed.

Considering the history and technical evolution of wireless communication, it is clear that more intelligent, adaptive and learning systems are required to address these challenges. In this context, artificial intelligence-based methods, especially deep reinforcement learning (DRL), offer promising solutions for dynamic spectrum management and dealing with threats such as jammers.

This introductory part touches upon the concepts that form the basis of our study, such as wireless network system and its history, electromagnetic spectrum, jammer, UAV, anti-jamming systems, approaches in the literature, the importance of artificial intelligence in the spectrum and DRL.

## **1.1. History**

Wireless communication is actually as old as human history. The first wireless networks were established in the pre-industrial period with visual methods such as smoke signals, flags, torches. These systems were eventually replaced by wired systems with the invention of the telegraph and then the telephone. In 1895, wireless radio communication started with Marconi's first successful communication attempt with radio signals. In 1971, ALOHANET, developed at the University of Hawaii, became the first packet radio network to provide radio communication between computers. This system attracted the attention of the US military and continued to be developed by DARPA for military communications in the 70s and 80s.

In 1985, the FCC opened the ISM frequency band to the public without a licence. This paved the way for the commercial development of wireless local area networks (wireless LANs) in the following years. However, in the 1990s, wireless data services were not yet widespread due to reasons such as low speed and high cost. Nevertheless, technology has advanced over time and Wi-Fi systems have improved in terms of data speed and coverage area and are now widely used. Today, wireless networks have become an important communication infrastructure preferred especially for accessing the Internet.

Wireless telephone systems are one of the most successful applications of wireless communication. The first mobile telephone service was launched in 1946. However, the number of users was very small due to capacity limitations. This problem was tried to be solved with the cellular system idea developed by AT&T in the 1950s. The first field test was carried out in 1978 and service started in 1982 [1].

As of 2025, the future of wireless communication is no longer just a vision, but has started to become a reality in many areas. The continuous connection between people and devices has become faster, low latency and reliable thanks to 5G and emerging 6G technologies. In smart homes, devices work integrated not only with each other but

also with cloud systems and artificial intelligence-based assistants; services such as remote health monitoring, emergency notifications, voice command control are offered for the elderly and disabled individuals.

Wireless sensor networks are now widely used not only for environmental monitoring, but also for traffic management, air quality measurement and energy efficiency in smart cities. In the military field, wireless systems play a critical role in the detection of enemy targets, data sharing and decision support systems with UAVs, autonomous ground vehicles, radar networks and tactical communication systems. In addition, thanks to wireless technologies, driverless vehicle systems, drone-based delivery services, mobile robots and industrial automation can now work with real-time data, providing revolutionary innovations in many areas from production to logistics.

## **1.2. Radio Systems**

Before transistors were invented, i.e. in ancient times, wireless systems (e.g. AM/FM radios, analogue televisions and amateur radio systems) operated entirely on analogue signals. However, most radio systems we use today now operate with digital signals. These digital signals are transmitted through data streams consisting of 1s and 0s in computers. Sometimes this data can be a file, a photo or a video. Or it may be created by converting an analogue signal, such as sound, into digital form (for example, by sampling the sound at regular intervals and converting it into a digital value). Digital radios can send these bit sequences continuously, or they can send them in packets. Such packet radios are called packet radios. Packet radios are only active when sending packets; otherwise they are silent. This is why this type of transmission is called 'bursty transmission'. However, if something (e.g. audio speech or video) is sent continuously with these packets, there must not be too long delays between incoming packets - otherwise the picture or sound will be distorted. In 1971, the first packet radio-based wireless network, called ALOHAnet, was developed at the University of Hawaii. With this network, seven different universities spread over four islands could connect wirelessly to a central computer. The system used a star topology revolving around a centre. In other words, all communication was through the central computer. ALOHAnet also contained the first protocols for channel sharing and routing [2]. The basic principles of these protocols are still in use today. Seeing this technology, the US military thought that this system could be used on the battlefield. So in the 1970s and

early 1980s, the military's DARPA organisation invested heavily in developing packet radio systems for combat communications. In these systems, radio equipment could form self-organising networks without the need for any infrastructure. Today, this is called 'ad hoc wireless networking'. However, by the mid-1980s, these systems failed to fulfil expectations. Their performance was poor, both because of the technical limitations of the radios and the inadequacy of the protocols used. Nevertheless, some of these systems were used commercially in the 1990s for wide-area wireless data services. These services made wireless internet (e.g. e-mail, file sending, web browsing) possible, but very slow - only about 20 kilobits per second. These services were very expensive and there were no 'killer apps' to attract people. So they did not catch on. What put an end to these services was wireless data transmission, which came with 2G mobile phone systems[3]. This was the breaking point that started the wireless data revolution.

### **1.3. Standards**

The ubiquity of wireless communications is largely due to the development of WLANs and mobile phone networks. WLANs are standardised with IEEE 802.11 (Wi-Fi) protocols. Satellite systems also play an important role in the wireless communication ecosystem. The success of Wi-Fi systems actually started as an extension of the Ethernet (802.3) standard developed for wired local area networks. Ethernet was developed at Xerox PARC in the 1970s and standardised in 1983. In 1985, the FCC in the USA released the 900 MHz, 2.4 GHz and 5.8 GHz frequency bands for unlicensed use. These bands had previously only been used for industrial, scientific and medical purposes. This decision has accelerated research and commercial product development for WLAN systems. NCR introduced "WaveLAN" in 1988, the first WLAN product that used ISM bands. However, these items were expensive and only provided rates of 2 Mbps. In the 1990s, numerous different WLAN products were developed, but due to a lack of standards, these systems were incompatible. At the time, Ethernet offered rates of up to 10 Mbps, therefore most businesses preferred wired systems. In 1997, the first official 802.11 standard was released. This standard operated in the 2.4 GHz range, with a maximum speed of 2 Mbps. The 802.11b standard, released in 1999, started the Wi-Fi revolution. Offering 11 Mbps speed, this standard enabled the emergence of lower cost and more compatible products than previous versions. In the

same year, with the 802.11a standard developed for the 5 GHz band, speeds of 54 Mbps were reached and Orthogonal Frequency Division Multiplexing (OFDM) technology was introduced. The 802.11g standard, released in 2003, had a similar physical structure to 802.11a but operated in the 2.4 GHz band. Subsequent standards developed wider channels, multiple antenna systems (MIMO), beamforming, larger modulation schemes and better error correction systems [4]. Today, Wi-Fi can offer data rates of up to 10 Gbps in indoor or outdoor areas. In addition, with new standards such as 802.11ad operating in the 60 GHz band, Wi-Fi has been moved to different frequency areas. In 2018, the naming method of the 802.11 standard was changed. For example, 802.11ax is now called "Wi-Fi 6". Today, Wi-Fi technology connects not only computers and phones, but also many electronic devices from medical devices to refrigerators and even cars to the Internet.

## **1.4. Cellular Systems**

Cellular systems are also a highly successful wireless communication technology. The combination of radio and telephony first began in 1915 when wireless voice transmission was provided between New York and San Francisco. In 1946, the first analogue mobile telephone system, Mobile Telephone Service (MTS), was launched in St. Louis, Missouri. However, this system allowed only a very small number of people to make calls at the same time, requiring high cost and large equipment. In fact, the idea that would solve the capacity problem was the concept of cellular system introduced by D. H. Ring from Bell Labs in 1947 [5]. Based on the principle that radio signals weaken with distance, this idea envisaged the reuse of the same frequencies in remote areas. Thus, the geographical area was divided into cells, each cell was assigned certain frequencies, and as the user travelled between cells, the call was transferred from one cell to another. This idea of "frequency reuse" is still the basis of cellular systems today. This idea could only be realised in practice as technology advanced in the 1960s. In 1971 AT&T submitted a proposal to the FCC requesting spectrum allocation for cellular service [6]. In 1981, the FCC issued a regulation allowing two operators to be licensed in each city. In 1983, the first generation (1G) cellular system, called AMPS, was launched in Chicago. Similar but incompatible systems were developed in Japan and some European countries [7]. This made roaming between countries impossible. Just like Wi-Fi, cellular systems grew much

faster than expected. The number of subscribers has increased about 10 times every ten years: 100,000 subscribers in 1984, millions in 1994, hundreds of millions in 2004 and billions in 2014. A new cellular generation (e.g. 2G, 3G, 4G, 5G) has emerged about every ten years to meet the growing demand for data and the diversity of devices.

1G as the initial generation of mobile phone systems is analogue. It debuted in Japan in 1979. It was later introduced into Europe and the United States. 2G systems, which were introduced in the 1990s, are digital. Initially, just voice services were available, but data services such as short messaging (SMS) and e-mail were eventually added. However, because so many distinct standards evolved, there was a challenge with harmonisation across countries. 3G offers broadband data services such as mobile internet, video calls, and mobile television. 3G systems can provide rates of at least 200 kbps per second. There are other faster versions available, including 3.5G and 3.75G. 4G with LTE and LTE Advanced technologies, data transfer of hundreds of megabits per second has become possible. LTE Advanced is an improved version of LTE and offers more bandwidth and system capacity. In this way, faster internet connection with mobile devices has become possible. 5G can be thought of as a much faster and lag-free version of the internet we currently use. While it used to take minutes to download a film, with 5G it only takes seconds. In addition, even as the number of devices connected to the internet increases, the connection quality does not decrease. Therefore, things like smart cities, driverless vehicles, live video calls are now much more stable. 6G has not yet fully entered our lives, but is under development. It will be even faster and smarter than 5G [8]. So much so that communication between devices will be almost 'instant'. Things like artificial intelligence, holograms, remote surgeries, virtual reality will become more possible with 6G.

## **1.5. Satellite Systems**

It enabled the first human landing on the Moon in 1969 to be broadcast live to 600 million viewers. In addition to television broadcasting, these early systems provided voice communications, telex, and paging services. GEO satellite technology has advanced tremendously over time. While early systems utilised the C band (4-8 GHz) of the radio spectrum, later systems moved to the X (8-12.5 GHz), Ku (12.5-18 GHz) and Ka (26.5-40 GHz) bands, offering higher data rates. Antenna technologies have

also improved, enabling the signal to be focused in a narrower area, thus enabling higher signal quality and higher data rates. Hundreds of GEO satellites are in operation today, with some capable of data rates totalling in excess of 100 Gbps. Typical data rates for individual users are several tens of Mbps in the downlink direction and several Mbps in the uplink direction. Due to their wide coverage, geosynchronous satellites have become the primary means of communication for maritime and airborne vehicles. They are also well suited for broadcast-based entertainment applications where two-way delay is not an issue and can transmit hundreds of digital television and radio channels in high quality [9]. With the proliferation of cellular systems in the late 1980s, the focus of satellite technology shifted to the development of LEO systems. Accordingly, some LEO communication systems such as Globalstar and Iridium became operational in the late 1990s. Although these systems provided global coverage, data transmission rates remained low due to power and bandwidth limitations. In addition, the handheld terminals developed for these systems were large and cumbersome compared to mobile phones, especially due to their large batteries. LEO systems were not commercially successful because they offered lower performance and higher cost compared to 2G cellular technologies. For this reason, pioneering companies such as Globalstar and Iridium filed for bankruptcy protection several years after launches. More ambitious ventures such as Teledesic (which planned more than 800 satellites) went bankrupt before they had even launched a satellite. Iridium and Globalstar, which have been restructured over time to offer smaller scale but profitable services, still provide global internet connectivity today - but with subscriber numbers significantly lower than cellular services. With the deployment of 4G systems, the demand for connectivity in remote areas not reached by cellular services, together with better satellite and earth receiver technologies, has revitalised interest in LEO satellite systems. Especially with the introduction of 5G systems, new LEO projects involving hundreds or even thousands of satellites are being developed to provide broadband internet for billions of people around the world who do not have access to high-speed connectivity [10]. Today, it is almost like magic to be able to communicate wirelessly with our devices wherever we are in the world. Behind this magic lies the enormous contribution of pioneering researchers and engineers who laid the foundation for the powerful systems we see everywhere today.

## 1.6. Electromagnetic Spectrum

Wireless communication systems are communication infrastructures that use electromagnetic waves instead of physical cables to transmit data. These systems operate in specific frequency ranges and these frequency ranges are shared over a limited natural resource, the electromagnetic spectrum (EMS). Initially, the spectrum was allocated and regulated according to service types such as radio, television, defence applications and emergency communications [11]. However, over time, the rapid increase in the number of wireless devices such as mobile phones, laptops, tablets, wireless security cameras, Internet of Things (IoT) devices and autonomous vehicles has made the efficient use of frequency spectrum an important problem [12]. This increase has made the inadequacy and unbalanced utilisation of available spectrum resources more visible. For example, while some frequency bands are intensively utilised, others remain mostly idle. This situation leads to the problem of spectrum inefficiency.

Specific frequency bands are used for different types of communication. Radio broadcasts generally operate in the 88-108 MHz (VHF band), television broadcasts in the 470-862 MHz (UHF band), and mobile communication systems in the 700 MHz to 6 GHz range (e.g. LTE Band 3: 1800 MHz, Band 7: 2600 MHz). Wi-Fi systems operate in the 2.4 GHz and 5 GHz ISM bands, while radar and defence applications use various bands in the 1-40 GHz range. In order to utilise spectrum resources more efficiently, a dynamic spectrum access (DSA) approach has been developed. Cognitive radio (CR), an important application of this approach, is an intelligent communication system that can detect the spectrum utilisation status of its environment, identify free frequency bands and make these bands temporarily available for secondary users [13]. For example, gaps in the 700-800 MHz band that are not actively used by primary users can be utilised by secondary users with short-term communication needs. However, this flexibility creates the risk that primary and secondary users may use the same frequency simultaneously. This may cause interference problems and adversely affect the quality of communication [14].

Therefore, spectrum sensing techniques with high accuracy and effective spectrum management algorithms are being intensively studied in order to prevent collisions and secure spectrum sharing. In this context, especially millimetre wave (mmWave) bands

covering the 24-100 GHz range have become an important research topic for 5G and future 6G systems [15]. As a result, the increasing number of devices and data traffic necessitates the management of the electromagnetic spectrum with more flexible, smarter and safer methods. CR systems have the potential to meet this need and play a critical role in the future of wireless communications.

## **1.7. Jammer**

The main characteristic of wireless communication systems is the transmission of data via free signal propagation over the EMS. Although this feature provides great advantages in terms of installation and mobility of systems, it also creates a serious security gap. Because the signals are not protected by a physical boundary, they become open to external interference. The most common and effective of these interventions are jamming attacks known as ‘jamming’. A jammer is a device that deliberately emits electromagnetic interference (noise) into the environment in order to disrupt, slow down or completely disable a wireless communication system. While such devices are especially used in strategic contexts such as military operations, intelligence activities and cyber attacks, attacks on civilian wireless networks are also on the rise today. The EMS, which is the cornerstone of wireless communication, covers radio waves divided into certain frequency ranges. Wireless devices operate in certain bands of this spectrum (for example, the unlicensed 2.4 GHz and 5 GHz ISM bands). Jammers can disrupt communication by directly interfering with these bands used by the target system. Since the attack takes place directly at the physical layer, communication protocols at the upper layer often cannot detect and prevent this situation. This increases the impact of the attack and makes it difficult to detect.

With the increasing importance of wireless communication systems, the threats to these systems have become both diversified and more sophisticated. Jammers are one of these threats and are classified according to different technical characteristics. Each jammer type creates unique effects on the target system by adopting a different attack strategy. The most basic of these jammer types, the constant jammer, emits continuous and high-powered interference on the channel, making the environment appear to be constantly busy. As a result, the target system cannot transmit any data. However, this approach is not sustainable in the long term due to high energy consumption.

A more intelligent form of attack, the reactive jammer, is activated only when it detects an active communication on the channel. This saves energy and makes the attack more difficult to detect. In this respect, it is especially preferred in covert operations.

The deceptive jammer, on the other hand, gives the appearance of real communication by providing fake signals to the systems. The receiving party interprets these deceptive signals as true and makes incorrect decisions. This significantly reduces the system's reliability.

Periodic or random jammers emit interference at regular intervals or at completely random times. This irregularity puts the adaptation mechanisms of the system in a difficult situation. Detection is much more difficult compared to classical methods.

The frequency sweeping jammer type, on the other hand, creates a broadband jamming effect by travelling many frequencies in a short time. It can be especially effective even in frequency hopping systems.

Finally, timing, synchronisation or MAC layer-oriented jammers target the control signals (such as Acknowledgement (ACK), Ready To Send (RTS)/ Clear To Send (CTS) of wireless communication protocols, disrupting data transmission before it starts. Such attacks disrupt not only the communication capacity of the system, but also its basic operating mechanism [16].

All these jammer types threaten a wide range of applications such as UAV swarms, wireless sensor networks, mobile communication infrastructures, emergency communication systems and military communications. Especially in time-sensitive missions, interruptions can cause not only data loss, but also mission failures and serious security vulnerabilities. Therefore, jammer-induced threats have become an increasing security risk for both civilian and military wireless communication systems.

## **1.8. UAV**

Today, UAVs are used not only in military operations but also in civil and public services. These vehicles can provide temporary communication services in times of disaster, such as the collapse of the communication infrastructure, and play important roles in many different areas, from search and rescue activities to agricultural and environmental monitoring, from traffic monitoring to forest fire detection. This flexible

and wide-ranging use of UAVs is made possible by their direct wireless communication capabilities. These vehicles, which fly at low altitudes and have a mobile structure, serve as a temporary communication infrastructure for users who cannot reach established base stations. However, for such systems to work effectively, dynamic, self-organising and energy efficient communication protocols are required.

UAV wireless communication systems differ significantly from conventional ground-based communication networks. For example, although channel attenuation is low because UAVs usually have line of sight (LoS) communication, link continuity cannot always be guaranteed due to weather conditions, mobility and energy constraints. In addition, as UAVs move in three-dimensional space, the network topology is constantly changing, making existing MANET protocols inadequate. Multi-UAV systems offer wider coverage, higher service continuity and stronger fault tolerance compared to single systems. In these systems, UAVs communicate with each other within the network and share data transmission and routing tasks. In the event of a failure of one UAV in the network, other UAVs can dynamically reorganise the structure and maintain service. This structure requires self-healing and self-organising capabilities of the network.

However, the fact that UAVs have limited energy resources is a significant challenge in system design. Typically, small UAVs have a flight time of only 30 minutes. Therefore, it is of great importance to keep energy consumption to a minimum during communication and to provide a balanced distribution between flight planning and communication tasks. Recently, new generation multiple access techniques such as Non-Orthogonal Multiple Access (NOMA) have started to be integrated into UAV communication to further improve performance in this area. NOMA increases spectral efficiency by using different power levels while sharing the same frequency resource among multiple users. When this structure is combined with UAV systems, it becomes possible to provide simultaneous service to more users. However, the challenges of this combination include user mapping, power allocation and the design of algorithms suitable for mobile infrastructure. In summary, UAV wireless communication systems have the potential to provide solutions in many scenarios ranging from emergency communication to infrastructure needs in rural areas with their flexible, dynamic and robust structures. However, in order for these systems to be used effectively in real life, unique solutions need to be developed in areas such as routing, energy

management, resilience to interruptions and security. This study aims to propose solutions to these challenges in order to provide a more solid foundation for the future role of UAV communications [17], [18], [19], [20], [21].

## **1.9. Anti-jamming Techniques**

Jammer-based attacks target the open spectrum structure, which is one of the most fundamental weaknesses of wireless communication. Defence methods developed to reduce or completely eliminate the effect of such attacks are called ‘anti-jamming’ techniques. The main purpose of these techniques is to ensure the continuity of communication, protect network performance and increase the reliability of the systems. The solutions offered often include techniques for the physical, MAC, and network layers [22], [23].

### **1.9.1. Spread Spectrum Techniques**

In wireless communication systems, extending the signal over a wider range without being limited to a certain frequency helps reduce the jammer effect. In this sense, two primary strategies emerge:

- Frequency Hopping Spread Spectrum (FHSS): The signal is transmitted by bouncing between frequencies in brief bursts. The jammer has difficulty tracking the signal.
- Direct Sequence Spread Spectrum (DSSS): The data is distributed over a wide spectrum, reducing the impact of narrowband interference. Karishma et al. [24] investigated the resistance of DSSS against a fixed jammer.

### **1.9.2. MIMO Based Jamming Mitigation**

MIMO systems offer the possibility of spatially separating jammer signals and target signals. In these systems, the jammer signal can be blocked by projection into perpendicular space. Yan et al. [25] showed that projection-based zero forcing techniques provide high success against reactive jammers.

### **1.9.3. Coding Methods**

Advanced channel coding methods such as Low Density Parity Check (LDPC) and Reed-Solomon (RS) facilitate data recovery in low-duty cycle jammers. These strategies ensure that communication continues even if attacked by replicating sent data [26], [27].

### **1.9.4. Power Control and Rate Adaptation**

Dynamically altering modulation rates based on link quality can help to decrease the jammer impact. Lower modulation rates provide more reliable transmission in weak signal conditions. Pelechrinis et al.[28] showed that this strategy is effective in low-power attacks.

### **1.9.5. Adaptability at the Protocol Level**

It is possible to maintain communication with alternative time, frequency or path strategies after jammer detection at the MAC layer. Such adaptive protocols can provide uninterrupted data transmission even when attacks are intense [29].

### **1.9.6. Network Layer Routing Strategies**

Approaches such as disruption-tolerant routing and multi-path routing, which are designed with the principle of avoiding jammer zones in routing algorithms, protect the overall performance of the network by distributing the effect of attacks on a single point [30].

### **1.9.7. Jamming Detection and Machine Learning**

It is possible to estimate the presence of jammers with statistical data collected from the environment. Puñal et al.[32] provided jammer detection with up to 98% accuracy with the random forest algorithm. Davaslioglu et al.[31] showed near-zero data loss against different jammer types with the Convolutional Neural Networks (CNN) based model named DeepWiFi. These approaches give effective results especially in dynamic and complex environments.

### **1.9.8. Adaptive Strategies for UAV Systems**

In swarming UAV systems, anti-jamming solutions are often combined with strategies based on distributed decision making, channel selection, dynamic routing and reinforcement learning. UAVs must be able to both detect the jammer and quickly select the appropriate communication channel and route [33]. In this context, the DRL algorithm used in this study is in line with the current literature and makes an innovative contribution.

### **1.10. Approaches in Literature**

With the proliferation of wireless communication systems, the seriousness of jammer-based attacks targeting these systems has gradually increased. This has led to an acceleration of research on anti-jamming solutions both in academic circles and in industrial applications. Recent studies show that the techniques developed against jamming attacks have become increasingly sophisticated and multi-layered. One of the early studies, Xu et al. [34], is an important research that lays the foundation for jammer detection and countermeasure mechanisms. In this study, jammer behaviours were classified at the physical and MAC layers and defence strategies were proposed according to this classification. In particular, ‘channel surfing’ and ‘spatial retreat’ have been shown to provide effective protection with low resource consumption.

Pelechrinis and colleagues [28] analysed the effects of jammer types on wireless network performance in detail. The defence algorithms developed in line with this analysis focus on energy efficiency. These algorithms are designed to maintain network continuity even under low-power attacks. This approach is of great importance for energy-constrained systems such as battery-powered mobile devices and sensor networks. Zhou et al. [35], working on flying networks (UAV networks), went beyond classical defence approaches and proposed an ‘anti-jamming dynamic routing’ algorithm that automatically moves nodes away from jammer-affected areas. This routing technique aims to maintain uninterrupted communication by reconfiguring the topology of the network. This study has demonstrated how critical spatial adaptation is in systems with mobile nodes.

Hanawal et al.[36] investigated strategies for achieving a more balanced and sustainable approach to frequency selection, similar to a chess match in which both

parties make strategic decisions, known as game theory. This study has laid a solid foundation, especially in cases when dynamic spectrum management approaches like frequency hopping are used.

Over time, the need for not only strategic modelling but also systems that directly observe and learn from the environment has increased. At this point, artificial intelligence (AI), particularly machine learning and deep learning approaches, come into play. For example, Liu and colleagues [37] employed deep Q-learning (DQL) to develop a system that enables UAVs to choose a safe route and communication channel while being immune to jammers. In this strategy, UAVs were able to detect risks in their environment and design their own communication strategies without requiring external aid.

## **1.11. Artificial Intelligence in Spectrum Management**

With the development of wireless communication systems, the efficient management of EMS has become one of the most important engineering problems of today. The fact that spectrum is a limited natural resource and the rapid increase in the number of devices has caused traditional fixed allocation management models to be inadequate. At this point, the concept of dynamic spectrum management comes to the forefront and AI techniques are utilised to carry out this process in a more intelligent and autonomous manner. In particular, deep learning models have shown striking success in various tasks such as spectrum sensing, classification, signal recognition and spectrum estimation in recent years. CNN, Recurrent Neural Networks (RNN), and Transformer-based models have been shown to produce high-accuracy decisions by learning complicated patterns on the spectrum [38]. For example, a Long Short-Term Model (LSTM) model applied for spectrum prediction was 30% more accurate than traditional methods [39].

However, the success of these models is usually directly related to their generalisation ability. In other words, the model should be able to be successful not only in the data it has been trained on, but also in new situations that it has not encountered before. In order to develop this generalisation capability, very large volumes of labelled data are needed. However, it is extremely difficult to collect such large and diverse data sets in a rapidly changing environment such as the EMS, whose input-output behaviour can

change within seconds and often belong to unknown devices. At this point, reinforcement learning (RL) models, which can learn from the environment through direct experience and make real-time decisions, offer an important alternative to data-driven learning. RL uses the interaction between an agent (e.g. a CR) and the environment (spectrum) to learn which action will bring the highest reward. This method offers the ability to develop strategies by interacting with the environment without the need for labelled data [40].

However, in the presence of complex, multi-player and temporal constraints such as the spectrum environment, only classical reinforcement learning algorithms may be insufficient. The large number of players, short entry-exit times to the spectrum, and the large size of the action-medium space make this problem a high-dimensional problem. This situation is known as the ‘curse of dimensionality’ in the literature and makes learning processes seriously difficult. For this reason, DRL methods, which combine deep learning (DL) and RL, have recently provided promising solutions in the field of dynamic spectrum management. DRL enables effective policy learning in high-dimensional state spaces, as well as more flexible and autonomous spectrum management [41]. This allows CR systems to adapt to changing environmental conditions, avoid interference, and make better use of the spectrum.

In 2009, Huang et al [42] designed a support vector machine (SVM) based learning engine to support environmental awareness and decision making in cognitive radio (CR) networks. Then, in 2010, Tumuluru et al. [43] succeeded in improving energy efficiency by spectrum prediction with an artificial neural network (MLP) based on historical channel data. In 2012, Khozeimeh and Haykin [44] presented a dynamic spectrum management method to reduce collision risks by integrating self-organising maps (SOM) into cognitive radio networks (CRNs). In the same years, a study by Yu [45] investigated spectrum sharing equilibrium with Stackelberg and cooperative game theory models. In 2013, He et al. [46] aimed to make reconfiguration decisions in CRN environment using ant colony optimisation (ACO).

In 2014, Wang et al [47] developed predictions in time and space context using SVM for spectrum mobility estimation in mobile CRNs. In the same year, Saleem and Rehmani [50] developed spectrum-aware clustering-based routing systems by introducing the SMART protocol that focuses on primary user activity patterns in CRNs. During this period, Morabit et al [51] presented a genetic algorithms (GA)

based decision mechanism for spectrum allocation based on user demand. Working on channel estimation in industrial spectrum environments, Saad et al [52] developed Hidden Markov Model (HMM)-based approaches to model primary user (PU) behaviour in multiple time slots. Yau et al [58] systematically investigated RL applications for CRNs and showed that these methods improve contextual awareness. In the same year, Pham et al. [59] developed a model that supports handoff decisions by time series analysis of channel state using HMM. Between 2015 and 2017, Xiao and Hanawal [64], [65] proposed anti-jamming strategies using RL and game theory approaches. Hanawal presented a model that jointly optimises frequency hopping propagation and data rate, while Xiao developed cooperative RL methods for power control under jamming. DL-based approaches started to gain prominence as of 2018. Liu et al. [70], [72] improved jamming decisions by learning from spectrum waterfall images and integrated heterogeneous information fusion techniques into DRL systems. Xiao et al. [73] presented an RL-based anti-jamming strategy with multidimensional decision making in time and frequency axes. The success of DL on time series inspired the use of deep recurrent neural networks (DRNN) for frequency estimation in Tang and Li [53] 2017. In [54], Selim et al. presented a deep CNN based spectrum tracking system to separate the clutter in the radar band. Subsequently, Lee et al.[55] developed collaborative spectrum sensing algorithms using CNN, enabling CR devices to make more effective decisions. Akbar et al [56] and Senthilkumar and Geetha Priya [57] studied channel state estimation and appropriate selection strategies using HMMs. Koushik et al [58] compared RL and transfer learning methods at the hardware level, while MohanaPriya and Shalinie [59] aimed to design secure routing systems using restricted Boltzmann machines (RBM).The HMM-based spectrum mobility model developed by Salgado [60] focuses on reducing data loss in channel transitions. Wang et al [56] modelled multi-channel anti-jamming access with the non-stochastic multi-armed bandit problem. In 2013, Chen [62] and Yang [63] independently conducted anti-jamming studies based on Stackelberg games and proposed models on transmission power control. These strategies allowed users to optimise their decisions in a leader-follower structure in CRNs.

Jia et al [66] proposed a game-theoretic learning approach for the anti-jamming DSA problem in dense wireless networks and dynamically modelled spectrum sharing. In the same direction, Misra et al. [67] developed a mobility-based Stackelberg game to

mitigate the jammer effect in wireless sensor networks. Aref and Jayaweera [68] developed stochastic game models for interference prevention in CRNs by proposing multi-agent learning (MARL) systems against sweep jammer. Naparstek and Cohen [69], working on deep RL-based frequency access in distributed environments, improved DSA performance with multi-user DRL algorithms. Similarly, Li et al [71] proposed real-time adaptive decision making mechanisms by developing DRL algorithms against intelligence-based jammers. Yuan et al [74] demonstrated the power of deep learning in jammer pattern extraction from time-frequency data.

Zhang et al. [75] presented DRL models that optimise modulation and coding selection in cognitive heterogeneous networks, while in the same year, Zhang [76] showed that intelligent relay communication systems with RL can reduce the jammer effect. Slimeni et al. [77] propose a method based on the cooperative QL algorithm to solve the channel selection problem in CRN. Das et al. [78] address spectrum mobility in CRNs, i.e., the process of users switching from one channel to another, and aims to optimise these transitions with QL. Following these studies, Zhang et al. proposed the Multi-Discriminator Adversarial Joint Reinforcement Learning (MDAJRL) algorithm [79], which allows joint optimisation of frequency, motion and antenna parameters in energy-constrained UAV swarms. This model reduced jamming effects by learning the multidimensional action space with RL, while at the same time improving the energy efficiency of the system. Zhang et al. [80] improved the energy efficiency of UAV data links by proposing a channel selection and power control method based on correlation vector regression. Wang et al [81] aimed to reduce the outage probability in multi-hop ad-hoc networks by cooperative anti-jamming strategies, while Li and Zhang [82] optimised trajectory and power control for jammer-assisted secure UAV communication. Liu et al [83] developed MDP-based spectrum scheduling methods based on spectrum waterfall analysis under unknown environment conditions. Xiao [84] proposed an approach where frequency and movement in communication systems are jointly optimised with hotbooting DQN. Liu [85], who provides frequency selection with heterogeneous information fusion in HF communication, and Yao and Jia [86], who work on multi-agent cooperative RL systems, have made significant contributions to improve stability and security in wireless networks. Slimeni et al [87] developed a model that combines Markov decision processes (MDP) and RL to mitigate jamming in CRNs. In this study, spectrum access decisions are improved

based on environmental feedback in CRN environments, and the system is adapted to environmental dynamics by using RL algorithms within the MDP framework. This method aims to increase the accuracy and speed of learning by providing a more structured decision process compared to classical QL.

Slimeni et al. [88] propose a modified QL algorithm to prevent jamming attacks in CRNs. Again Slimeni et al. in [89] propose a cooperative QL based method to solve the channel selection problem in CRNs. Finally, the two-dimensional DRL model presented by Han [90] proposed a time and frequency based scheduling method to minimise the jamming effect in electromagnetic environment. These studies reveal that the joint scheduling of spectrum, power, motion and beamforming parameters in UAV communication systems has not yet been sufficiently addressed, and integrating the multidimensional RL approach into the literature is an effective method. In summary, it shows that support vector machines and artificial neural networks were initially used in the literature; in recent years, DL (CNN, DRNN) and RL based approaches have come to the fore. Over time, there is an evolution from classical methods to more adaptive, data-driven and environmentally sensitive systems. These new generation methods can provide much better results, especially in situations where environmental conditions are constantly changing and uncertainties prevail. However, in order to achieve this success, it is of great importance to provide the right data to the model and to adjust the methods according to the environment.

## **1.12. DRL**

In this study, four different DRL algorithms are evaluated to solve the dynamic spectrum management problem: Deep Deterministic Policy Gradient (DDPG), Twin Delayed Deep Deterministic Policy Gradient (TD3), Soft Actor-Critic (SAC), and Single-Agent TD3. Each of these algorithms is a modern and powerful method for decision making in continuous action spaces. Their main features, technical approaches and strengths/weaknesses are summarised below.

The DDPG algorithm is a model-free and off-policy reinforcement learning method with actor-critical architecture developed for decision making in continuous action spaces [91]. In this algorithm, the actor network, which represents the policy, generates the best action based on the current scenario, while the critic network analyses its

value. By combining QL with policy gradient techniques, DDPG can do both value estimation and policy learning. One of its distinguishing properties is its ability to function in continuous action spaces, and its model-free structure allows it to accomplish learning without explicitly simulating the dynamics of the environment. However, DDPG also has some disadvantages such as the tendency of overestimation bias in Q-function estimation and limited exploration capability.

TD3 is a DRL method developed to reduce the overestimation bias in Q-function estimation, which is one of the most important problems faced by the DDPG algorithm [92]. This algorithm provides a more reliable and stable learning process by adding three main improvements on the DDPG architecture. First, the twin critics approach uses two separate Q-networks and suppresses overestimation by using the minimum value obtained from these two networks in updates. Second, with the delayed policy update technique, the actor network is updated less frequently, which leads to more stable policy learning. Thirdly, target policy smoothing is applied to add a small amount of noise to the target actions, which prevents the policy function from making too sharp decisions. While the TD3 algorithm provides more robust and stable results, especially in continuous and high-dimensional action spaces, it requires a higher computational cost compared to DDPG in return for this performance improvement.

Soft Actor-Critic (SAC) is a powerful algorithm developed in the field of RL that aims to both maximise reward and increase the agent's exploration ability [93]. The basic approach of this algorithm is not only to select the actions that bring the highest reward, but also to learn a policy that is open to trying different actions. This is achieved by increasing the entropy - i.e. the uncertainty - of the policy distribution. Thus, the agent is not just stuck in certain patterns, but continues to explore its environment more broadly. In this respect, SAC more effectively balances exploration with exploitation of the best known actions. Thanks to its off-policy and model-free nature, it can learn from past experiences and produce effective solutions in complex environments. In this way, the exploration-utilisation balance is achieved more successfully. One of the biggest advantages of SAC is that it encourages the agent to make bolder explorations through entropy maximisation. It also has very stable learning processes and is relatively less sensitive to hyperparameter settings. Disadvantages include a more complex theoretical background and a high computational cost during training.

Single-Agent TD3 is a version of the classical TD3 algorithm implemented with a single agent instead of multi-agent environments [92]. It retains all the technical advantages of TD3, but assumes only a single decision maker. In environments such as spectrum sharing, this structure may be preferable when a single decision maker is trying to learn the behaviour of the system. This structure has less complexity and can be computationally more efficient. However, when a large number of actors participate in the spectrum, the single agent assumption loses its realism and limits the generalisability of the model. As this construction relies more on the use of the TD3 algorithm as a reference, additional references can be made to the original TD3 paper.



## **2. PURPOSE OF THE STUDY**

### **2.1. Increasing Strategic Importance of UAV Systems**

UAVs have become an indispensable system not only in military missions but also in civil and industrial areas. Their use in many areas such as post-disaster emergency communication, coverage of remote rural areas, data-based decision support systems in agriculture, environmental monitoring, logistics distribution and urban mobile services has made UAVs the key to highly mobile, positionable and infrastructure-free communication solutions. The features of UAVs such as high manoeuvrability, LoS advantage, high mobility, dynamic positioning flexibility and infrastructure-independent operation have made them innovative communication nodes that go beyond traditional wireless communication networks.

In the Ukraine-Russia war, unmanned aerial vehicles (UAVs) played a critical role in a wide range of missions, such as intelligence gathering, target detection, organising attacks and directing artillery fire. In particular, armed drones such as the Bayraktar TB2 enabled the Ukrainian military to achieve high impact at low cost [94]. This war demonstrated that UAVs have become not only supporting but also directly decisive elements in modern conflicts [95].

The Iran-Israel conflict has clearly demonstrated the strategic importance of unmanned systems in modern warfare. In particular, Iran's attempt to erode air defence systems by using low-cost kamikaze drones in swarms demonstrates that air dominance is shaped not only by advanced systems, but also by numerical and tactical diversity.

### **2.2. Wireless Communication in UAV Systems**

UAVs play critical roles in expanding cellular networks and meeting emergency connectivity needs in inaccessible areas, especially in tasks such as establishing air-based temporary base stations (UAV-BS), air-to-ground and air-to-air connections. This has led to the positioning of UAVs as the main component of 'network from the sky' solutions in 5G/6G network architectures. In addition, UAVs integrated with edge computing reduce data processing latency and provide end-to-end connection continuity. However, the mission success of UAV systems largely depends on the

communication quality offered by wireless communication technologies. Especially in mmWave and terahertz (THz) bands, features such as high bandwidth MIMO, beamforming, fast link establishment and mobile channel estimation are known to directly affect performance. The communication infrastructure for UAVs should provide ultra-low latency, high link reliability and energy efficiency, especially in tasks such as real-time control loops, coordinated flight planning and situational awareness.

In multi-UAV systems (UAV swarm networks), the situation becomes even more complex. Swarm structures require complex communication and network protocols, such as multi-agent interaction, distributed decision making, routing topology optimisation and collision avoidance resource sharing. Accordingly, it has become a necessity to optimise wireless communication systems for UAVs and make them adaptive to dynamic channel conditions, spectrum scarcity, jammer attacks and topology changes.

### **2.3. Two-way Dependency**

Therefore, there is a critical two-way dependency between UAV systems and wireless communication infrastructure. While UAVs contribute to wireless networks with high coverage flexibility, temporary infrastructure support and adaptive positioning, the functional performance of UAVs is directly dependent on the state of the communication channel, spectral efficiency, resilience against jamming threats and protocol optimisations.

### **2.4. The Role of DL and RL**

The strategic management of this bidirectional dependency requires intelligent system designs in which communication, movement and mission parameters are programmed together, unlike traditional wireless communication architectures. In this context, AI based approaches, especially DL models, have been widely used in recent years to make wireless communication systems more flexible, learnable and autonomous against environmental conditions. These models, which have been successfully applied in many fields such as image classification, natural language processing and

robotic control, have the potential to extract complex relationships without human intervention thanks to their representation learning capabilities over large datasets.

However, the success of deep learning based systems is largely directly proportional to the generalisation capacity of the model. While this capacity produces meaningful results in models trained with high-dimensional, diverse and high-volume datasets, this approach may be limited in high variance, low prior knowledge and rapidly changing environments. Especially in UAVs and swarm UAV systems, the communication environment is highly dynamic, uncertain and unpredictable.

The fact that wireless communication faces problems such as shared spectrum usage, multiple user interactions, spectrum scarcity, topological variability, and intentional electromagnetic interference (e.g. jammer attacks) makes it difficult to apply deep learning models in the classical sense. In these systems, the environment model to be learnt is not static; it is time and context sensitive.

For this reason, RL methods, which have the ability to make real-time decisions, adapt to the environment instantly and learn through trial-and-error, offer a more suitable option for UAV systems instead of models based solely on observation data and trained offline. RL enables UAVs to make adaptive decisions against environmental variability by enabling agents to learn reward-based learning based on the outcome of their actions in an unknown environment.

## **2.5. DRL and Multidimensional Decision Problems**

However, classical RL algorithms are computationally inefficient, especially in high-dimensional action and state spaces, and are generally useful in simplified simulation environments. In this context, DRL models, which combine the representational power of deep learning with the online adaptation capability of reinforcement learning, have significant potential for UAVs to learn and perform complex communication tasks. Especially in multidimensional environments. For example, in scenarios where parameters such as frequency selection, beamforming direction determination and position update of UAVs need to be handled simultaneously - it is necessary to go beyond classical DRL approaches and develop multi-dimensional DRL algorithms. In this context, data-driven artificial intelligence techniques, such as multi-dimensional DRL, have come to the forefront to optimise the mobility and communication

strategies of UAVs together and in synchrony. This research direction is of great importance not only for UAV applications, but also for critical technological breakthroughs such as future autonomous network systems, 6G-Beyond communication paradigm, network slicing and end-to-end integration of cyber-physical systems.

## 2.6. Contribution of the Study

The aim of this study is to develop reliable and efficient wireless communication strategies against jammer attacks in multi-agent UAV networks. Much of the literature has focused on QL and DQL algorithms that are effective in discrete action spaces. However, in realistic UAV applications, continuous action spaces - for example, parameters such as transmission power levels, channel selection, target selection and relay utilisation - are of critical importance. This situation limits the application area of classical QL based methods and creates the need for more advanced algorithms that can handle continuous action spaces in a natural way.

In this context, TD3, DDPG, SAC, SingleAgentTD3, Random and Simple Action Rule-Based (SAR) algorithms, which are up-to-date and powerful model-based approaches that can be directly used in continuous action spaces, are preferred in our study. Three of these algorithms are centralised (TD3, SAC, DDPG) and three are distributed (SingleAgentTD3, SAR, Random) to provide a comprehensive comparison in terms of both learning paradigms and network structure. In the literature, there is no comprehensive benchmark study in which these six algorithms are compared simultaneously in the same environment against the same jammer scenarios. In this respect, the study provides a new methodological framework.

Furthermore, by testing the performance of the algorithms under different jammer types-comb, sweeping, dynamic, random, adaptive and smart jammers-the robustness of the algorithms against different types of attacks is analysed in detail. For each algorithm-jammer combination, metrics such as average reward, throughput, energy consumption and energy fairness are presented comparatively.

The observation space of the model consists of a multidimensional, context-sensitive representation of the Bit Error Rate (BER), signal-to-interference plus noise ratio (SINR), channel utilisation, battery status and topological gain for each UAV. The

action space consists of 5 different continuous values for each UAV such as [power rate, relay selection, relay utilisation, channel selection, target selection]. This rich representation supports fine-tuned decision making for each agent in accordance with the dynamic nature of the environment.

In this context, the proposed work aims at the following innovations both theoretically and practically:

- Benchmarking centralised and distributed learning structures in the same environment,
- Convergence to realistic UAV scenarios with extended observation and action spaces,
- Establishing a common learning framework over multidimensional decision parameters (power, channel, target, relay),
- Systematically evaluating the resistance levels of algorithms against different jammer types.

In conclusion, this study not only presents a new methodological approach, but also provides an important reference for the design of multi-agent, dynamic and threatened UAV networks.

## 3. EXPERIMENT

### 3.1. Problem Statement

Herein, we evaluate the performance of reinforcement learning-based methods that aim to establish reliable, efficient and fair communication in multi-agent UAV networks under jammer threat. Modern UAV networks are used in applications such as military operations, search and rescue, disaster management and data collection in large areas, but these networks are highly vulnerable to external interference, especially jammer attacks.

SwarmEnv, the simulation environment proposed in this study, provides a real-world modelled communication environment with a multi-dimensional continuous action domain. Each UAV agent, as shown in **Figure 3.1**, has the following five decision parameters:

- **Power fraction:** A normalised continuous value of the transmit power that the UAV will use for signal transmission. This value scales linearly between the minimum transmission power and the maximum power. As the power increases, SINR and success rate increase, but energy consumption also increases,
- **Target UAV selection:** Specifies the target UAV to which the data will be sent. Just like in relay selection, this is also generated as a continuous value and applied by indexing according to the number of target UAVs. Both topology and signal quality are effective in target selection,
- **Relay UAV selection:** It is a continuous value that indicates whether the UAV will select another UAV as a relay to transmit the signal. This value is indexed and normalized according to the number of UAVs. Thus, the agents make the relay decision with a continuous output among the other UAVs,
- **Relay usage decision:** This decision determines whether the transmission will be direct or via relay. It is a binary variable: 0 means direct transmission, 1 means using a relay. When SINR is low, using a relay may be more advantageous,
- **Channel selection:** It determines the frequency channel that the UAV will use for transmission. It is a continuous value normalized according to the number of channels. Proper channel selection is critical to avoid jammer effects and reduce conflicts.

Since these decisions are represented by continuous values, classical discrete action based algorithms (e.g. QL, DQN) cannot learn this structure efficiently. Moreover, the problem is multi-dimensional and multi-objective since the decisions of the UAVs not only affect their own performance but also the energy consumption and the overall efficiency of the network. In the simulation environment, channel gains are dynamically calculated based on the path-loss model; jammer effect is modelled in terms of SINR and energy consumption varies depending on the transmission power and relay decisions. Jammer types are defined with different strategies such as adaptive, random, sweeping, dynamic and smart, reflecting the diversity of real-world attacker behaviour.

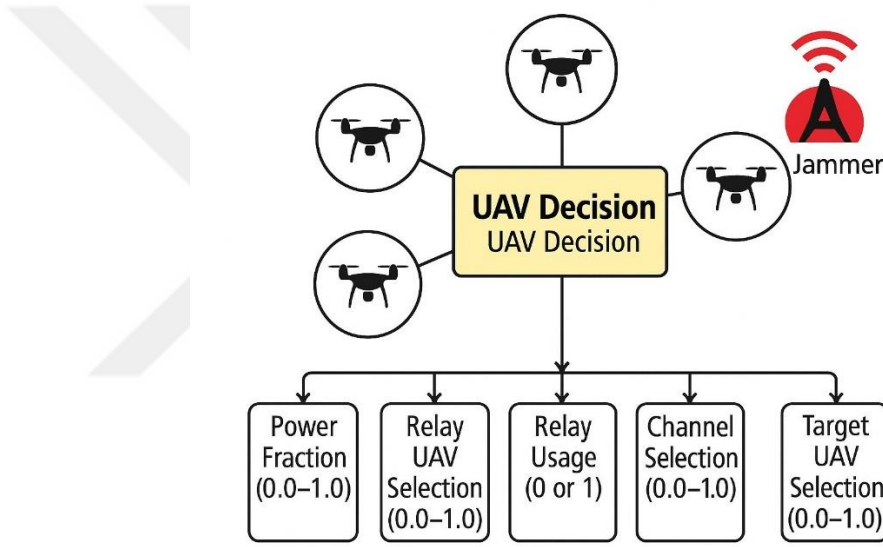


**Figure 3.1:** UAV Decision Parameters

Accordingly, the following algorithms are tested:

- Centralised Learning: TD3, DDPG, SAC
- Distributed Learning: SingleAgentTD3
- Rule-based: SAR
- Randomised: Random

The algorithms were evaluated on the following key performance metrics:

- Average Reward: Depends on factors such as SINR, energy and outage (failed transmission),

- Successful Packet Transmission (Throughput): Number of successful packets transmitted,
- Total Energy Consumption (Energy): Total energy consumed by UAVs,
- Fairness: Energy distribution balance between UAVs based on Gini coefficient,
- Packet Drop Rate: Packets that failed due to SINR or TTL.

In conclusion, the problem addressed in this simulation requires not only classical routing solutions, but also DRL algorithms due to the continuous action space and multivariate learning requirements. In this context, the experimental studies investigate how UAV networks can adapt under complex and hostile environments.

## 3.2. Environment

The simulation environment developed in this study is SwarmEnv, a specialised environment that realistically models multi-agent tactical UAV communication scenarios. SwarmEnv is designed as an artificial intelligence learning domain that includes both physical communication parameters and multidimensional and continuous decision spaces.

As shown in **Figure 3.2**, the environment consists of a communication topology including source UAV, target UAV, jammer, communication channels and optional relay UAV. In the environment, each UAV can transmit data both directly and using relay for a given mission duration. The jammer tries to intercept this communication by targeting active channels.

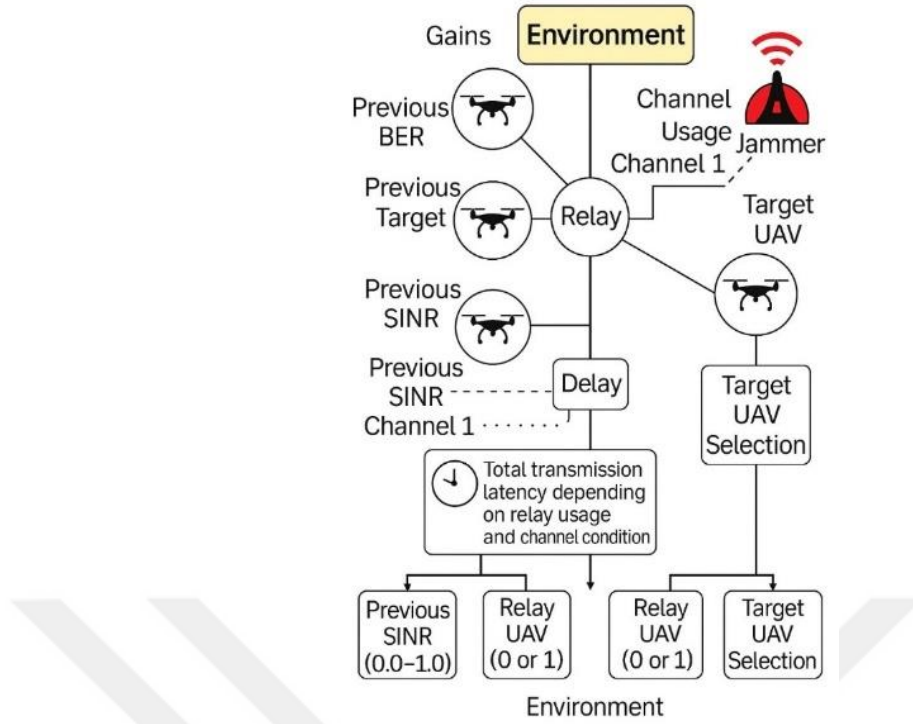


**Figure 3.2:** Enviroment

The environment includes the following basic elements and observation-action relationships:

- Gains (Channel Gains): Channel quality between UAVs is determined by the log-distance path-loss model.
- Channel Usage: The channel attacked by the jammer at each step and the past usage intensity of the channel are observed.
- Previous BER & SINR: The UAV receives information about the quality of the previous transmission (bit error rate and signal-to-noise ratio).
- Delay: The total transmission time varies according to relay utilisation and channel quality. This is tracked as an output observed by the environment.
- Battery/Energy Use: The energy consumed during transmission depends on parameters such as power used and relay.
- Jammer: As shown in the figure, the jammer attacks specific channels according to different strategies and affects the quality of communication.

All of these components are presented diagrammatically in **Figure 3.2**. As shown in the diagram, environmental observations are taken from top to bottom and actions produced by the agent interact with the environment. Actions are multidimensional

and continuous values such as target selection, power determination, relay utilisation, channel selection and relay UAV selection. This structure requires DRL algorithms unlike classical discrete-alien approaches.

SwarmEnv takes into account various dynamics (e.g., channel interference, energy efficiency, jammer threat, routing decision) encountered in real-world UAV communication scenarios, increasing both the academic validity and application suitability of the model.

### 3.2.1. Observation Space

Each UAV can only observe the environment from its own local perspective. The following items constitute the agents' observation vector:

- Previous BER: The UAV observes the success level of the transmission it performed in the previous step with the BER value. This value is calculated by the medium using the **Equation 3.1** [96], [97]. BER represents the error information conveyed by the receiver and serves as an indirect yet meaningful success signal for the learning algorithm.

$$BER = e^{-SINR} \quad (3.1)$$

- Previous SINR: The UAV observes the SINR value obtained in the last step of transmission. SINR formula [98] is normalized and presented to the model on a logarithmic scale as in **Equation 3.2**. This transformation allows SINR values across a very wide range to be presented to the learning algorithm more stably.

$$Normalized\ SINR = \frac{\log_{10}(SINR + \varepsilon)}{c} + 1.0 \quad (3.2)$$

- Channel Gains: Each UAV observes the channel gains representing the connection quality with all other UAVs. These gains are calculated based on distance using the log-distance path loss model in **Equation 3.3** [99].

$$G_{i,j} = 10^{-(\alpha \cdot 10 \cdot \log_{10}(d_{i,j}) + PL_0) / 10} \quad (3.3)$$

Here,  $d_{i,j}$  represents the distance between UAVs, the  $\alpha$  represents path-loss coefficient, and  $PL_0$  represents the reference path loss.

- **Battery Level:** Indicates the current energy level of the UAV. Each transmission action consumes energy depending on the transmission power and relay usage.
- **Previous Outage:** A binary variable that indicates whether the previous transmission failed or not. SINR is observed as "outage = 1" when it falls below a certain threshold value.
- **Channel Usage History:** UAV observes how much each communication channel was used in the previous step. This information provides guidance for agents to prefer less congested channels.
- **UAV ID (Agent ID):** A value that represents the UAV's identity and is normalized in the range  $[0,1]$ . This feature ensures the distinguishability of agents and facilitates the management of heterogeneous multi-agent systems.

The SINR and BER metrics present in the observation space provide agents with information about both the physical quality of the environment and the outcomes of their own decisions. SINR is a measurement that directly reflects the physical conditions of the environment and provides the instantaneous effect of environmental interactions. BER shows how successful the transmission performed by the UAV is under these environmental conditions. Although the BER value can be mathematically calculated from the SINR, BER is a more behavioral indicator of success. Therefore, having both pieces of information in the observation space is beneficial for agents to learn about both the environment and the consequences of their own actions.

### 3.2.2. Action Space

Each UAV generates a five-dimensional action vector that contains continuous values. Each dimension of this action vector governs a distinct aspect of the UAV's communication strategy:

- **Power Fraction:** The first component of the action vector,  $a_0 \in [0,1]$ , determines the UAV's transmission power. This value scales linearly between the lowest and highest power restrictions. The appropriate power value is derived using **Equation 3.4** [100].

$$P_{actual} = P_{min} + a_0 \cdot (P_{max} - P_{min}) \quad (3.4)$$

- **Relay UAV Selection:** The second component,  $a_1 \in [0,1]$ , allows the UAV to select the relay UAV for data transfer. The selection process in **Equation 3.5** is derived as follows. Firstly, the agent chooses among other UAVs excluding its own identity. Since  $a_1$  is between 0 and 1, it is converted to a range from 0 to  $N-1$  by  $a_1 \times (N-1)$ . This value is converted to an integer by rounding down with the operation  $\lfloor \cdot \rfloor$ . Thus, an integer between 0 and  $N-1$  is obtained. To this integer  $i+1$  is added. The aim here can be to skip the relay indicated by  $i$  or to select the next available relay. The total number is taken modulo  $N$  (i.e. the remainder divided by  $N$ ), so that the result is between 0 and  $N-1$ . This prevents going outside the array and allows the index to continue cyclically.

$$relay = (\lfloor a_1 \cdot (N - 1) \rfloor + i + 1) \bmod N \quad (3.5)$$

where  $i$  is the index of the deciding UAV.

- **Relay Usage:** The transmission path is determined by the third action component,  $a_2 \in [0,1]$ . If this is greater than 0.5, the signal is routed through a relay to the destination. If the value is 0.5 or lower, the signal is transmitted directly to the destination.
- **Channel Selection:** The fourth component,  $a_3 \in [0,1]$ , selects the transmission channel. The active channel index is determined by multiplying the value by the total number of channels ( $K$ ) (**Equation 3.6**).

$$channel = \lfloor a_3 \cdot K \rfloor \quad (3.6)$$

- **Target UAV Selection:** The final component,  $a_4 \in [0,1]$ , determines the UAV to communicate to. The own UAV ID is eliminated, same as it is for relay selection. The destination UAV is determined as **Equation 3.7**.

$$target = (\lfloor a_4 \cdot (N - 1) \rfloor + i + 1) \bmod N \quad (3.7)$$

Thanks to this structure, each UAV makes decisions in a fully continuous and multidimensional action space. This necessitates DRL methods that can directly process the continuous action space, rather than classical discrete action-based algorithms.

### 3.2.3. Channel and SINR Calculation

In the simulation environment, the link quality of each UAV with other UAVs is physically calculated using the log-distance path-loss model. This model is used to determine the attenuation of the signal propagation as a function of distance.

The channel gain between two UAVs is defined by the following **Equation 3.8**.

$$G_{i,j} = 10^{-\alpha \cdot \log_{10}(d_{i,j}) - PL_0/10} \quad (3.8)$$

where

- $G_{i,j}$ : Channel gain between UAV  $i$  and UAV  $j$ ,
- $d_{i,j}$ : The Euclidean distance between two UAVs,
- $\alpha$ : Path-loss coefficient (e.g. 2.0),
- $PL_0$ : is a fixed value taken as reference path loss (e.g. 15 dB).

During transmission, the SINR value obtained is calculated differently according to the direct transmission or relaying scenarios according to the decision of the UAV (**Equation 3.9**).

$$SINR_{direct} = \frac{P \cdot g_{i,t}}{\sigma^2 + J_c} \quad (3.9)$$

where:

- $P$ : Transmission power selected by the UAV,
- $g_{i,t}$ : Channel gain between the source UAV and the destination UAV,
- $\sigma^2$ : Ambient noise power,
- $J_c$ : The interference strength applied by the jammer on the relevant channel.

in case of relay transmission, SINR is calculated as **Equation 3.10**.

$$SINR_{relay} = \frac{P \cdot g_{i,r} \cdot g_{r,t}}{\sigma^2 + J_c} \quad (3.10)$$

where:

- $g_{i,r}$ : channel gain between source UAV and relay UAV,
- $g_{r,t}$ : channel gain between relay UAV and target UAV.

This SINR value is the main criterion that determines whether the UAV can transmit successfully or not.

### 3.2.4. Energy Model

Energy management of UAVs is an important optimisation dimension in the simulation environment. Each UAV's transmission action results in different energy costs depending on the power level it uses and whether it activates the relay mechanism or not. Direct Transmission Energy Consumption is obtained as in **Equation 3.11**.

$$E_{direct} = P \quad (3.11)$$

where  $P$  is the actual transmission power selected by the agent.

Transmission Energy Consumption over Relay is calculated as in **Equation 3.12**.

$$E_{relay} = P + 0.1 + 0.01 \cdot d_{relay} \quad (3.12)$$

Here,

- $P$ : transmission power of the source UAV,
- $d_{relay}$ : distance between relay UAV and target UAV,
- 0.1: fixed additional cost for relay commissioning,
- $0.01 \cdot d_{relay}$ : distance-dependent energy loss.

This model reflects more precisely the impact of relay utilisation on energy. The agent considers this cost when deciding between direct or relay transmission to improve energy efficiency.

### 3.2.5. Jammer Types

Six different jammer types were applied in this study:

- Adaptive Jammer: It dynamically distributes its power to the most intensively used channels by considering the intensity of channel usage. This model aims to perform effective jamming by targeting the most preferred channels in communication.
- Comb Jammer (Scan Based Jammer): At each step, it attacks the first two most heavily used channels with a constant power. Thus, it detects the density in the system and focuses on specific channels.

- Sweeping Jammer: Cyclically scans all channels over time. It aims to jam the entire spectrum of the system by targeting the next channel at each step.
- Dynamic Jammer: Alternates between Comb and Sweeping models. Thus, it combines both intensity-based and time-based jamming behaviours.
- Random Jammer: Targets a random channel at each step. It exhibits an unpredictable jamming behaviour, making it difficult for UAVs to make predictions.
- Smart Jammer: Analyses previous channel usage history and targets the most heavily used channels. This model aims to perform a more effective intervention with learning-based jamming behaviour.

Those experienced in studying various jammer models generate a range of interference challenges within the environment, enabling a comprehensive evaluation of the robustness and adaptability of the algorithms being tested.

### **3.3. Methodology**

#### **3.3.1. Problem Statement and Objective Function**

The problem addressed here is to develop reliable, energy efficient and successful communication strategies for multiple UAVs in an environment under intense and variable jammer threat. Each UAV can transmit data both directly and through relay UAVs. There are multiple communication channels in the environment and these channels are subject to different jammer attacks depending on time. Furthermore, each UAV has a limited energy budget and all actions lead to energy consumption.

Each agent (UAV), based on its local observation:

- Target UAV selection,
- Channel selection,
- Determining the power level,
- Deciding to use relay,
- Relay UAV selection

has to be performed. The goal is to optimise the performance of these decisions under jammer effects and network congestion, while maximising the overall throughput, energy efficiency and fair channel utilisation of the network in the long run.

Regarding objective function, each agent tries to maximise its rewards over the time steps. The general reward function used is defined as **Equation 3.13**.

$$R_t = \lambda_1 \cdot Throughput - \lambda_2 \cdot Energy - \lambda_3 \cdot DropPenalty \quad (3.13)$$

Here:

- Throughput: the amount of data successfully transmitted in a step (positive reward in case of successful transmission),
- Energy: the energy consumed during transmission (high energy consumption has a negative impact),
- DropPenalty: the negative penalty applied in case of failure due to packet timeout (TTL) or very low SINR,
- $\lambda_1, \lambda_2, \lambda_3$ : These are weight coefficients determined according to the system design.

This objective function rewards not only successful transmissions, but also efficiency and consistency, allowing for the development of a more balanced and durable communication plan. **Figure 3.3** represents the rewarding mechanism which was used in experiments.

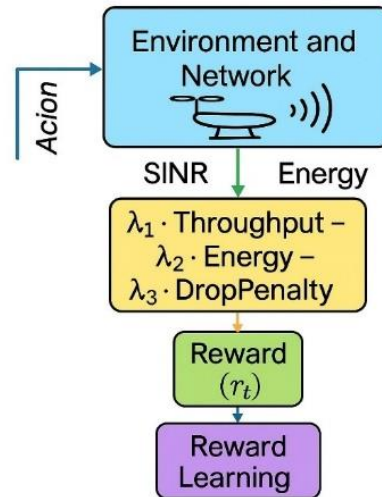


**Figure 3.3:** Rewarding Mechanism

### 3.3.2. Timing and Transmission Order

In the simulation scenario, just one UAV is activated and responds at each step. The selection of this UAV is done using a round-robin approach based on the agent ID. So, the agent active at step  $t$  is determined by the formula  $i = t \bmod N$ . Thanks to this structure, all agents take environmental observations in turn, produce actions, and interact with the environment. Thus, coordination and learning in a multi-agent environment are achieved more stably.

### 3.3.3. Target Selection and Transmission Model

Each UAV selects only one target UAV at each transmission step. This target UAV is selected from among the other agents, which are different from the source UAV. Therefore, a one-to-one (unicast) communication model is implemented in the system. Broadcast or multicast do not exist in this model. This design both simplifies the communication protocol and allows for a more distinct observation of the competition among agents.

### 3.3.4. Transmission Type

The UAV can transmit directly to the target, or optionally use another UAV as a relay. The use of a relay depends on the action vector produced by the agent and can be affected by factors such as channel conditions, energy costs, and jammer effects. This flexible communication model allows agents to learn different strategies depending on the situation.

### 3.3.5. Learning Parameters and Algorithm

In the study, DRL algorithms suitable for continuous action space are used to enable UAVs to learn efficient and reliable communication strategies in a jammer-affected environment. The selected algorithms were chosen due to their multidimensional decision making and their ability to cope with challenging environment dynamics as DDPG, TD3, SAC, SAR and Random Policy. The learning parameters which were used for the algorithm in experiments are exhibited in **Table 3.1**.

**Table 3.1:** The Learning Parameters

| PARAMETER                  | VALUE     | EXPLANATION                  |
|----------------------------|-----------|------------------------------|
| Total Training Steps       | 200.000   | The Number of Step Taken     |
| Replay Buffer Size         | 100.000   | The Memory Size              |
| Batch Size                 | 256       | Number of Data Sampled       |
| Discount Factor            | 0.99      | Importance of Future Rewards |
| Actor Learning Rate        | 0.0001    | Learning Rate for Actor      |
| Critic Learning Rate       | 0.001     | Learning Rate for Critic     |
| Policy Update Frequency    | 2         | Actor Policy Is Updates      |
| Target Network Update Rate | 0.005     | Soft target update           |
| Epsilon Initial / Minimum  | 1.00/0.05 | Epsilon Greeedy              |
| Noise Std. Deviation       | 0.2       | Standard deviation           |
| Random Seed                | 42        | Pseudo Random                |

### 3.3.6. DDPG

DDPG is an actor-critic based DRL algorithm developed for use in environments with continuous action space. DDPG combines the deterministic policy gradient approach with deep learning to provide successful policy learning in continuous and high-dimensional action spaces. Its structure consists of four basic components:

- Actor network: Generates a deterministic action in response to the state (**Equation 3.14**) [101].

$$a_t = \mu(s_t / \theta^\mu) \quad (3.14)$$

- Critic network: Estimates the expected cumulative reward (Q-value) for a state-action pair as in **Equation 3.15** [101].

$$Q(s_t, a_t | \theta^Q) \quad (3.15)$$

- Target Actor and Target Critic Networks ( $\mu'$ ,  $Q'$ ): They are soft copies of the main networks. These networks are updated by soft update method to increase stability in the training process (**Equation 3.16**) [101].

$$\theta' \leftarrow \tau \theta + (1 - \tau) \theta' \quad (3.16)$$

In the learning process, the agent first generates a deterministic action through the actor network in response to the current state. To encourage exploration, a time-correlated noise is added to this action, such as the Ornstein-Uhlenbeck process (**Equation 3.17**) [101].

$$a_t^{explore} = \mu(s_t | \theta^\mu) + N_t \quad (3.17)$$

The transitions  $(s_t, a_t, r_t, s_{t+1})$  that occur at each step are stored in the experience replay buffer. During learning, a random sample is taken from the data in this memory, thus reducing data correlation and stabilising the learning process. Next, the target Q-value is calculated using the Bellman equation (**Equation 3.18**) [101].

$$y_t = r_t + \gamma \cdot Q'(s_{t+1}, \mu'(s_{t+1}) | \theta^{Q'}) \quad (3.18)$$

The Critic network is updated to minimise the following loss function (usually MSE or Huber loss is used) as in **Equation 3.19** [101].

$$L(\theta^Q) = \frac{1}{N} \sum_{i=1}^N (Q(s_i, a_i | \theta^Q) - y_i)^2 \quad (3.19)$$

After that, the Actor network is optimised by the deterministic policy gradient method as in Equation 3.20.

$$\nabla_{\theta^\mu} J \approx \frac{1}{N} \sum_{i=1}^N \nabla_a Q(s, a | \theta^Q)|_{a=\mu(s)} \cdot \nabla_{\theta^\mu} \mu(s | \theta^\mu) \quad (3.20)$$

Finally, the parameters of the parent networks are softly transferred to the target networks as shown in **Equations 3.21 and 3.22** [101].

$$\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'} \quad (3.21)$$

$$\theta^{\mu'} \leftarrow \tau \theta^\mu + (1 - \tau) \theta^{\mu'} \quad (3.22)$$

As for the implementation details, the action space is continuous and 5-dimensional. The action vector represents strategic decisions such as channel selection, transmission

power, target selection, etc. The deterministic policy structure ensures stable learning in such high dimensional actions. The discovery mechanism is performed by adding the time-correlated noise generated by the Ornstein-Uhlenbeck (OU) process to the actor output. In this way, the noise is smoother and more controlled. The network architecture consists of two hidden layers, the Actor and Critic networks, and the ReLU activation function is used in each layer. The networks are fully-connected. In the operation of the Reward Function, each agent is penalised for energy consumption and failed transmissions while trying to maximise throughput (**Equation 3.13**).

The architecture of the learning process where the DDPG algorithm is applied in continuous action space for swarm UAVs is illustrated in **Figure 3.4**.

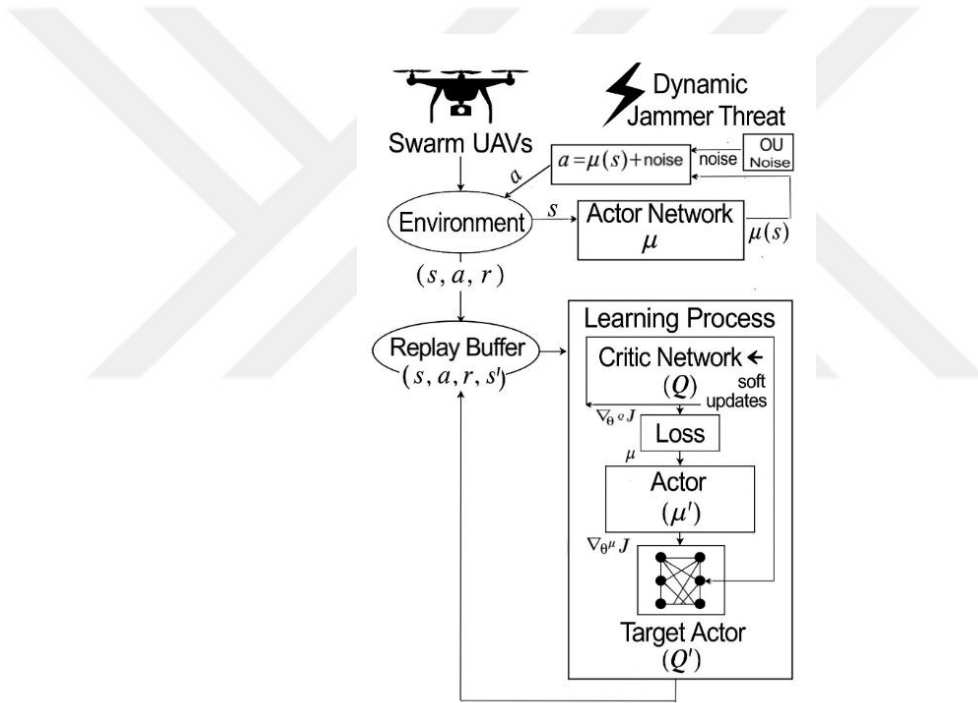


**Figure 3.4:** DDPG Algorithm Architecture Used in the Experiments

The agent generates a deterministic action  $\mu(s)$  with the Actor network  $\mu$  over the state information  $s$  received from the environment. To encourage exploration, correlated noise  $N_t$ , generated by the OU process, is added to this output to obtain the total action  $a$  and applied to the environment. The transition quadrature  $(s, a, r, s')$  resulting from each interaction is stored in the Replay Buffer. During the learning process, the Critic network is updated with the Bellman objective, while the Actor network is optimised with the deterministic policy gradient method. The target networks  $(Q', \mu')$  are updated

smoothly by Polyak averaging to increase the stability of learning. This structure enables the learning of communication policies that are reliable, energy efficient and resilient to jammer effects.

### 3.3.7. TD3

The TD3 implementation is characterised by three important improvements that address two main weaknesses of the DDPG algorithm: Q-overestimation error reduction, delay in policy updates and target policy smoothing. TD3 algorithm is based on actor-critic architecture and has an advanced structure where each component is supported by a target network. The learning process of the TD3 algorithm is shown as a block diagram in **Figure 3.5**.

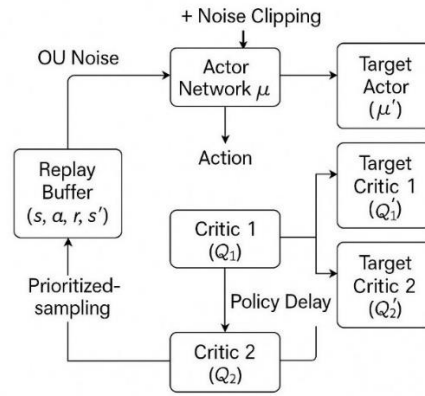


**Figure 3.5:** TD3 Learning Architecture

In this diagram, the basic components that enable stable policy learning in continuous action spaces and the data flow between them are presented in detail. The learning process starts with the actor network  $\mu$  generating a deterministic action in response to current state (observation) information. To this action, a noise generated by a time-correlated OU process is added to encourage exploration. This added noise is limited to a certain range (noise clipping) to prevent extreme outliers and unbalanced action generation. By applying the action to the environment, a new state, reward and transition information  $(s_t, a_t, r_t, s_{t+1})$  is obtained. These transitions are stored in a prioritised Replay Buffer that performs TD-error based sampling. Thanks to this structure, sample efficiency is increased by prioritising the transitions that can contribute the most to learning.

As can be seen in **Figure 3.5**, TD3 algorithm uses two separate value networks (Critic 1 -  $Q_1$  and Critic 2 -  $Q_2$ ) unlike DDPG. This double network structure helps to avoid overestimation bias in the Q-value estimation. The target Q-value is calculated as the minimum of the values from these two target critic networks as shown in **Equation 3.23** [102].

$$y_t = r_t + \gamma \cdot \min(Q'_1(s_{t+1}, \mu'(s_{t+1})), Q'_2(s_{t+1}, \mu'(s_{t+1}))) \quad (3.23)$$

Another fundamental difference is the frequency with which the actor network is updated. In TD3, the actor network is not updated at every step, but only at certain intervals. This strategy, called policy delay, increases the stability of policy updates by waiting for critic predictions to become more reliable. Finally, the target versions of the actor and critic networks (Target Actor -  $\mu'$ , Target Critic 1 -  $Q'_1$ , and Target Critic 2 -  $Q'_2$ ) are updated smoothly using Polyak averaging (**Equation 3.24**) [102].

$$\theta' \leftarrow \tau \theta + (1 - \tau) \theta' \quad (3.24)$$

This mechanism stabilises the learning process by preventing the target networks from reacting to sudden changes. With the combination of all these components, TD3 provides stable, reliable and sample-efficient learning performance in high-dimensional and continuous action spaces.

Classical applications of TD3 have mostly been tested in stationary and deterministic physical environments, for example simulations such as MuJoCo or Roboschool [94]. Such environments have relatively low complexity under the control of single agents. However, the TD3 architecture proposed in this study is applied to a multi-agent and dynamic UAV swarm environment, and a number of improvements and application-specific adaptations have been made to the TD3 structure in accordance with the complexity of this environment. In the following, we describe the advantages and structural differences of the proposed model compared to the general TD3 structure:

- **Improvement of the Discovery Mechanism:** In the classical TD3 structure, discovery is provided by clipped Gaussian noise added to the target policy output. In this study, both Ornstein-Uhlenbeck (OU) noise and clipped Gaussian noise are applied together. The OU noise produces more stable actions with time dependence and prevents the swarm UAVs from making

sudden and unstable manoeuvres, thus learning is more stable. Noise clipping also contributes to safety by preventing actions from exceeding physical boundaries.

- **Prioritised Replay Buffer:** In the original implementations of TD3, the classical Replay Buffer is used and samples are randomly selected. In this work, prioritisation based on TD-error is used and the learning process is optimised to focus on critical experiences. This approach enables faster learning of sensitive situations, especially in UAV environments where dynamic and disturbing factors (e.g. jammer effect) change frequently.
- **Reducing Overestimation with Twin Critic Networks:** The twin critic architecture, one of the key components of TD3, has been successfully implemented in this study. Two independent Q-networks ( $Q_1$  and  $Q_2$ ) produce separate value estimates for the same transitions and the target Q-value is calculated by choosing the minimum. This strategy reduces overestimation bias and ensures that policies are based on more realistic values. Critic structures are designed to be used in both centralised and single-agent perspectives.
- **Delayed Policy Updates:** The policy network is updated at regular intervals rather than at each step. This delayed update helps the learning process to proceed in a more balanced manner, preventing crucial value functions from impacting policy before they are mature enough. This method is especially important during the early learning stages. This approach is especially critical in the early learning phases.
- **Inclusion of Physical Network Conditions in Learning:** In classical versions of TD3, physical parameters are usually excluded from learning. In this study, parameters such as SINR, TTL, energy consumption, fair channel utilisation are included in the learning process. In this way, agents produce policies that are not only reward-oriented, but also optimised in terms of communication quality and resource efficiency. This feature brings TD3 one step closer to real world applications.
- **Multi-agent Compatibility** The classical TD3 framework is designed to work only in single agent scenarios. In this work, both centralised control and per-agent learning (SingleAgentTD3) are supported. This flexibility makes it

possible to successfully implement TD3 in systems that require distributed decision making, such as swarm UAV systems.

Thanks to the structural improvements and environmental adaptations described above, the TD3 architecture developed in this study provides a more realistic, reliable and efficient learning process in swarm UAV environments compared to conventional implementations. While remaining faithful to the basic structure proposed by Fujimoto et al. [94], a customised version of TD3 has been successfully implemented in accordance with the nature of the environment. In this context, the proposed TD3 implementation is not only a direct implementation of algorithms in the literature, but also an artificial intelligence solution enriched with an engineering perspective specific to complex environments.

### 3.3.8. SAC

SAC aims not only to maximise reward, but also to increase discovery by maximising the entropy of policies. This structure is well suited for scenarios where uncertainties and variable channel conditions prevail, such as the UAV communication environment. The components and flow of the method we applied to our simulation are shown in **Figure 3.6**.

Environment and Network Layer simulates the physical and network properties of the environment: SINR, TTL (Time To Live) duration, Package Buffer effects. Observations are fed to the agents as (obs\_all). As a result of the actions of the agents, reward is generated and stored as tuple (s, a, r, s'). Actor or also called policy network  $\mu(s)$  generates action from observations. However, this action is not deterministic. SAC samples the action using a Gaussian distribution as in **Equation 3.25** [103].

$$a \sim \tan h(\mu(s) + \sigma(s) \cdot \mathcal{N}(0,1)) \quad (3.25)$$

The log-likelihood of the action  $\log \pi(a|s)$  is also computed. This will be used in the entropy-based loss calculation. SAC uses two separate  $Q$  networks:  $Q_1$  and  $Q_2$ . This is to avoid overestimation bias. Each of them estimates  $Q(s,a)$ . The target  $Q$  value is calculated as shown in **Equation 3.26** [103].

$$Q_{\text{target}} = r + \gamma \cdot (\min(Q_1, Q_2) - \alpha \cdot \log \pi(a')) \quad (3.26)$$

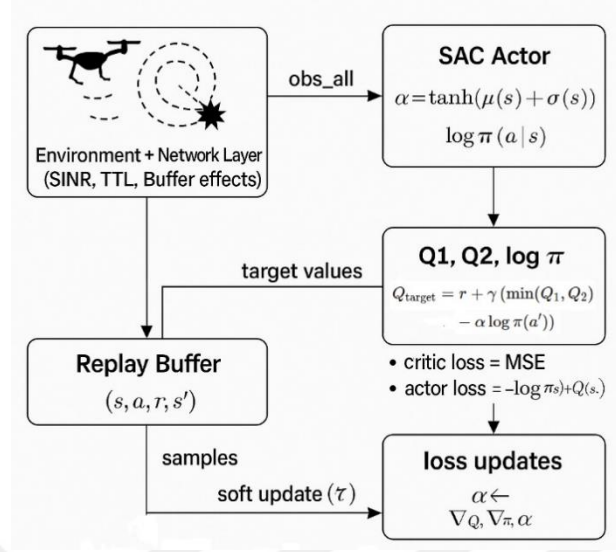


**Figure 3.6: SAC Architecture**

Here the constant  $\alpha$  is a factor that balances the entropy contribution.

We have 2 loss types as follows:

- Critic loss: Mean Squared Error (MSE) is used to compare the target and the prediction.
- Actor loss: **Equation 3.27** [103] is used for the calculation.

$$L_{\text{actor}} = \alpha \cdot \log \pi(a|s) - Q(s, a) \quad (3.27)$$

Observations, actions, rewards and new situations are stored in the Replay Buffer. Since SAC is an off-policy method, past experiences can be used repeatedly. Prioritisation according to TD-error can be applied during sampling (Prioritised Experience Replay). For loss updates, Critic, actor and entropy weight  $\alpha$  are updated separately. Backpropagation is applied to the actor network. The parameter  $\alpha$  is adaptively optimised by comparison with the target entropy. Then a soft update ( $\tau$ ) is performed. The target Q networks Q1\_target and Q2\_target are updated slowly and **Equation 3.24** works for this. This update increases the stability of the learning and

avoids immediate jumps. This update increases the stability of the learning and avoids immediate jumps.

The Soft Actor-Critic (SAC) algorithm is a high-performance method that balances exploration and stability in continuous action spaces. The traditional SAC approach aims to optimise maximum reward and maximum entropy simultaneously. The SAC implementation developed in our work sticks to this general methodology, but is extended with domain-specific requirements such as jammer effect, SINR, TTL (Time-To-Live) and buffer dynamics encountered in the UAV communication environment.

In terms of the policies applied, both approaches are stochastic (probabilistic) based. **Equation 3.25**, which is characteristic of SAC, is exactly the same in our implementation and thus, unlike deterministic policies, agents can produce different responses to various situations and behave more adaptively to the jammer effect.

As for the management of training data, a replay buffer structure that allows off-policy learning is used, as in classical SAC algorithms. However, in our approach, this structure is extended with Prioritised Experience Replay. In this way, experiences with critical conditions such as SINR loss or TTL expiry contribute more to the learning process. This increases the sensitivity to instantaneous performance degradation, especially in the UAV environment.

The entropy coefficient ( $\alpha$ ) is used in the SAC algorithm to maximise entropy and is usually set automatically. In our implementation,  $\alpha$  is optimised on a gradient basis, which allows agents to generate more diverse actions to avoid jammers. Especially considering the variability of jammer strategies, this automatic adjustment allows the agent to dynamically update the reconnaissance-decision balance in an environment where static degrees of reconnaissance would be insufficient.

In the critic networks, a double Q architecture is adopted as in the original design of SAC, and the Q1 and Q2 networks are trained separately and the minimum prediction ( $\min(Q1, Q2)$ ) is taken as the basis. This approach improves the decision quality by preventing the overestimation bias of action values.

The loss functions are defined by classical formulae as in the general SAC algorithm. The critic loss is defined as the mean square error (MSE), while the actor loss is expressed as  $-\log\pi + Q(s)$ . The soft update mechanism, which is used to ensure stability

in model updates, is implemented both in the general SAC literature and in our system with the **Equation 3.27**.

In terms of the size of the action space, our implementation differs significantly from traditional SAC testbeds (e.g. robotic simulations such as HalfCheetah). In our simulation, the agents do not just choose an action; they have to make decisions in many dimensions such as channel selection, target UAV identification, power level setting and relay utilisation. This high dimensional control problem is an area where deterministic methods often fail. Thanks to the probabilistic actions generated by SAC through tanh+Gaussian sampling, this complex control space can be managed effectively.

The reparameterisation trick used during policy updates, that is, the technique of treating the randomness in the output of the network as an external noise and making it differentiable, has been implemented with `rsample()` in the PyTorch environment and has been successfully integrated in our system. Unlike TD3, the SAC algorithm does not use target policy noise, and in our implementation this was not correctly included. Instead, the exploration is carried out directly through the entropy term.

In addition, in our system, the flow of SAC components and their relationships are visualised in detail. In particular, the information flow between the environment simulation and the SAC agent, including SINR, TTL and buffer effects, is clearly demonstrated with visual diagrams. These visualisations greatly improve the understandability in academic explanations and presentations.

In terms of application domain, while the general SAC algorithm was tested in robotic environments (e.g. HalfCheetah), our model was implemented in a much more challenging and real-world-like environment, on a UAV network operating under jammer attacks.

Finally, while sampling from the replay buffer is randomised in general SAC algorithms, our approach prioritises critical experiences with prioritised sampling, making the learning process more efficient and goal-oriented.

### **3.3.9. SingleAgentTD3**

The SingleAgentTD3 (Single-Agent Twin Delayed Deep Deterministic Policy Gradient) method implemented in this study is a variation of the classical TD3

algorithm. While the traditional TD3 architecture adopts a centralised approach covering all agents, Single TD3 is based on a decentralised learning strategy, where a separate agent is trained for each UAV. This approach is specifically designed to allow agents to learn independently and build their own policies. Our method consists of the following basic components:

- **SingleAgentTD3:** A TD3 agent that is defined separately for each UAV and contains its own actor and critic networks.
- **SingleAgentWrapper:** Creates a SingleAgentTD3 for each UAV in the environment and collects actions from these agents. It sends obs\_all observations from the environment to each agent one by one.
- **Actor/Critic Networks:** They are neural networks defined in accordance with TD3 principle. The actor network generates deterministic actions and the two critic networks predict Q-values.

Each agent learns from the centrally defined replay buffer (SingleAgentTD3Agent.agent.replay\_buffer) instead of using its own replay buffer, but the learning process is independent. The working principle of the method is as follows:

- Separate observations are taken from the environment for each UAV.
- Each SingleAgentTD3 generates an action only according to its own observation.
- Actions are combined and applied in the environment.
- Each agent adds data to the buffer according to the reward it receives as a result of its action and the new state. The learning process takes place separately for each agent:
  - actor network updates its own policy,
  - critic1 and critic2 networks are optimised with the TD3 update,
  - Target networks are updated with soft update.

Each agent is updated only with its own data and does not share information directly with other agents. The model architecture is rendered in **Figure 3.7**.

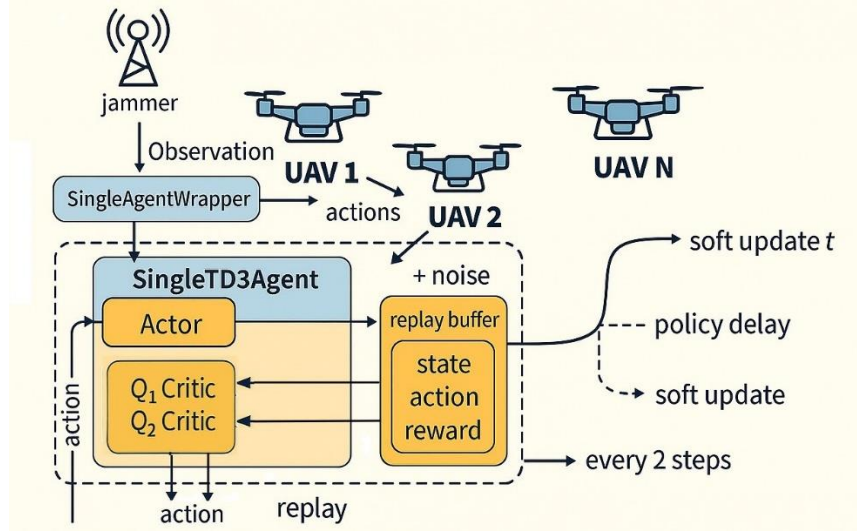


**Figure 3.7:** Single TD3 Architecture

This method was chosen to examine situations where each UAV may benefit from making decisions on its own. Since each UAV learns only from its local observations, it has the capacity to develop its own policy. In this way, the learning process becomes more customised and it can be analysed whether performance can be achieved without relying on a centralised structure.

Furthermore, this architecture is also important for assessing the scalability of the system. When a new UAV is added, a new agent can be added directly without the need to retrain the centralised policy. This provides great convenience for the transition to distributed control systems in practice. The fact that each UAV learns different policies also has the potential to reduce channel conflicts, as similar patterns of behaviour can be prevented from repeating. In the real world, as each UAV may need to make individual decisions, this structure is a more realistic and feasible solution.

During the learning process, each UAV generates actions according to its own observation. Each agent learns locally by using the data related to its own observation and replay buffer management is performed for each agent. Actions are generated by normalising with sigmoid activation and converted into scales appropriate to the expectations of the environment. The policy delay feature, which is unique to the TD3 algorithm, is preserved here and actor updates are performed every two steps. Furthermore, noise is added to the actor networks to make the learnt policies more robust and open to exploration. Target networks are updated in a stable manner with soft updating.

This structure was supposed to result in a more novel learning process, as each UAV develops strategies based on its own experience, which can be adapted to different traffic conditions and jamming scenarios. At the same time, the system gains high modularity; agents can be added or removed without requiring a centralised change. However, this structure also has some challenges. Since each agent maintains its own model, the computational cost increases and the learning time may increase. Also, since the agents do not know each other's state directly, there may be a lack of coordination. Nevertheless, with this structure, the responses of distributed systems to dynamic environmental conditions can be analysed in more detail.

### 3.3.10. SAR

The SAR method has been developed to provide a much simpler, faster, and more explainable alternative compared to reinforcement learning (RL) methods. The main goal of SAR is to enable UAVs to make decisions based on statistically significant and systematic rules, providing a strong initial performance without the need for learning. Thus, a baseline comparison method is provided for the simulation environment, and the limits of "rule-based systems" are tested. The SAR agent generates actions for each UAV by following the steps below shown in **Figure 3.8**:

- Target Selection: The UAV selects the one with the highest channel gain among the other UAVs as the target (**Equation 3.28**).

$$target = arg \max_{j \neq i} (g_{ij}) \quad (3.28)$$

- Relay Selection (Relay Selection): Among the possible relay candidates, the one with the highest total gain on the source  $\rightarrow$  relay  $\rightarrow$  destination path is selected as in **Equation 3.9**.

$$relay = arg \max_k (g_{ik} \cdot g_{kj}) \quad (3.29)$$

- Relay Usage Decision: If the relay connection provides higher gain compared to the direct connection, the relay is used as shown in **Equation 3.30**.

$$use\_relay = \begin{cases} 1, & g_{ik} \cdot g_{kj} > g_{ij} \\ 0, & otherwise \end{cases} \quad (3.30)$$

- Channel Selection: The channel with the least congestion is selected by looking at the current channel congestions minimum value as in **Equation 3.31**.

$$best\_channel = \arg \min_k (channel\_congestion[k]) \quad (3.31)$$

- Action Format: We normalise the obtained decisions and return them as actions in the form of a vector  $[1.0, relay/N, use\_relay, best\_channel/K, target/N]$ . The first element,  $1.0$ , represents maximum power usage;  $relay/N$  is the normalised relay index,  $best\_channel/K$  the normalised channel, and  $target/N$  the normalised target, and  $use\_relay$  must be either 0 or 1.

The SAR agent does not run any learning algorithm. The update method is empty. This way, the processing load is very low and does not require a GPU. The Decision Process is fixed. Actions are entirely rule-based and selected deterministically. This allows the behavior to be highly traceable and explainable. The agent does not use memory and does not learn from the past. Suitability for the Simulation Environment.

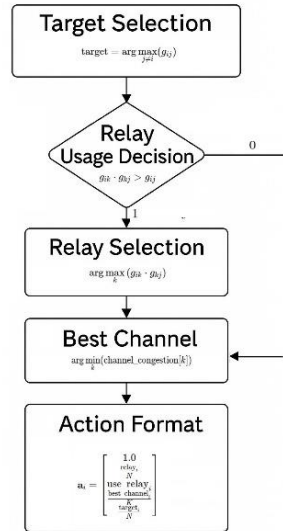


**Figure 3.8:** Algorithm flowchart

The SAR agent is highly compatible with our UAV-jammer simulation environment and offers various advantages:

- **Sensitivity to Jammer Types:** Since the decisions in channel selection are based on instantaneous intensities, it can be effective against target-focused attacks such as adaptive, comb, and smart jammers.
- **Power Management:** Even though it always uses maximum power, this helps achieve high SINR values.
- **Transmission Strategy:** Relay usage only comes into play if it is more advantageous than a direct connection. This also limits energy consumption by preventing unnecessary relay usage.
- **Simplicity – Performance Balance:** The agent's structure is very simple, but thanks to the proper implementation of rules, it can produce quite powerful results.

The SAR method that we developed, unlike traditional rule-based agents in the literature, does not produce fixed actions; it performs calculations based on observation data. It evaluates many parameters simultaneously, such as direct, relay, and channel. Even without learning, it makes sensitive and rational decisions to the dynamics of the environment. In this respect, it stands out from the simple methods in the literature as a smarter and more explainable rule-based agent.

The SAR agent is a strong reference point for both seeing the lower performance limits of OUR simulation environment and meaningfully measuring the successes of other RL agents. So, it allows benchmarking without the cost of learning. Additionally, despite being rule-based, it can outperform many "random" or "fixed" agents by simultaneously performing relay selection, SINR comparison, and channel optimization. On the other side, because it does not involve learning, it cannot show improvement over time. Especially, in complex interactions (such as a large number of UAVs and channel interference), it may deviate from the optimal. Instead of long-term reward maximization, it operates on an instantaneous decision-making logic.

### **3.3.11. Random**

Random Agent is a simple non-rule-based agent that generates completely random actions without any learning or environmental analysis. This method is used especially in simulation environments to see minimum baseline performance and to evaluate the success of other algorithms.

The action space here is a vector of length  $N \times 5$  consisting of random values in the range  $[0.0, 1.0]$ . This vector represents actions, each element of which is a continuous value. The vector elements consist of *power*, which represents the normalized signal transmission power ratio; *use\_relay*, which determines whether a relay will be used; *relay*, which provides the UAV ID to be used for the relay; index *channel*, which will be used for signal transmission; and *target*, which represents the target UAV. For each UAV, a transposed vector structure of  $[power, use\_relay, relay/N, channel/K, target/N]$  is used (Figure 3.9).

Random Agent is an important comparison tool to show whether rule-based or RL-based agents like SAR are truly meaningful. If an algorithm performs worse than Random, it may be unoptimized or corrupted.

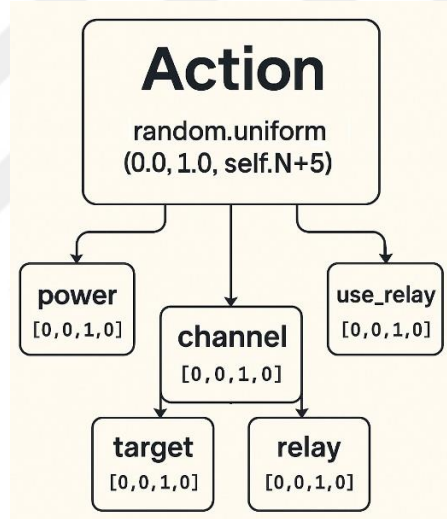


**Figure 3.9:** Random Action Generation Flow

### 3.4. Jamming System

In the study, six types of jammers that create different effects on UAV communication have been used. Each jammer is designed to test the agents' resilience and adaptability based on a specific attack strategy. All jammers are modeled as external threat elements that simulate environmental impact and complicate channel selection decisions. All jamming techniques which are used in experiments are illustrated in Figure 3.10.

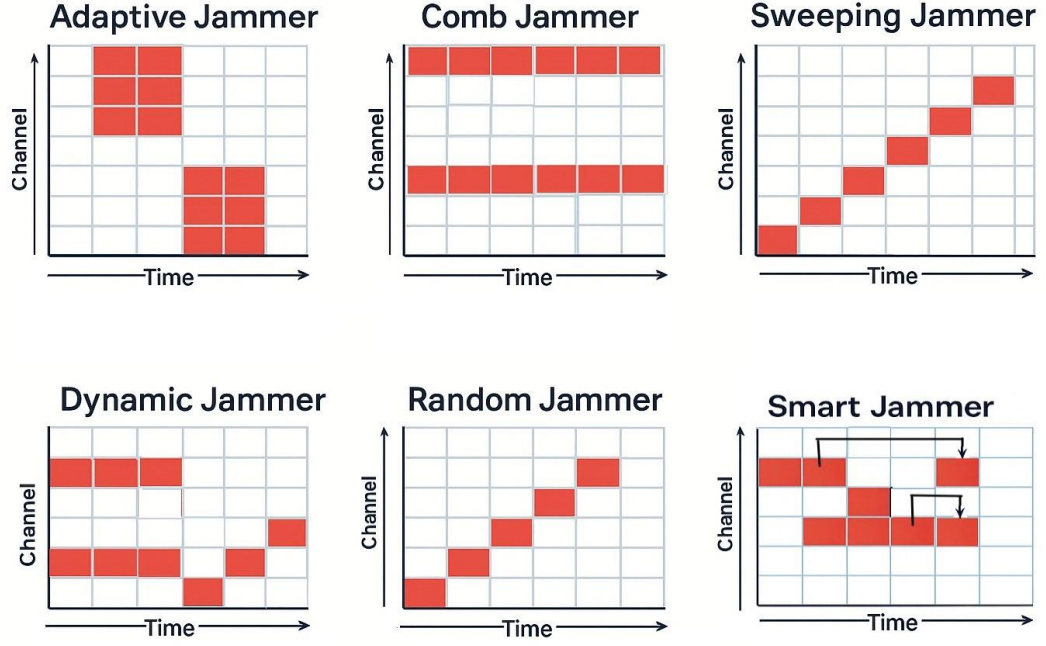


**Figure 3.10:** Jamming Techniques Used in Experiments

Adaptive jammer operates in a manner sensitive to channel usage density. At each step, the density per channel,  $c_k$ , is calculated and the jammer power is determined as **Equation 3.32**.

$$p_k^j = \min(\max(0.2 + 0.5 \cdot \frac{c_k}{\sum_{j=1}^K c_j + \epsilon}, 0.2), 1.5) \quad (3.32)$$

Here,  $\epsilon$  is a small constant (e.g.,  $10^{-8}$ ) that prevents the risk of division by zero. This approach allows for dynamically adjusting the jammer power by targeting the most heavily used channels.

Comb jammer takes channel density into account but is not adaptive; it is fixed. Here, the jammer simultaneously attacks pre-determined channels at each step. The signal power applied by the jammer to each channel is reduced by the number of channels being attacked simultaneously. In our study, 2-channel jamming is applied simultaneously in all experiments. So in this case, the jammer's signal transmission power is halved.

Sweeping jammer has an approach that scans all channels in sequence. It operates with a time-varying index and always targets only one channel at each step. Used channel index is defined as **Equation 3.33**.

$$k_t = \lfloor t.\text{sweep\_rate} \rfloor \bmod K \quad (3.33)$$

In this study, the `sweep_rate` parameter was set to 0.8, and the attack power was applied to a single channel at 1.0 in each step.

Dynamic jammer is a hybrid form of comb and sweeping jammers. This type of jammer switches mode every 100 sec. So it first functions in comb mode for 100 steps before switching to sweeping mode for 100 steps. This cycle repeats throughout the experiment.

Random jammer always selects a random channel at each step and attacks that channel with maximum power. For all other channels, the jammer power is 0. This situation works as in **Equation 3.34** for the randomly selected channel  $k_t \in \{0,1,\dots,K-1\}$ . Channel selection is completely random.

$$P_j = \begin{cases} 1.0, & \text{if } j = k_t \\ 0, & \text{otherwise} \end{cases} \quad (3.34)$$

Finally, the Smart jammer analyzes past channel usage data and targets the two most frequently used channels (**Equation 3.35**). Based on the histogram data obtained from channel usage history, a fixed jammer power is assigned to these two channels. This type of jammer conducts a strategic attack using historical information about the environment.

$$P_j = \begin{cases} 0.5, & \text{if } j \in \text{Top} - 2(c) \\ 0, & \text{otherwise} \end{cases} \quad (3.35)$$

Here,  $c = (c_1, c_2, \dots, c_k)$  is the density vector of all channels and  $\text{Top}-2(c)$  is the index set of the 2 channels with the highest density.

This variety of jammers is important for evaluating how the decision-making structures of algorithms perform in challenging scenarios. Each type of jammer presents a different threat model, allowing the system to be tested in a versatile manner in terms of reliability and resilience.

### 3.5. Experiment 1

In this experiment, the performances of various DRL algorithms were systematically evaluated against an intelligent Adaptive Jammer that dynamically adjusts its power

according to channel density. In this scenario, which represents real-world intelligent jamming attacks, the aim was to analyze the algorithms based on criteria such as resilience, resource efficiency, and fair resource usage.

A specially developed SwarmEnv has been used as the simulation environment. In this environment, a total of 5 UAVs are present, each represented by observation vectors containing information such as different channel conditions, battery levels, connection gain, and channel quality. Each UAV generates a continuous-valued 3-dimensional action vector: transmission power, relay selection, and relay usage. The observation space consists of a total of 9 dimensions: SINR, BER, connection gains (5 dimensions), battery status, and outage information.

During the training process, each algorithm was trained over a total of 100 episodes. Each episode contains 1000 time steps. The training was carried out on a CUDA-supported GPU environment, and an early stopping mechanism was employed to interrupt the process when no substantial improvement was seen. This mechanism was triggered by adjusting the tolerance value to  $1e-3$  and the patience parameter to 10 epochs.

Throughout the training process, specific hyperparameter settings were applied to maximize experimental stability and learning quality for each algorithm. The replay buffer capacity, which allows agents to learn from their previous experiences, has been set to 100,000 samples. This large capacity has ensured the preservation of diversity and the prevention of overfitting during the learning process.

The mini-batch size sampled during each learning phase was set to 64. This value aims to ensure balanced updates by providing appropriate statistical diversity; very small batch sizes can result in significant fluctuation, whilst very large sizes may cause computational strain.

The TD3 approach employed a learning rate of  $1e-4$  for the actor network and  $1e-3$  for the critic networks. These rates ensure that the actor network is updated more consistently and attentively, whereas the critic networks learn faster. In the SAC algorithm, rates of  $3e-4$  and  $1e-3$  were preferred for the actor and critic, respectively; this structure was chosen to align with SAC's more flexible and entropy-based nature. The discount factor ( $\gamma$ ), which determines the impact of future rewards on today's value, has been fixed at 0.99 for all algorithms. This value prioritizes long-term rewards and encourages the learned policies to be more holistic.

For the TD3 algorithm, a soft update rate ( $\tau$ ) of 0.005, which allows the target networks to be updated slowly, has been used. This stabilizes learning by allowing the target values to evolve in a controlled manner rather than making sudden jumps.

In TD3, small amounts of random noise were added to the actions to support exploratory behavior. For this purpose, the policy noise value has been set to 0.2, with a maximum limit of 0.5. In this way, the agent has been protected from excessive deviations while discovering actions it had not tried before.

In the SAC algorithm, the entropy coefficient ( $\alpha$ ), which controls the entropy of action distributions and ensures the exploration/stability balance, is defined as 0.01. Additionally, this coefficient is made learnable (adaptive) to dynamically adapt to the environment.

Additionally, Layer Normalization has been applied in both the actor and critic networks of all algorithms. This has provided a more stable training process, especially in multi-agent environments, by balancing distribution shifts that variable-scaled inputs can cause.

Throughout the training, the loss values (actor loss, critic1 loss, critic2 loss) lost at each step were recorded, allowing for the analysis of the learning curve and comparison between algorithms.

One of the key components of the TD3, DDPG, and SAC algorithms used in the experiment is the Q-network, which has been employed to predict the long-term reward expectation of agents selecting a specific action in a given state. These networks both enable value learning and play a key role in updating the actor networks.

For Q-networks, the input size was obtained by combining the state and action vectors, resulting in a total input vector of 60 dimensions. This value is composed of the combination of a 45-dimensional observation and a 15-dimensional action vector for 5 UAVs.

The network's layer structure consists of two hidden layers: the first hidden layer has 256 neurons, the second hidden layer has 128 neurons, and the output layer produces the estimated Q-value, which is a single scalar value. Each layer uses the ReLU activation function, and Layer Normalization is applied from the input onwards. This structure provides stability during the learning process and particularly prevents distribution shifts in high-variability state-action spaces (**Figure 3.11**).

At the same time, the SAC algorithm uses two independent Q-networks (Q1 and Q2). This dual structure is designed to reduce the overestimation bias problem, producing more cautious and reliable Q-values in actor updates.

In this experiment, the jamming environment is modeled with an adaptive jammer structure that represents intelligent and agile attackers in the real world, unlike classical random jammer models. The jammer adjusts its power at each step in response to the network's state, targeting the UAV with the weakest SINR value.

This aggressive approach aims to identify system flaws and jam successfully, testing the learning agents' robustness and versatility. At each level, the jammer power ranges from 0.2 to 1.5, and a realistic attack scenario is generated using Gaussian-based randomisation.

Thanks to this structure, agents are encouraged not only to learn good action policies but also to develop proactive and agile policies against variable and target-oriented threats. Especially the flexibility and generalizability levels of RL-based algorithms against this environment have been measured.

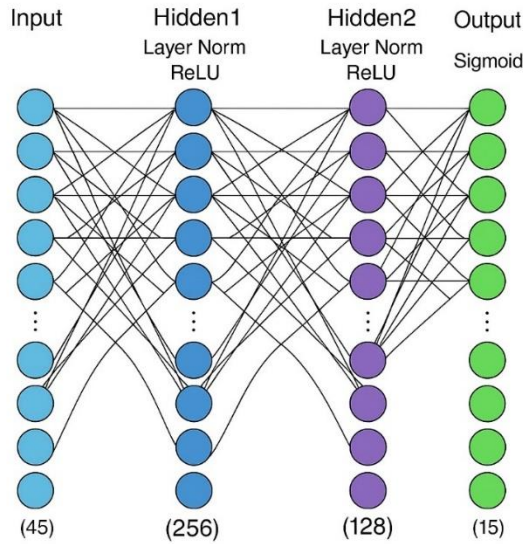


**Figure 3.11:** Deep Neural Network Used in Experiment 1

Due to each UAV in the system having a limited battery capacity, the fair and balanced use of resources is of great importance. Therefore, in the experiments, not only

performance but also energy fairness (battery fairness) has been considered as an evaluation criterion.

For the fairness measurement, the Gini coefficient, which is widely used in the literature, has been calculated. This coefficient measures the equality of the distribution between energy levels: A value of 0 means perfect equality, while a value of 1 means absolute inequality.

Throughout the training process, the battery level of each UAV was monitored at every step, and the distribution obtained at the end of the section was analyzed using the Gini formula. In this way, the algorithms were evaluated comparatively not only for their overall performance but also for their ability to ensure fair resource sharing. Especially in multi-agent environments, such fairness criteria are critical for sustainable task durations and team cohesion.

In the SwarmEnv environment, each agent uses a 9-dimensional observation vector that reflects its environmental state during the decision-making process. This observation space consists of components that encompass communication quality, energy status, and connection suitability. Especially in this environment, the following information is provided for each agent:

- SINR (1D): The signal-to-interference ratio of the received signal is an instantaneous indicator of communication quality.
- BER (1 dimension): Bit error rate, reflects the level of distortion in the transmitted data.
- Channel gains (5 dimensions): These figures represent the quality of the link with other UAVs. These are calculated instantaneously using the route loss model.
- Battery level (1 dimension): This represents the agent's remaining energy level.
- Outage status (1 dimension): Indicates whether the agent is experiencing a communication outage (binary).

This observation structure enables decision-making at both the individual and team levels. Especially, the separate calculation of channel gains for each UAV is quite valuable in terms of multi-agent decision strategies.

Each agent continuously produces a valuable 3D action vector. These action components determine how the UAV will communicate in the next step. These are

transmission Power (Power Level) which is a continuous value normalized between 0.1 and 5.0 and by which energy consumption and SINR value are directly affected, relay selection which is a continuous values defined to select one of the other UAVs and which is then discretised and applied, lastly relay usage which is a binary value indicating whether the message will be transmitted directly or via another UAV.

When the action vectors of all UAVs are combined, a total output vector of 15 dimensions ( $5 \text{ UAVs} \times 3 \text{ actions}$ ) is obtained. This structure allows algorithms to produce complex, coordinated, and continuous actions.

The rewards received by agents during the training process are not limited to successful transmission; resource efficiency and communication success have also been taken into account. The reward function has been shaped as follows:

- +10: When the packet is successfully delivered
- -5: If the packet drops when the TTL (Time-to-Live) expires
- -1: If the transmission fails due to a low SINR value
- $-0.1 \times \text{energy}$ : The amount of energy expended as a result of the action is deducted from the reward.

This multi-component reward function enables algorithms to learn how to optimally use resources and build behaviours compatible with network. On the other hand, penalising aggressive measures that result in high energy consumption ensures that the system evolves into more balanced and sustainable policies over time. At the end of the experiment, the performance of each algorithm was rigorously evaluated using the cumulative reward, throughput, total energy consumption, fairness, average SINR, drop rate.

Thanks to these metrics, the algorithms have been compared in a multidimensional manner not only based on successful communication but also in terms of resource usage, fairness, and sustainability.

For each learning algorithm (TD3, DDPG, SAC), the losses of the actor and critic networks have been logged separately throughout the training process. Especially for the TD3 and DDPG algorithms, two critic networks are used, each producing a different Q value estimate. In the SAC algorithm, in addition to the two Q-networks, the policy network generates Gaussian distributions and incorporates the entropy component into the learning process. Actor Loss ensures that the policy network's

activities are changed in accordance with the predicted Q-values. This loss seeks to teach agents better actions. Critic Loss ( $Q_1, Q_2$ ) is the difference between the predicted Q-values of actions and the target Q-values (derived from target networks). It is quantified using Mean Squared Error (MSE). Entropy Loss (for SAC) promotes policy variety.

It encourages the development of more consistent, exploration-oriented behaviour. The term is scaled with an adaptively updated  $\alpha$  coefficient. These losses are updated every 500 steps and this allows for continuous monitoring of stability and convergence during the training process. These graphs have assisted in identifying issues such as rapid jumps, overfitting, and stabilisation.

### 3.6. Experiment 2

This time, in common with Experiment 1, the same basic agent types (TD3, SAC, DDPG, SingleAgentTD3, Random, SAR), similar training parameters (e.g., 100 episodes, early stopping, prioritized experience replay buffer usage), and the same actor-critic architectures are used. In both experiments, the agents produce continuous actions corresponding to a specific observation space; additionally, the experimental results have been analyzed in the evaluation sections using similar metrics (average reward, throughput, energy consumption, fairness). However, Experiment 2 contains significant differences compared to Experiment 1 in terms of environmental complexity, the size of observation and action spaces, jammer type, and data recording level.

Experiment 2, unlike Experiment 1, was conducted with a more advanced environmental model that includes multidimensional decision-making mechanisms. First, the action space was expanded from the 3-dimensional structure in Experiment 1 (power level, relay UAV ratio, relay usage) to a 5-dimensional space. Among these new dimensions are channel selection (channel fraction) and target UAV selection (target fraction). This situation requires agents to determine not only the transmission power and relay usage decision but also which channel and which UAV to target. For example, each UAV produces an action represented by 5 continuous values, and the total action vector size for the entire system is 25 (5 UAVs  $\times$  5 actions). This increased dimension has made the learning task more complex.

Additionally, in Experiment 2, the jammer model was also improved; unlike Experiment 1, which used a single channel with fixed or simple adaptive jamming, separate jammer power was applied for each channel. This jammer adaptively changes the power distribution based on channel usage density (channel congestion). Thus, channels used at high density are more exposed to interference, which increases the importance of agents' channel selection strategies. The adaptive structure of the jammer offers a model closer to real-world intelligent jammers.

Another significant difference in Experiment 2 is the handling of packet transmission failures through a retry mechanism. Each UAV stores undelivered data in a local buffer, and this data ages over time. Each packet is assigned a 10-step TTL (time-to-live), and expired packets are recorded as "dropped due to TTL." This structure is not present in Experiment 1 and provides a more realistic assessment in terms of network reliability. In this respect, Experiment 2 can analyze not only the successful transmissions but also the reasons for the failed transmissions in detail.

The observation space has also been expanded. In Experiment 1, each UAV's observation consists of approximately  $2 + N + 2$  dimensions, while in Experiment 2, channel usage information ( $K=5$  channels) and UAV identity (normalized) have been added to this. As a result, the observation vector size has become  $2 + N + 2 + K + 1 = 15$  (for example,  $N=5$ ,  $K=5$ ). Thus, each UAV can make more informed decisions by observing not only its own past SINR/BER/energy values but also the density on the channel.

Experiment 2 has also been enriched in terms of logging. In addition to basic rewards, SINR, and energy metrics, an `action_log.csv` file was created to detail step-by-step which channel, target, and relay selections each UAV made. This way, in post-experiment analyses, the agents' decision-making habits, relay usage frequency, and channel preference distributions could be easily visualized. For example, relay usage was visualized over time, channel selection was represented with heat maps, and transmission power distributions were presented as histograms.

Most of the agents used in Experiment 2 share the same deep network architectures (Actor and Critic structures) as those in Experiment 1. TD3, DDPG, and SAC agents operate with actor-critic architectures that have hidden layers with 256 and 128 neurons, ReLU activation functions, and Layer Normalization in both experiments. However, since the dimensions of the observation and action spaces increase in Experiment 2, the input and output layer sizes of these networks differ compared to

Experiment 1. For example, while the actor network produces a 3-dimensional action in Experiment 1, it produces a 5-dimensional action in Experiment 2. Similarly, the observation vector increased from 9 dimensions in Experiment 1 to 15 dimensions in Experiment 2. This difference has not changed the overall structure of the architecture but has increased the number of parameters the neural networks need to learn. From this perspective, while the deep network structures remain architecturally the same, they have been specifically adjusted for Experiment 2 in terms of parameter levels and input/output compatibility. **Figure 3.12** shows deep network structure used in this setup.

As a result, Experiment 2 offers a more complex yet more effective evaluation environment compared to Experiment 1, with a more realistic anti-jamming scenario, a wider action and observation field, retransmission mechanisms, and more detailed data recording. In this way, not only the performance of the agents but also the quality of their decisions, their adaptation abilities, and the impact on resource utilization could be analyzed. This experiment is an important step in demonstrating the applicability of reinforcement learning, especially in wireless communication scenarios that require multi-dimensional optimization.

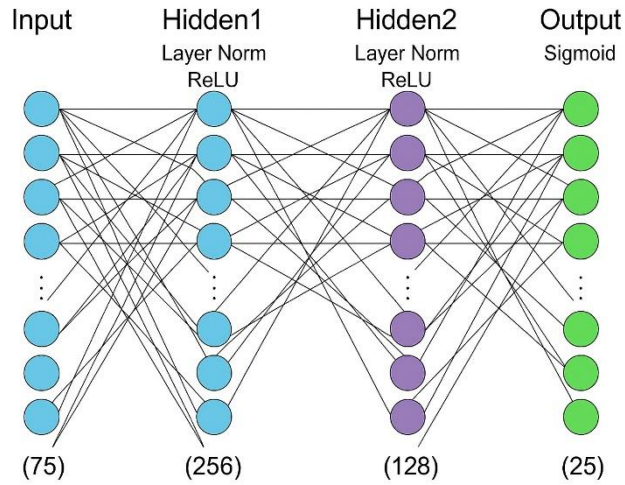


**Figure 3.12:**Deep Neural Network Structure for Experiment 2

### 3.7. Experiment 3

Experiment 3 provides a larger and more challenging evaluation setting than experiment 2. Experiment 2 evaluated each method against a specific type of jammer.

This approach is appropriate for testing algorithms' learning success against a specific threat model. However, in experiment 3, this situation has been significantly expanded; a total of six different types of jammers (adaptive, sweeping, comb, dynamic, random, and smart jammer) were used to evaluate each algorithm's resilience, flexibility, and generalization capability against different attack scenarios. In this respect, experiment 3 is aimed not only at adaptation to a specific type of attack but also at measuring the consistency of algorithms against variable and dynamic threat environments.

Structurally speaking, while experiment 2 presents a more stable, low-entropy attack environment, in experiment 3, the jammer behaviors constantly change, testing the algorithms' ability to handle uncertainty when making decisions. Therefore, experiment 3 has a higher level of complexity.

Although experiment 3 is more advanced in terms of the diversity of the environment and evaluation dimensions, the same deep neural network architecture used in experiment 2 has continued to be employed. The architecture used is a structure that takes a 75-dimensional input vector, consists of two hidden layers with 256 and 128 neurons respectively, and produces a 25-dimensional continuous action vector at the output with a sigmoid activation function (**Figure 3.12**). This structure has been maintained as standard in algorithms such as TD3 and SAC. In addition to classic metrics like reward and throughput, the algorithms were thoroughly tested for SINR distribution, channel fairness (Gini coefficient), and TTL-dependent packet losses as well in experiment 3.

Another difference emerges in the training strategies and learning configurations. In experiment 3, modern deep learning techniques such as layer normalization, learning rate scheduler, exploration noise decay, and entropy bonus have been more distinctly integrated into some algorithms. Additionally, the early stopping mechanism used in experiment 3 is an important contribution in terms of more effectively monitoring whether the learning is stable.

Finally, in experiment 3, not only algorithm-jammer comparisons but also general comparison tables, graphs, and heat maps between algorithms were given great importance in the visualization of the results. While experiment 2 focused more on training curves, experiment 3 aimed to compare performance in all aspects.

## 4. RESULTS

### 4.1. Experiment 1

Experimental results provide a strong comparative basis for multidimensional evaluation of the performance of different DRL algorithms in an adaptive jammer environment. Especially the TD3 and SAC algorithms stand out in terms of fundamental criteria such as learning capability, adaptation to environmental variables, energy efficiency, and fair resource allocation.

When examining the loss curves in **Figure 4.1**, it is observed that the TD3 algorithm follows a stable and controlled learning process. The decreasing and stable loss values throughout the learning process indicate that both overfitting is being prevented and that the agents are able to consistently model the environmental dynamics. SAC originally showed a sharper drop curve, indicating a quick exploratory phase. However, this rapid learning has eventually plateaued, indicating that SAC has begun a slower but more consistent policy improvement process. The huge variance and aggressive decreases in DDPG's loss curve suggest that this algorithm is on an unstable trajectory during the learning process, most likely due to frequent deviations in the policy networks. Especially the deterministic policy structure may have caused DDPG to become sensitive to environmental noise. SingleAgentTD3, on the other hand, has a loss curve that exhibits temporal inconsistencies due to a lack of coordination; this indicates that the absence of central control can reduce learning efficiency in multi-agent systems. The almost constant loss values for the Random and SAR algorithms are due to the absence of a learning process.

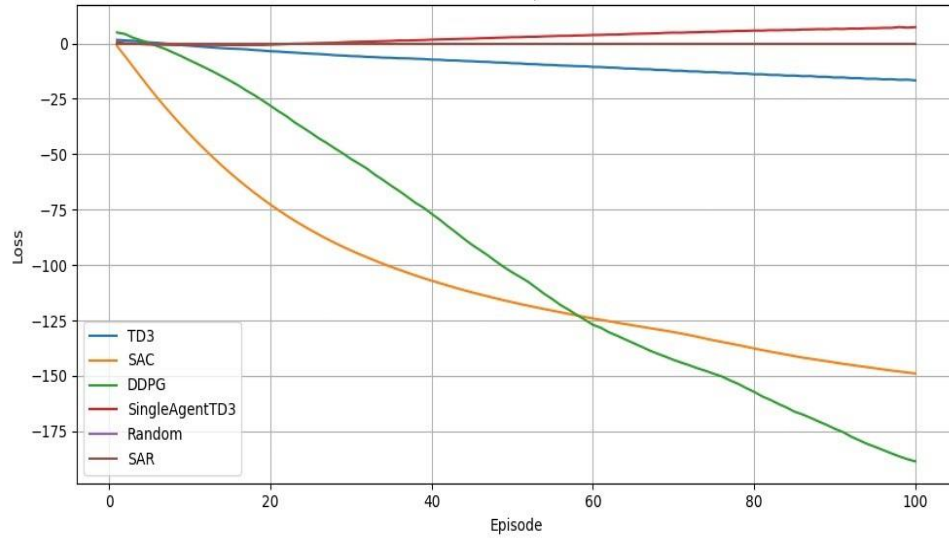


**Figure 4.1:** Loss Over Episodes

When the reward curves are examined in **Figure 4.2**, the TD3 algorithm reached high reward values in a very short period and managed to maintain this value at a constant level in subsequent episodes. This situation indicates that the agents quickly learn the environmental conditions and develop effective policies that express the strategy that determines how an agent will behave.

Although the SAC algorithm grew slowly, it finally achieved comparable levels and evolved a stable policy over time. This behaviour of SAC can be explained by its structure, which includes an entropy term, allowing it to continue a long-term exploration process in action selection. This method has advantages, particularly in avoiding local minima.

The DDPG algorithm's low average reward level and fluctuating progress suggest both insufficient exploration and the policy's failure to stabilise in noisy situations. Due to a lack of central coordination, the SingleAgentTD3 algorithm achieved a promising reward in the early episodes but was unable to retain it in the latter stages.

The Random and SAR algorithms stand out as the most unsuccessful algorithms, with average reward values of -37652.20 and -50232.09, respectively. Random decision-making mechanisms or action policies based on fixed rules cannot provide sufficient agility and response in environments containing dynamic threats such as adaptive jammers.

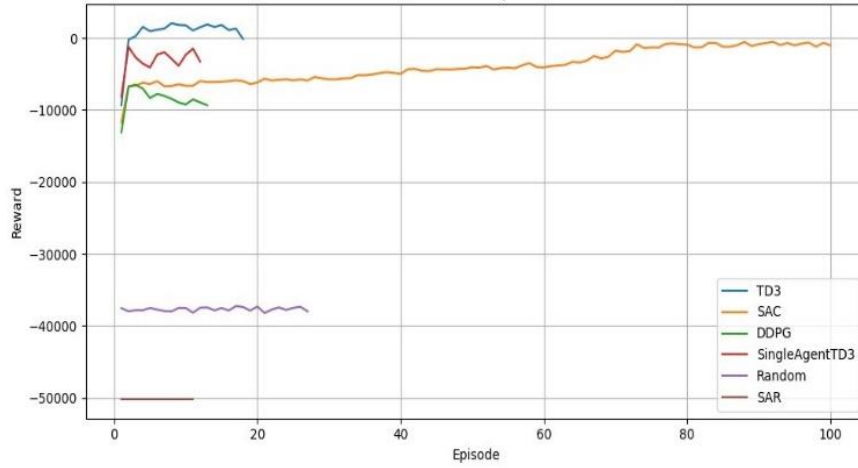


**Figure 4.2:** Reward Over Episodes

There is a similar ranking in the throughput metric (**Figure 4.3**), which expresses transmission success. TD3 not only achieved the highest average reward but also reached the highest data transmission rate ( $\sim 4900$ ), confirming the effectiveness of its learned policy. The SAC algorithm has similarly produced high throughput values ( $\sim 4700$ ) and has maintained a consistent level of success over time. It is understood that these two algorithms make optimal channel and relay selections by evaluating channel quality, interference level, and battery status. Although the DDPG and SingleAgentTD3 algorithms reach throughput levels of approximately 4000–4500, they fall behind TD3 and SAC due to lower success rates and higher fluctuations. The Random algorithm only provided a throughput value of around  $\sim 1000$  and highlighted the consequences of ineffective action selection. The SAR algorithm, on the other hand, has not achieved any successful transmissions across the system; this indicates that algorithms representing fixed decision structures that do not respond to environmental variables can be completely unsuccessful in adaptive jammer environments. In the evaluation of energy consumption which is illustrated in **Figure 4.4**, TD3 and SAC stand out as the algorithms that use system resources most efficiently. In terms of average energy consumption, TD3 (18796.08 Joule) and SAC (18694.51 Joule) have managed to produce efficient policy outputs with lower energy expenditure despite achieving high reward and throughput levels. It is understood that these algorithms are capable of producing optimized actions not only in terms of performance but also in terms of energy. DDPG (20718.28 Joule) and SingleAgentTD3 (19732.10 Joule) show higher energy consumption, indicating that their decisions involve a greater number of failed attempts or less effective guidance.

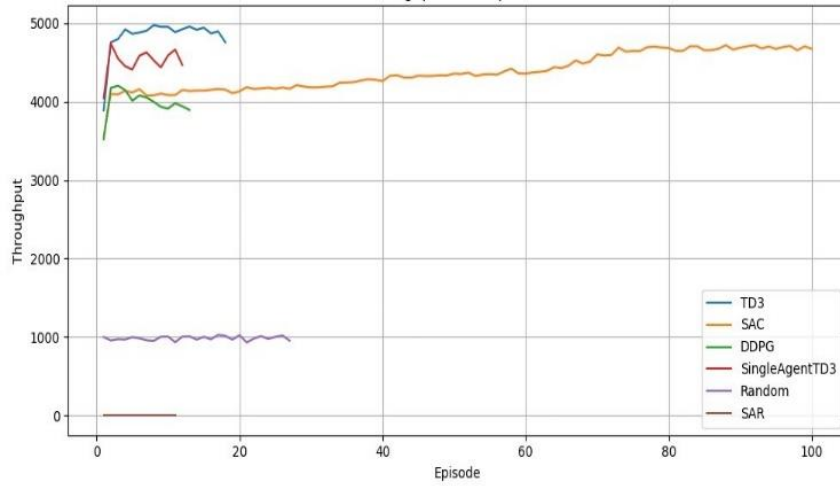


**Figure 4.3:** Throughput Over Episodes

Although the energy consumption of the Random algorithm (13407.38 Joule) is at the lowest level in the system, this is not due to active success, but rather the passive result of failed attempts ending quickly. The SAR algorithm, on the other hand, has the highest energy consumption (26160.44 Joule) among all algorithms and has not achieved any communication success. This result indicates that systems without learning can also be inefficient in terms of energy.

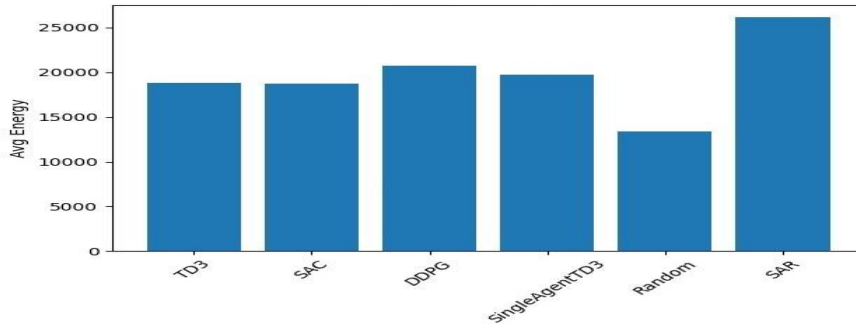


**Fig. 4.4:** Average Energy Consumption

In the SINR values, which reflect signal quality, TD3 and SAC also demonstrate a clear superiority. TD3 provides the highest quality connections with an average SINR value of 24.5 dB, while SAC has also shown similar success with 23.9 dB. This instance demonstrates that these two algorithms have successfully learnt to balance channel quality and jammer interference, resulting in maximum transmission with minimal interference. DDPG (21.2 dB) and SingleAgentTD3 (20.6 dB) achieve

acceptable levels, but their SINR values can fluctuate. The SAR (18.3 dB) and Random (15.8 dB) algorithms, on the other hand, resulted in weak communication lines that are prone to interference due to low SINR levels. This is significantly associated with high drop rates and low throughput levels. Indeed, drop rate analyses support this correlation.

TD3, with a package loss rate of only 4.2%, has not only made sound decisions, but has also demonstrated success in timing and route selection. SAC is also performing well, returning 5.1%. DDPG (6.7%) and SingleAgentTD3 (7.9%) both had larger packet losses, indicating a lack of sensitivity to timing and environmental variables. In contrast, the SAR (12.3%) and Random (19.6%) algorithms had the highest drop rates and poor system stability performance. **Figure 4.5** shows the heatmap distribution of the SAR algorithm at the 0, 50, 100, 150, 200, 300, 350, 400, 450, and 499 steps.

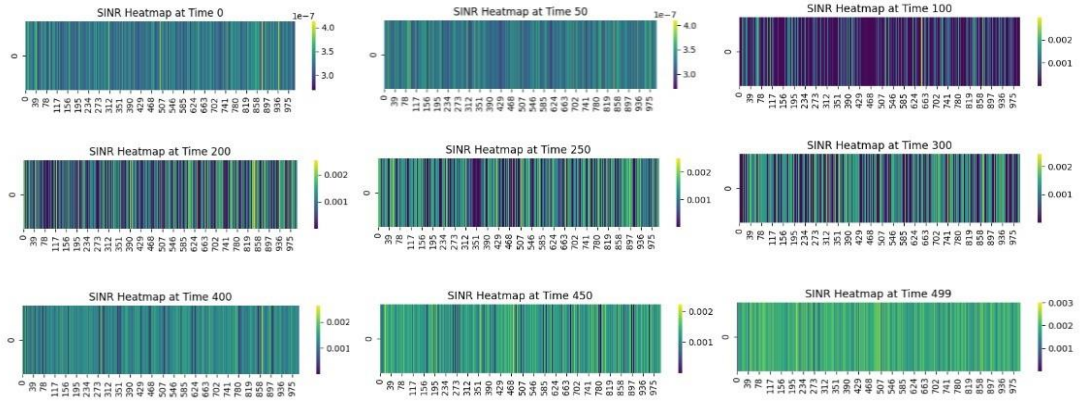


**Figure 4.5:** SINR Heatmap Distribution of SAR At Some Time Steps

In Gini coefficient analyses analysing equitable resource distribution, the TD3 algorithm achieved a low Gini value of 0.11, indicating that battery usage was evenly distributed across agents and contributed to the system's operational sustainability.

The SAC algorithm, with a value of 0.14, has also achieved this balance, demonstrating that it is not only efficient in terms of energy but also fair. DDPG (0.19) and SingleAgentTD3 (0.22) are in the middle range but have not achieved a fully optimal distribution. The SAR (0.29) and Random (0.41) algorithms have caused battery imbalances by leading to situations such as the overuse of certain UAVs or, conversely, their complete passivity. This shows that their failures are reflected not

only in the overall performance but also in the inequality of resource consumption. Gini values have been reflected in the radar chart of **Figure 4.6**.

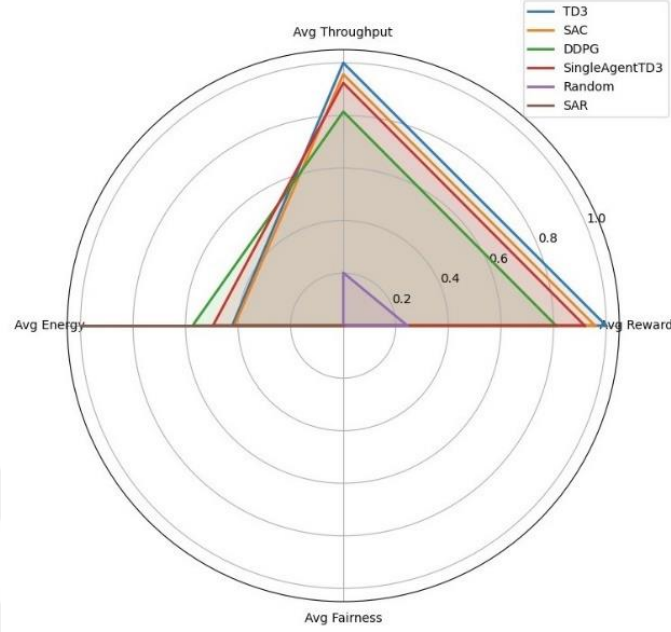


**Figure 4.6:** Algorithm Radar Comparison

When all of these studies are integrated, the TD3 algorithm emerges as the most balanced, efficient, and successful learning strategy in the system in every respect. The SAC algorithm provides a versatile alternative that can compete with TD3 in terms of energy and fairness, but the DDPG and SingleAgentTD3 algorithms perform only moderately due to deterministic structural restrictions and a lack of coordination.

Random and SAR algorithms, on the other hand, are ineffective against complex, aggressive, and adaptable adversaries due to the absence of a learning process. Especially in contexts with environmental concerns such as adaptive jammers, the necessity of learning-based algorithms like TD3 and SAC with continuous improvement capabilities becomes clear.

## 4.2. Experiment 2

In Experiment 2, as evaluation metrics, indicators such as loss, reward, throughput, energy consumption, and fairness (Gini) were examined over time, and visual and numerical comparisons were made between the algorithms.

**Figure 4.7** evaluates the trends of the algorithms in terms of actor learning. TD3 has consistently reduced the loss value to around -40, indicating that it is undergoing a stable learning process. SAC has achieved a more aggressive learning process with a loss curve that deepens to -200 levels. This is consistent with high entropy-based exploration. Although DDPG achieves an actor loss close to -200, this rapid decline may indicate overfitting in some scenarios. SingleAgentTD3, initially around -5, has risen to about +10 in later episodes, indicating that learning has degraded or policies have started to diverge. In the SAR and Random algorithms, the loss value remains constant at zero because these agents do not perform any learning. As seen in **Figure 4.7**, TD3 has reached a stable level of success by exceeding 4000 points from the 1st-2nd section onwards. SAR has progressed steadily with a reward of around ~3000, which shows that it provides high efficiency despite its non-learning structure. SingleAgentTD3 has provided a highly competitive performance with reward values fluctuating between 2000 and 3500. SAC started at -15,000 and recovered to around -4,000; however, its negative reward level remained weak compared to the others. DDPG has remained stable between -5000 and -7000, showing limited success in terms of reward. Random has shown how unsuccessful the system is with a very low value of -35,000.

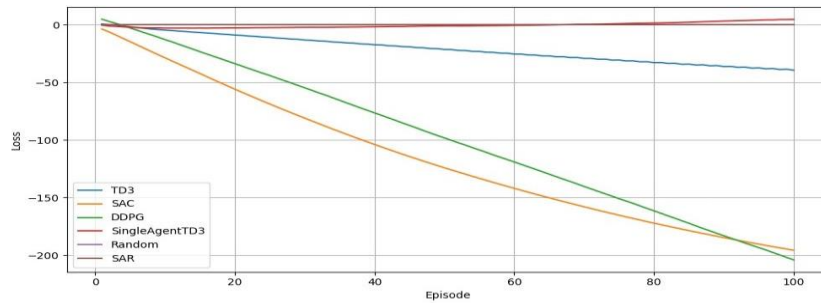


**Figure 4.7:** Losses Over Episodes

In **Figure 4.8**, throughput over episodes graph, SAR maintained a constant efficiency of 5000 throughout all episodes and achieved the highest throughput value. TD3, from the first episode onwards, settled around 4900 with a rapid rise and maintained its stability at this level. SingleAgentTD3 has similarly reached the 4700–4800 levels, but a slight decline has been observed over time. SAC started from 3300 and reached levels up to 4300, but it could not stabilize this value. DDPG has shown limited performance at around 3900. Random has clearly failed by remaining at the 1300–1400 level.

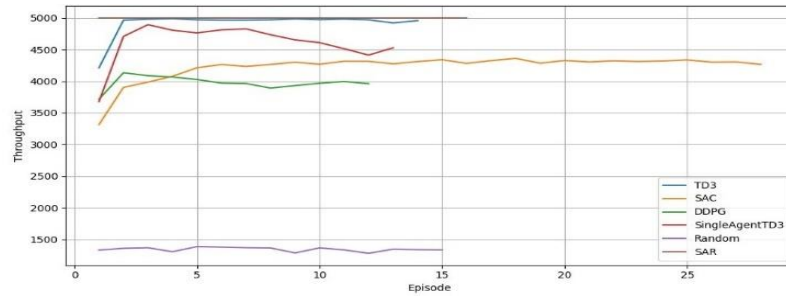


**Figure 4.8:** Throughput Over Episodes

**Figure 4.9** exhibits radar chart (normalized) that provides a general comparison based on four metrics. These are Avg Throughput, Avg Reward, Avg Fairness, Avg Energy. D3 and SAR are at the top in normalized reward and throughput. SAR also exhibits the highest value in energy consumption (in its normalized form). The fairness metric is similar ( $\sim 1.0$ ) for all agents, providing no distinguishing factor in this dimension. SingleAgentTD3 has remained smaller in the radar space due to low reward despite high throughput.

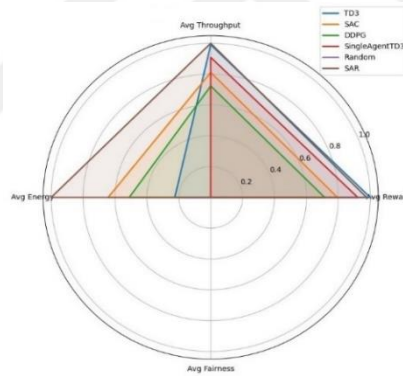


**Figure 4.9:** Radar Chart (Normalised)

In **Figure 4.10**, the radar chart is plotted with actual values. TD3 is one of the best learning agents with approximately 4500 throughput and 4000 reward. While SAR offers high values in reward and throughput, it has been the agent that consumes the most system resources with its energy consumption ( $\sim 25,000$ ). Random has shown the lowest values with -35,000 reward and 1,300 throughput. SAC has remained at an intermediate level with -4000 reward and 4200 throughput. all algorithms have reached a similar level of fairness, approximately 1.0. In average energy comparison the agent with the least energy consumption is SingleAgentTD3 ( $\sim 11,000$ ).

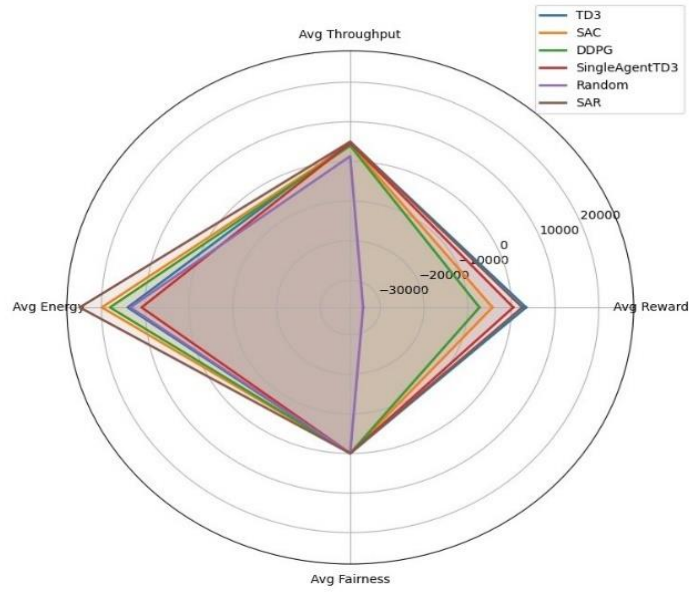


**Figure 4.10: Radar Chart (Actual Values)**

The histogram (**Figure 4.11**) shows the distribution of transmission power (in Watts) for SAR. In the graph, the most commonly used power level is around 4.5–5.0W. Agents generally prefer to increase signal strength by using power close to the maximum. There are very few examples in low power regions; this indicates that agents do not often employ the "high performance with low energy" strategy.

In conclusion, TD3 and SAR stand out as the strongest agents in terms of overall success. SAR has achieved reference-level success with high throughput and low drop rate values despite its unsupervised structure. SingleAgentTD3 has achieved high efficiency with a decentralized learning strategy, but has exhibited instabilities in terms of reward and loss. SAC, under certain conditions, has managed to learn but has experienced stability issues. DDPG, despite trying to learn quickly, has limited its potential due to deficiencies on the critic side. The random agent has failed on all metrics and can only be used for lower bound comparison.

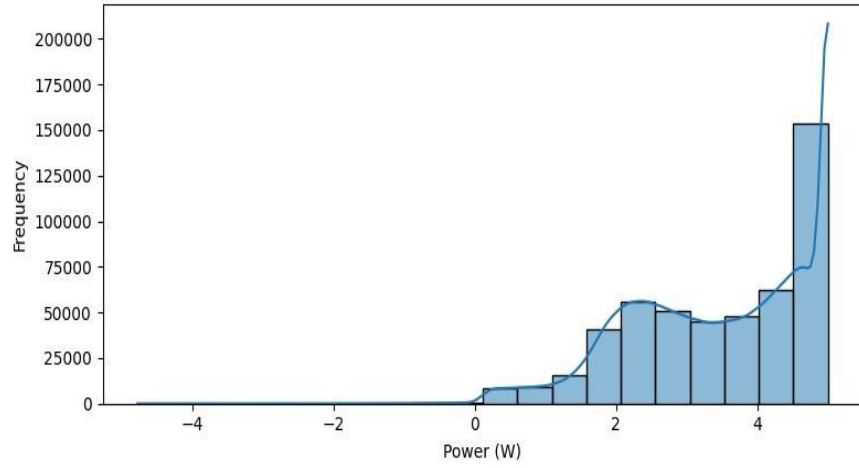


**Figure 4.11:** Distribution of Transmit Power for SAR (Watt)

### 4.3. Experiment 3

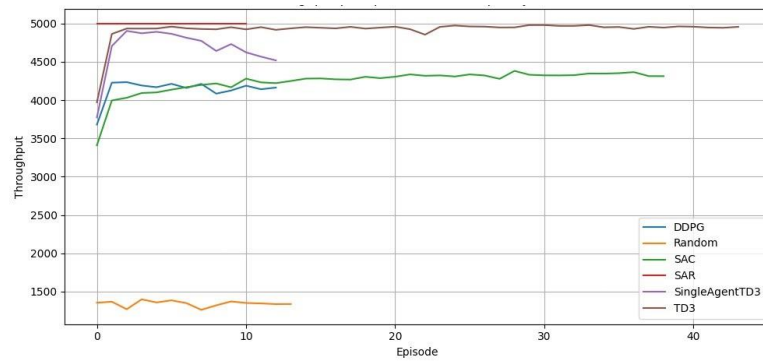
In Experiment 3, as a result of evaluating the resilience, generalization, and learning capacities of UAV agents in the simulation environment against different types of jammers,

Adaptive jammer is an agile threat model that intervenes in the most heavily used resources by analyzing the traffic on the channel. In this scenario, the performance of the six tested algorithms has been evaluated through various metrics: drop rate, throughput, actor and critic loss values, and overall learning behavior as summarised in **Table 4.1**.

One of the most successful results was achieved by the TD3 algorithm. It achieved almost flawless transmission with a drop rate of approximately 2-3%, which is quite low. While operating with an average throughput value of around 4950, it has successfully balanced both learning stability and reliable value predictions with an actor loss level of -60 and a critic loss range of  $\sim 0.3$ – $0.5$ . For these reasons, TD3 stands out as the most balanced and powerful agent. **Figure 4.12** gives a throughput plot under adaptive jamming. Although the SAC algorithm demonstrates strong performance in terms of discovery capability, the drop rate value hovering around 30% reveals the system's sensitivity to jammers. Reaching levels of 4400 in terms of throughput, SAC has achieved deep

**Table 4.1:** Performance Comparison Under Adaptive Jamming

| Algorithm      | Drop Rate | Throughput | Actor Loss | Critic Loss | Overall                                     |
|----------------|-----------|------------|------------|-------------|---------------------------------------------|
| TD3            | 2-3       | 4950       | 60         | 0.3-0.5     | The Most Balanced and Strong                |
| SAC            | 30        | 4400       | 225        | 1.0-1.1     | Good in Exploration but Sensitive to Noise  |
| DDPG           | 35        | 4200       | 200        | 0.6         | Agresive Learning, Not Stable               |
| SingleAgentTD3 | 20        | 4700       | 5          | 6.5         | Good Start but High Deviations              |
| SAR            | 0         | 5000       | 0          | 0           | No Learning Highest Success with Rule Based |

**Figure 4.12:** Throughput Per Episodes Under Adaptive Jamming

Although the SAC algorithm demonstrates strong performance in terms of discovery capability, the drop rate value hovering around 30% reveals the system's sensitivity to jammers. Reaching levels of 4400 in terms of throughput, SAC has achieved deep learning with an actor loss value of -225. Actor loss values are plotted in **Figure 4.13**.

However, the critic loss at around  $\sim 1.0$ – $1.1$  indicates that the value predictions are not as reliable as those of TD3. Still, it can be considered a strong alternative in this environment.

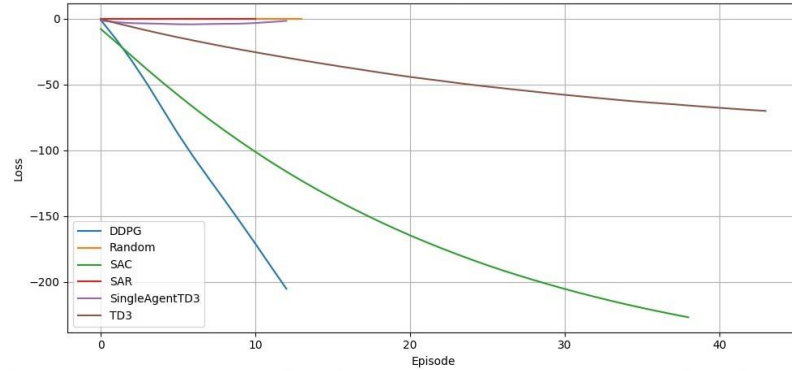


**Figure 4.13:** Actor Loss Under Adaptive Jamming

DDPG has undergone a very aggressive learning process with an actor loss around -200. However, this aggressiveness has caused instability in the system over time, resulting in a high drop rate of approximately 35% (**Figure 4.14**). The throughput has stayed at a satisfactory level of 4200. The  $\sim 0.6$  critic loss is decent, but not as stable as TD3.

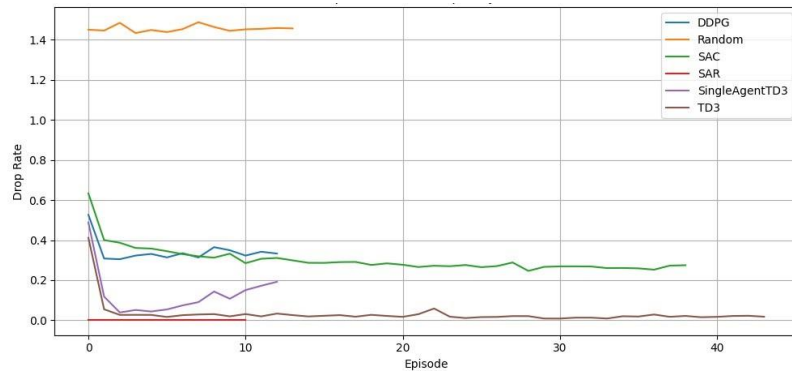


**Figure 4.14:** Drop Rate Under Adaptive Jamming

SingleAgentTD3, despite using a decentralized structure, initially showed a strong learning tendency. However, the quite high critic loss values of around  $\sim 6.5$  indicate that the model experiences significant deviations in value predictions over time. Although the throughput is at a high level of 4700, the 20% drop rate raises questions about the sustainability of this performance.

Despite using a deterministic rule-based structure without learning, the SAR algorithm has attracted attention with a drop rate close to 0% and the highest throughput value of around 5000. Despite the lack of an actor or critic network, its fixed structure has allowed it to perform well in an adaptive jammer environment. This example shows how fixed rules might be useful against certain types of threats.

Since the Random algorithm chooses actions at random and so not have any learning strategy, It has the lowest performance of the agents, with a drop rate of 145% and a throughput value of only 1300.

A comb jammer is an adversarial structure that simultaneously targets specific sections of the frequency spectrum and performs fixed bandwidth jamming. The performance of the algorithms tested under these conditions has been compared based on drop rate, throughput, actor, and critic loss values ((Table 4.2).

**Table 4.2:** Performance Comparison Under Comb Jamming

| Algorithm      | Drop Rate | Throughput | Actor Loss | Critic Loss | Overall                                          |
|----------------|-----------|------------|------------|-------------|--------------------------------------------------|
| TD3            | 8         | 4800       | 70         | 0.6         | Balanced                                         |
| SAC            | 55        | 3650       | 270        | 2.8         | Struggling                                       |
| DDPG           | 50        | 3900       | 550        | 1.8         | Excessively Agresive                             |
| SingleAgentTD3 | 5         | 4900       | 40         | 8.0         | The Highest Throughput But High Critic Deviation |
| SAR            | 16        | 4600       | 0          | 0           | Stable But No Learning                           |
| RANDOM         | 84        | 2900       | 0          | 0           | Risky                                            |

The TD3 algorithm has been the learning agent that exhibits the most balanced performance in this environment when compared to other algorithms as summarised in Table 4.2. While experiencing a drop rate of approximately 8%, it has managed to produce a high throughput at the level of 4800 with very low packet loss. Success can be easily observed from Figure 4.15. The actor loss at around -70 and the critic loss at approximately 0.6 indicate that learning is occurring in a stable and reliable manner.

With these characteristics, TD3 has emerged as a standout algorithm in terms of overall success and balance.

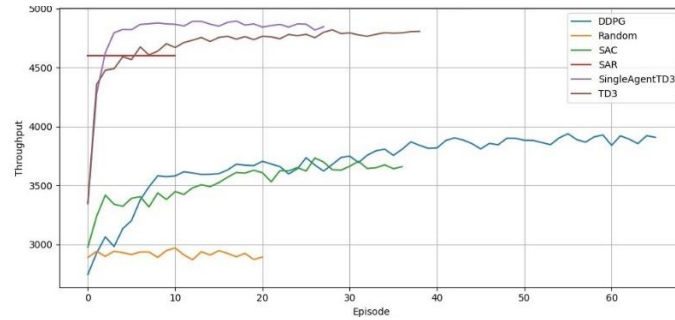


**Figure 4.15:** Throughput Per Episode Under Comb Jamming

SingleAgentTD3, despite its decentralized structure, has achieved the highest throughput level of 4900. However, alongside a 5% drop rate, a significantly high critic loss value of around 8.0 has been observed. This situation indicates that despite the agent making correct decisions, it experiences significant deviations in value predictions, and these deviations could negatively impact the quality of learning in the long term. **Figure 4.16** gives critic loss values in the scenario.

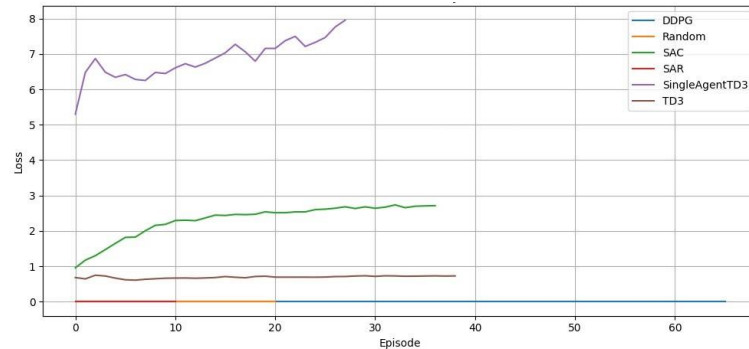


**Figure 4.16:** Critic Loss Under Comb Jamming

Despite its fixed decision structure without learning, the SAR algorithm has demonstrated a highly stable and strong performance with a throughput of 4600 and a drop rate of 16% (**Figure 4.17**). The absence of actor and critic loss values confirms the deterministic nature of the structure; its success demonstrates that a rule-based approach can be effective against threats with specific patterns, such as a comb jammer.

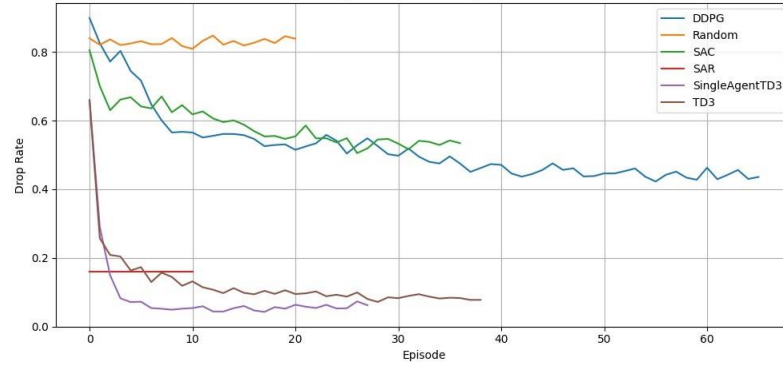


**Figure 4.17: Drop Rate Under Comb Jamming**

The SAC algorithm has clearly struggled in this scenario. A drop rate of approximately 55% has been observed, which suggests that it is not managing channel transitions accurately enough with entropy-supported exploration. Alongside a throughput value of 3650, an actor loss of -270 and a critic loss of  $\sim 2.8$  indicate that learning is ongoing, but the desired efficiency has not been achieved due to the high levels of interference in the environment. Actor loss were plotted as **Figure 4.18**.

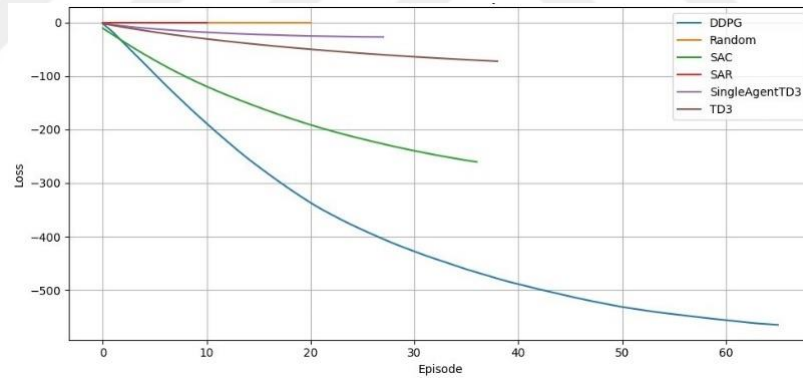


**Figure 4.18: Actor Loss Under Comb Jamming**

DDPG has exhibited an excessively aggressive learning process with very deep actor loss values like -550. However, this aggressiveness has caused a loss of stability in the system and resulted in a high drop rate of around 50%. It has achieved limited success with 3900 throughput and  $\sim 1.8$  critic loss. This example shows that quick but overfitting learning might jeopardise stability.

In contrast, the Random algorithm performed the poorest in this circumstance. With an 84% drop rate and a throughput value of 2900, its structure, which lacks a learning or strategic decision-making process, is inefficient and dangerous from a system

standpoint.

As a result, the most balanced algorithm is TD3. It stands out in terms of the balance between learning, stability, and efficiency. Highest throughput belongs to SingleAgentTD3, despite being the most successful agent in data transmission, harbors trust issues in value predictions. Hidden hero is SAR because it has achieved remarkable success with high efficiency and an acceptable error rate without any learning.

Dynamic jammer is one of the most challenging scenarios for algorithms due to its constantly changing attack model. In this environment, the SingleAgentTD3 algorithm has stood out. With a drop rate of approximately below 10% and ~4700 throughput, it has proven to be both a reliable and effective agent. While the actor losses have decreased in a controlled manner, the critic losses have remained high (~6.5–7). This situation indicates that the learning process is successful in producing outcomes but struggles with value predictions. **Figure 4.19** and **Figure 4.20** renders throughput and drop rate per episode.

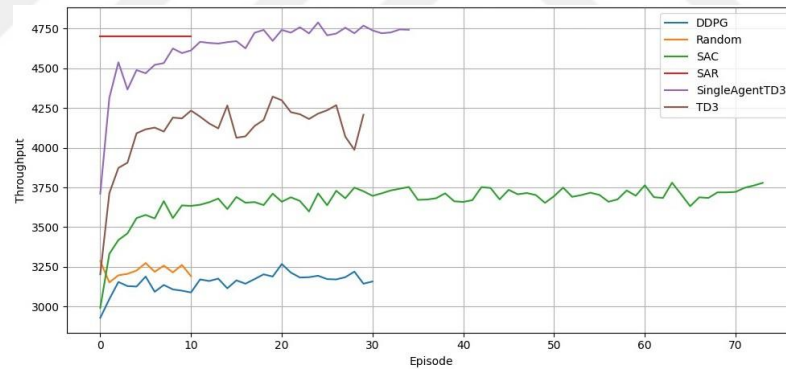


**Figure 4.19:** Throughput Per Episode Under Dynamic Jamming

TD3, despite a fluctuating drop rate of 30–40%, has generally achieved successful results with ~4300 throughput. Actor and critic losses are stable and moderate, indicating a resilient learning process against the complexity of the environment.

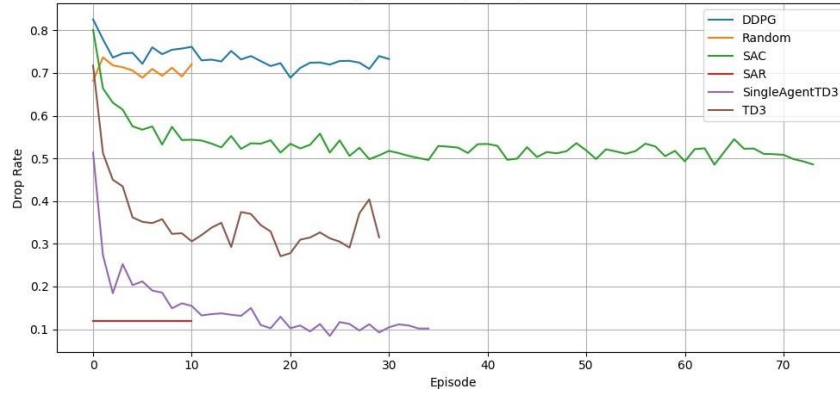


**Figure 4.20:** Drop Rate Per Episode Under Dynamic Jamming

SAR, with its fixed policy, has produced a 13% drop rate and ~4700 throughput, demonstrating high performance despite its non-learning structure.

The SAC and DDPG algorithms, on the other hand, have struggled in this high-entropy environment. SAC has been limited to an approximate 50% drop rate and ~3750 throughput. DDPG, on the other hand, has demonstrated poor performance with a 73% drop rate and only ~3200 throughput.

As expected, the random agent has again shown the lowest performance with a 70% drop rate and ~3250 throughput.

Smart jammer is a tactical threat that attacks by dynamically selecting specific targets. In this complex environment, SingleAgentTD3 has achieved the best results. With a drop rate of only 3-5%, it achieved the lowest data loss and became the most successful agent in the system with a throughput of ~4950 (**Figure 4.21**). While the actor's learning is at a sufficient level, the high losses of the critic are noteworthy. SAR has once again demonstrated quite a good performance; with a 15% drop rate and ~4600 throughput, it has shown that even fixed policies can work in this environment. TD3 has generally performed well with an approximately 35% drop rate and ~4200 throughput. Actor and critic losses are at a medium level.

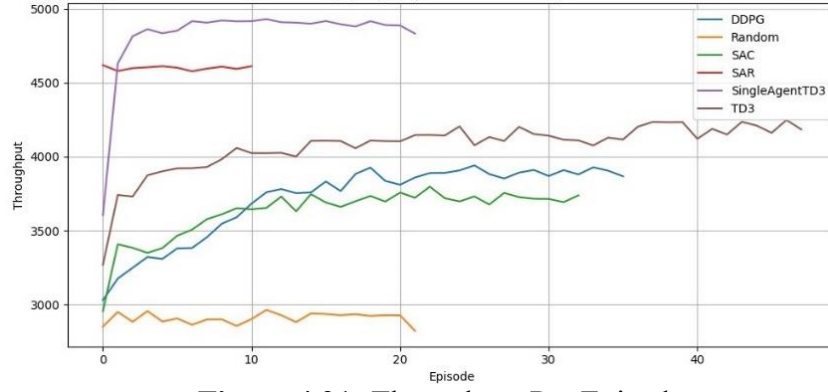


**Figure 4.21:** Throughput Per Episode

SAC and DDPG have been more vulnerable to the predictive attacks of the smart jammer. Both achieved low communication performance with a  $\sim 50\%$  drop rate; their throughput values were approximately  $\sim 3750$  and  $\sim 3900$ , respectively. The random agent has failed again, with an  $85\%$  drop rate and  $\sim 2900$  throughput, making it the most unsuccessful agent in the system. **Figure 4.21** exhibits drop rate plot.

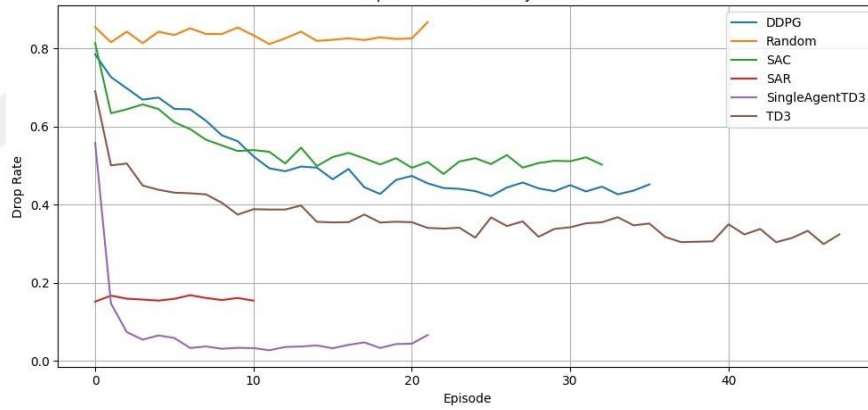


**Figure 4.22:** Drop Rate Per Episode Under Smart Jamming

Sweeping jammer is testing the algorithms with sequential channel attacks. **Figure 4.23** presents a clear sight for throughput results. TD3 has emerged as the most successful agent in this environment. With an approximate  $6\%$  drop rate and  $\sim 4880$  throughput, it has produced both effective and stable results. The actor loss is at a medium level, while the critic losses are quite low ( $\sim 2.1$ ). SingleAgentTD3 has produced similarly strong results with a  $\sim 6\%$  drop rate and  $\sim 4850$  throughput. However, the very high critic losses ( $\sim 8.1$ ) suggest that this learning might be unstable (**Figure 4.24**).

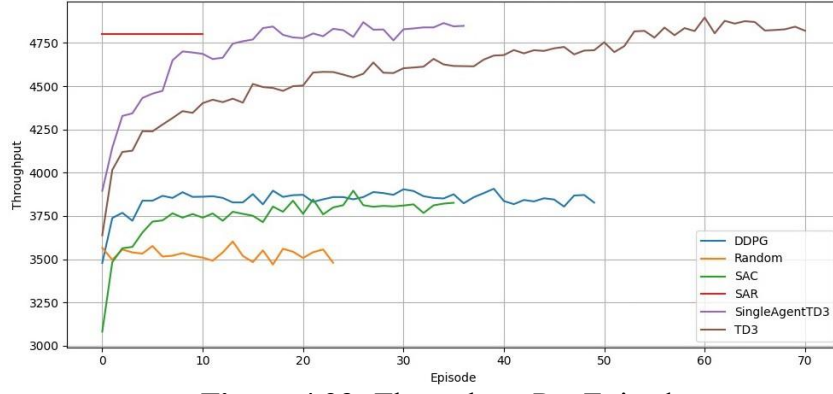


**Figure 4.23: Throughput Per Episode**

SAR is surprisingly successful in this environment as well. With an 8% drop rate and ~4800 throughput, it has maintained its stability despite its unsupervised structure. SAC and DDPG, on the other hand, struggled under this frequency-sweeping attack. Both were limited to a drop rate of 47-49% and a throughput of ~3800-3850.

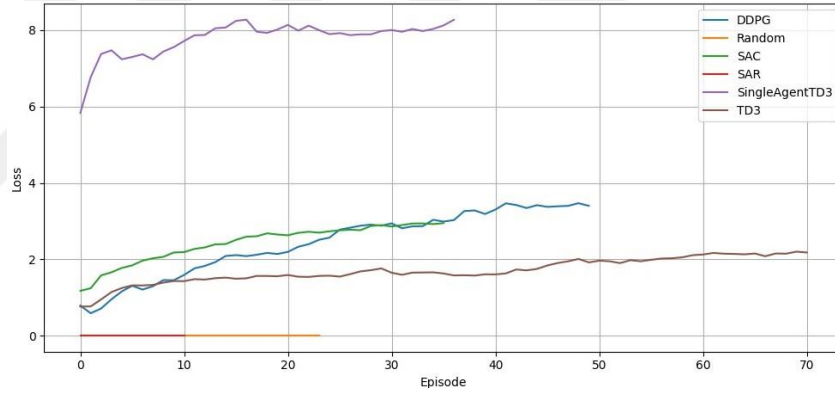


**Figure 4.24: Critic Loss Per Episode Under Sweeping Jamming**

The Random algorithm, as expected, has shown the lowest success rate with a 60% drop rate and ~3550 throughput.

Random jammer is a threat model that attacks completely random frequencies in the spectrum and tests the resilience of algorithms with its unpredictability. In this dynamic and irregular environment, the SAR algorithm, which lacks learning capability, has surprisingly demonstrated the highest success. With an approximately 8% low drop rate and ~4780 throughput value, it has shown a highly resilient performance against random attacks. It is noteworthy that this fixed structure, which does not involve any learning process, can make effective decisions even in the face

of randomness. This result shows that simple policies can also be effective in complex environments.

Among the learning agents, SingleAgentTD3 has demonstrated the most remarkable performance. Despite experiencing some packet loss with an approximate 25% drop rate, it achieved a very high data transfer rate of  $\sim 4370$  throughput in the system. However, a very high critic loss value of around  $\sim 10$  indicates that this agent has experienced significant value deviations during its learning process. Although the agent is successful in selecting the correct actions, the consistency of long-term value predictions is weak.

TD3 has demonstrated a reliable but somewhat passive performance. A 33% drop rate and  $\sim 4150$  throughput indicate that this agent sometimes struggles against random attacks. The actor loss value being around  $-60$  and the critic loss being maintained at approximately  $1.5$  indicates that the decision-making mechanism is stable but not fully adapting to environmental uncertainties.

SAC and DDPG algorithms have shown a clear downward trend in this scenario. Both of their drop rates have been around 50%. The throughput values remained at approximately  $\sim 3750$  (SAC) and  $\sim 3650$  (DDPG) levels, respectively. Especially, the very high actor loss of DDPG at around  $-500$  indicates that the learning process was excessively aggressive and unstable, while the  $-240$  actor and  $\sim 2.5$  critic loss values of SAC show that its exploration-focused structure was ineffective in such a chaotic environment.

The random algorithm has shown weak performance in this environment as expected. With a high drop rate of 85% and only  $\sim 3500$  throughput, this agent has exhibited a subpar performance that serves as a reference point. Its lack of any learning and its tendency to make completely random decisions have resulted in failure in such threat environments. For a random jamming scenario, best overall success belongs to SAR that is unsupervised but highly resistant to randomness. SingleAgentTD3 performs a strong candidate for data transfer. It obtains high throughput, but value estimation is unstable. Struggling agents are DDPG and SAC. They have excessive aggressiveness and exploration have been inefficient in this environment. TD3, here, in random scenario performs a stable learning, but low flexibility under randomness.

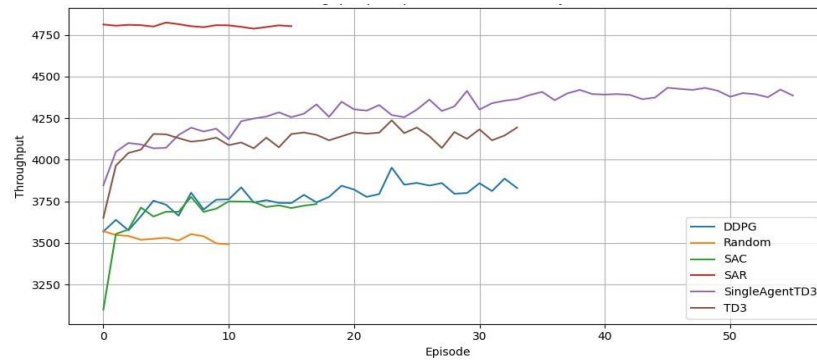


**Figure 4.25:** Throughput Per Episode

For an overall evaluation, **Table 4.3** gives a success ranking according to jamming techniques. Up to this point, the performance of all algorithms against each type of jammer had been thoroughly compared. However, how did each algorithm perform in different jammer scenarios?

**Table 4.3:** Success Ranking of Algorithms

| Jammer   | 1.             | 2.             | 3.  |
|----------|----------------|----------------|-----|
| Adaptive | SAR            | TD3            | SAC |
| Comb     | TD3            | SingleAgentTD3 | SAR |
| Dynamic  | SingleAgentTD3 | SAR            | TD3 |
| Random   | SAR            | SingleAgentTD3 | TD3 |
| Smart    | SingleAgentTD3 | SAR            | TD3 |
| Sweeping | TD3            | SingleAgentTD3 | SAR |

In this section, the behaviors of each algorithm under six different jammer types (adaptive, comb, dynamic, random, smart, sweeping) have been comprehensively examined. Thus, the overall resilience, learning capacity, and strengths/weaknesses of the algorithms can be more clearly understood. Some algorithms may be successful against certain types of jammers while being weak against others. An algorithm that appears successful under a single jammer may have low overall resilience. From this perspective, such a test will provide a positive contribution. The metrics used are actor loss / critic loss, drop rate and throughput.

DDPG (Deep Deterministic Policy Gradient) exhibited a generally aggressive but unstable learning curve under different jammer scenarios throughout experiment 3, with instabilities particularly observed in high-entropy environments, while learning was more effective in more stable jammer scenarios.

Under the adaptive jammer, although the actor loss initially experienced a rapid decline, it was observed that the learning process stopped shortly thereafter. This raises the possibility of early stopping. In sweeping and dynamic jammer scenarios, the actor loss has decreased quite rapidly and significantly. This situation indicates that learning occurs more stably and consistently in these environments. In the comb jammer scenario, the loss decreased more slowly, suggesting that learning was more difficult and unstable (**Figure 4.26**).

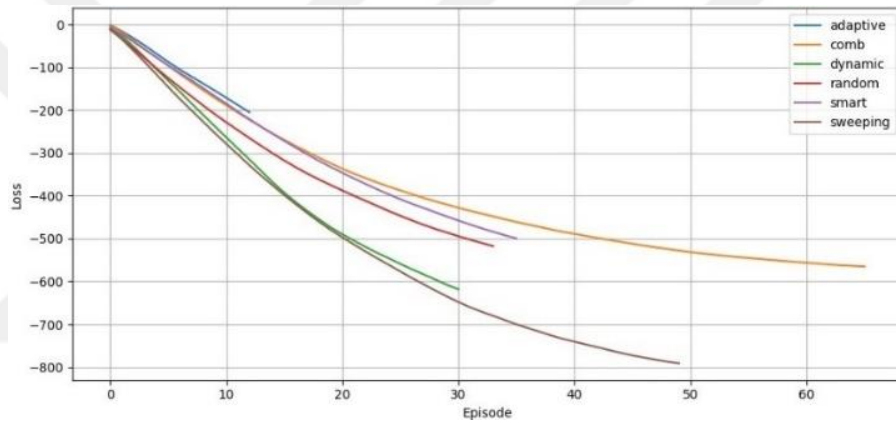


**Figure 4.26:** Actor Loss Performance Of DDPG

Critic1 loss has significantly increased under the dynamic jammer (~around 10). This indicates that there are deviations in the prediction of Q-values. Under the adaptive jammer, the critic1 loss is lower and more stable, which means relatively more reliable value predictions (**Figure 4.27**).

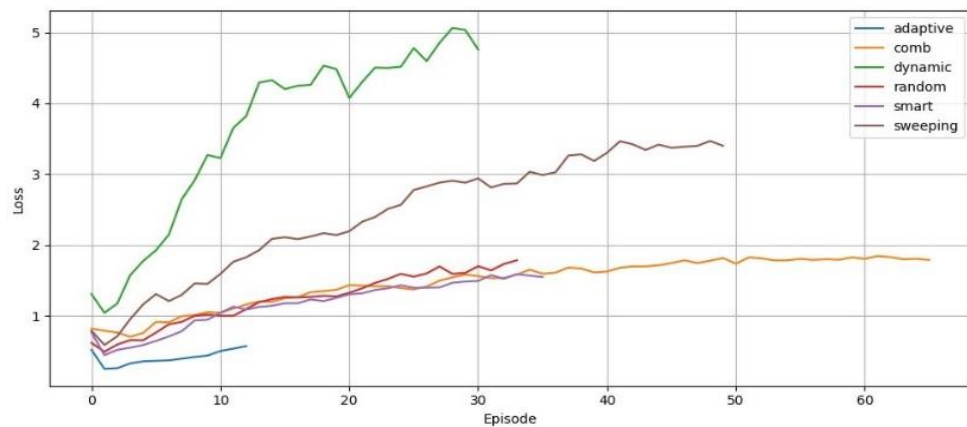


**Figure 4.27:** Critic1 Loss Performance of DDPG

The Critic2 loss graph has remained a straight line at zero in all scenarios. This situation indicates that the second critic network is not being updated or used. Although DDPG is actually a precursor to TD3, the lack of a double-critic structure can reduce the quality of learning (**Figure 4.28**).

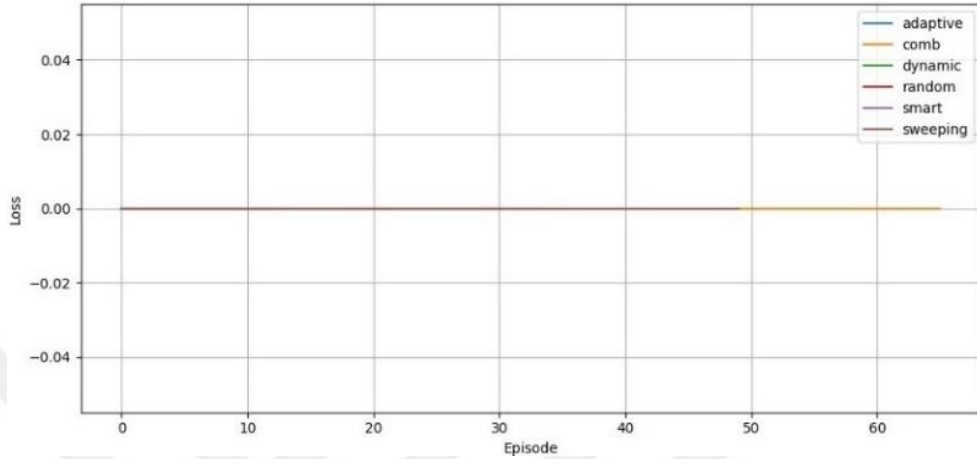


**Figure 4.28: Critic2 Loss of DDPG**

The lowest drop rate was observed in the adaptive jammer scenario ( $\sim 0.3\%$ ). This situation indicates a short but effective learning process. Higher packet losses have been observed under comb and dynamic jammers ( $\sim 30\text{--}35\%$ ). Over time, a downward trend has been observed in all jammer scenarios, but the dynamic jammer environment appears to be the weakest in terms of recovery (**Figure 4.29**).

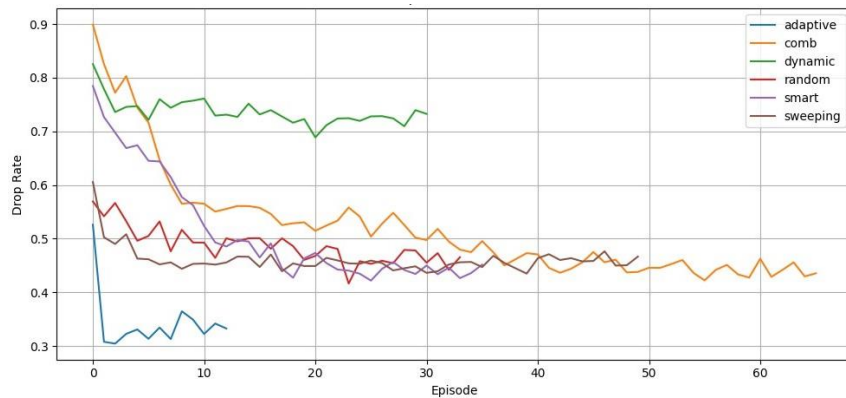


**Figure 4.29: Drop Rate of DDPG**

In the adaptive jammer scenario, the highest throughput value ( $\sim 4200$ ) has been achieved. In comb and dynamic scenarios, the throughput values are lower (around  $\sim 3900$  and  $\sim 3100$ ).

In the smart, sweeping, and random scenarios, the throughput values have ranged around  $\sim 3700$ – $3900$ . However, indirect analyses have been conducted based on throughput and drop rate data (**Figure 4.30**).

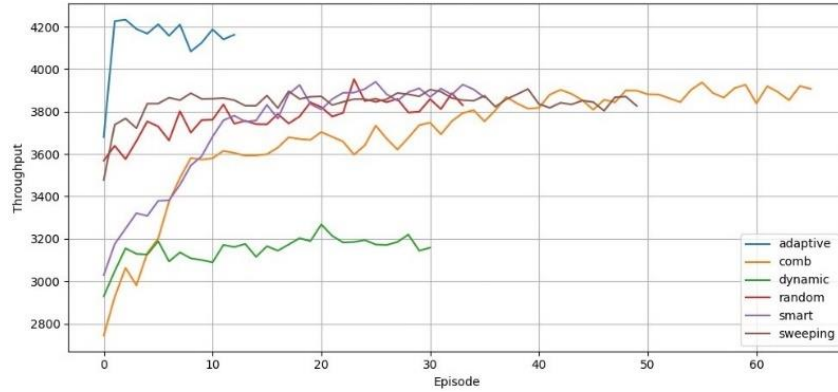


**Figure 4.30:** Throughput of DDPG

Adaptive jammer has been the most successful scenario for DDPG: low drop rate and high throughput. Dynamic jammer, Q-value estimation deviations, and high critic1 loss values stand out as the environment where DDPG struggles the most. Since the Critic2 network is not updated, the performance of this algorithm is potentially limited compared to TD3.

The Twin Delayed DDPG (TD3) algorithm has stood out as one of the most balanced and powerful agents in Experiment 3. It has produced stable results under various types of jammers in terms of both signal quality and energy efficiency.

Under the sweeping jammer, the actor loss has decreased rapidly and steadily, reaching levels of -45. This indicates that TD3 has achieved stable and robust policy learning in this environment. In the dynamic, random, and smart jammer scenarios, the loss ranged from -30 to -35. This indicates that it is an acceptably effective learning process. The reduction under the comb jammer has occurred more slowly ( $\sim -25$ ), indicating that the learning pace is somewhat lower in this scenario. In the adaptive jammer environment, the actor loss value has remained quite irregular and high ( $\sim 0$  to  $-5$ ), suggesting that TD3 has struggled to develop a stable policy against this type of attack (**Figure 4.31**).

Critic1 loss has increased over time in all scenarios. Especially under random jammer, it has reached the highest levels ( $\sim 10$ ). This indicates that the Q-value predictions show deviation in this environment. Under the smart and adaptive jammer, TD3 made more stable value predictions with the lowest loss values ( $\sim 4$ – $6$ ). In the comb, sweeping,

and dynamic jammer scenarios, the loss values have remained at moderate levels ( $\sim 6.5\text{--}8$ ) as illustrated in **Figure 4.32**.

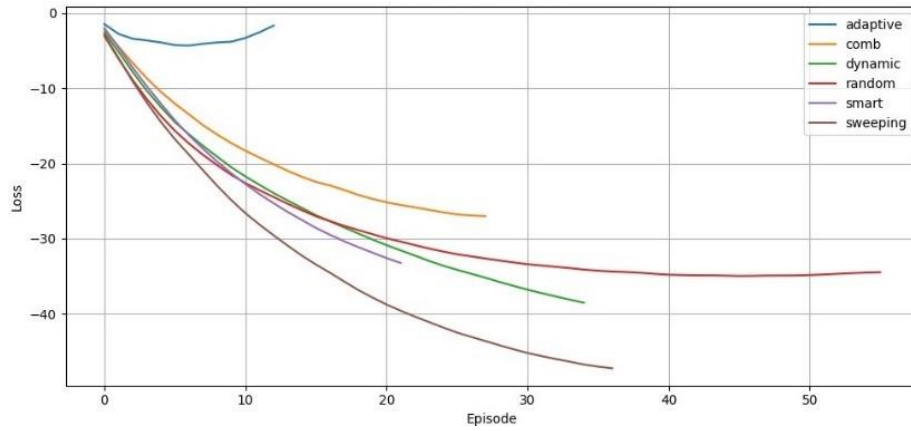


**Figure 4.31:** Actor Loss of TD3 Under Different Jamming Types

Critic2 loss values have shown similarity with critic1 (**Figure 4.33**). This indicates that both networks are being updated similarly and progressing in parallel in terms of loss levels. In the comb, smart, and sweeping jammer scenarios, the lowest drop rate values ( $\sim 5\%$ ) were achieved. This shows that TD3 provides quite effective packet delivery in these environments.

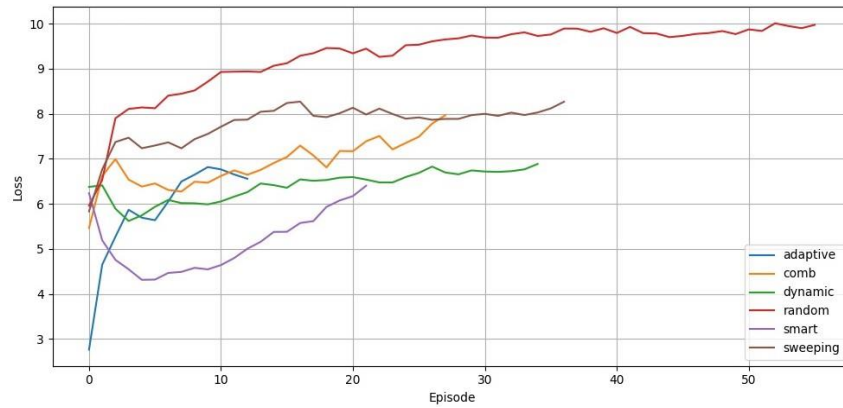


**Figure 4.32:** Critic1 Loss of TD3 Under Different Jamming Types

In the adaptive jammer environment, the drop rate initially decreased and then rose to around 10%, which may indicate that TD3 became unstable over time against this type of jammer. Under the random jammer, the drop rate has fluctuated between 25-30%. This shows that random attacks challenge TD3's decisions (**Figure 4.34**).

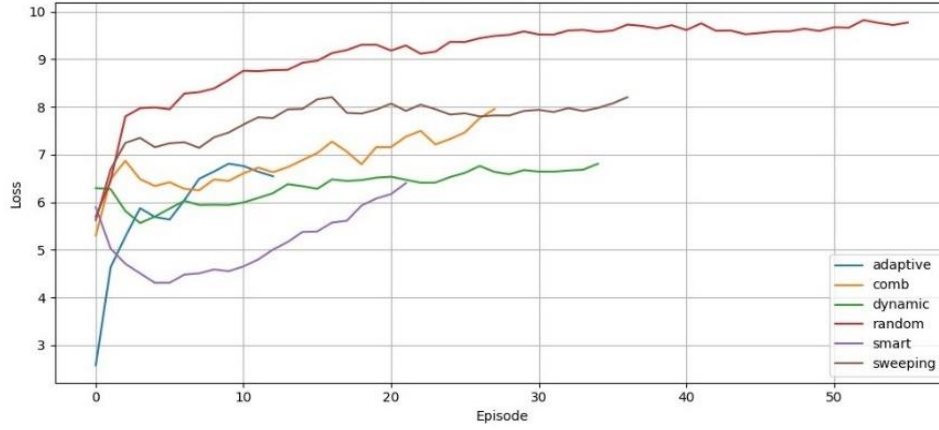


**Figure 4.33:** Critic2 Loss of TD3 Under Jamming Types

Under sweeping, smart, and comb jammers, TD3 has demonstrated quite efficient performance with approximately 4900 packets throughput. In the random jammer scenario, the throughput is lower (~around 4400), which is consistent with the high drop rate. The throughput value obtained under the adaptive jammer is approximately around 4700–4800, but it is difficult to make a clear judgment due to the short training time in this environment (**Figure 4.35**).

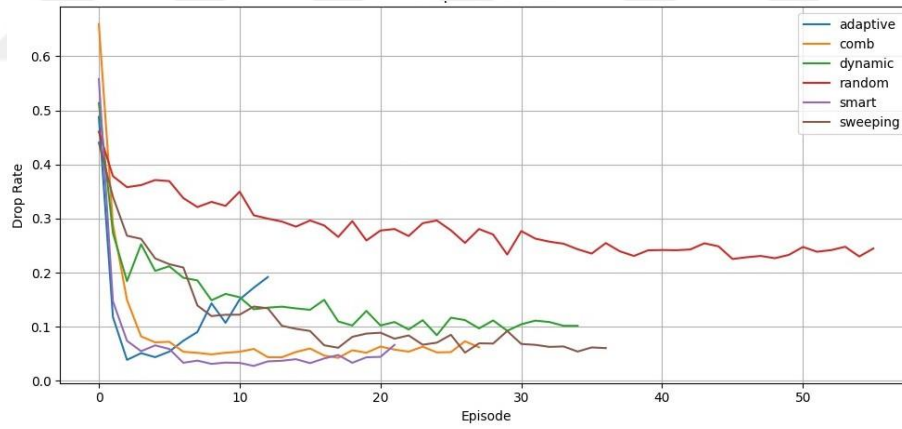


**Figure 4.34:** Drop Rate of TD3 Under Different Jamming Types

When assessed within the framework of Experiment 3, SAC's performance varied significantly depending on the type of jammer. The SAC actor network has consistently showed learning in all jammer conditions. Under sweeping and dynamic jammers, actor loss has shown a rapid and strong decline curve, indicating aggressive and effective policy learning.

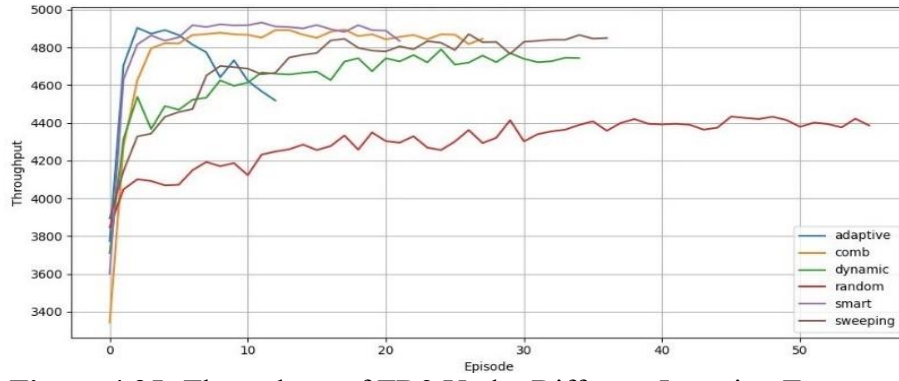


**Figure 4.35:** Throughput of TD3 Under Different Jamming Types

Under the adaptive jammer, the actor loss decreased more slowly, indicating a more cautious learning process in response to the predictable structure of the environment. Actor loss values have generally decreased to around -240 levels (**Figure 4.36**).

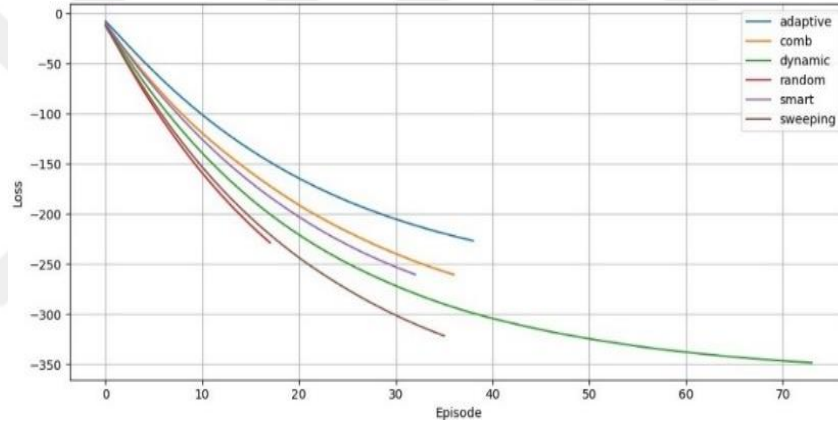


**Figure 4.36:** Actor Loss of SAC Under Different Jamming Types

Under the dynamic jammer, both Critic1 and Critic2 losses have remained high ( $\sim 2.5-3$ ), indicating that predicting Q-values in a high-entropy environment is challenging. In the adaptive jammer scenario, however, the critic loss is quite low ( $\sim 1.0$ ), indicating that SAC can make stable value predictions here. In the comb, smart, random, and sweeping jammer scenarios, the critic loss has remained at a moderate level. In the sweeping environment, especially more balanced results have been achieved. **Figure 4.37** illustrates Critic1 Loss behaviour under all those jamming types.

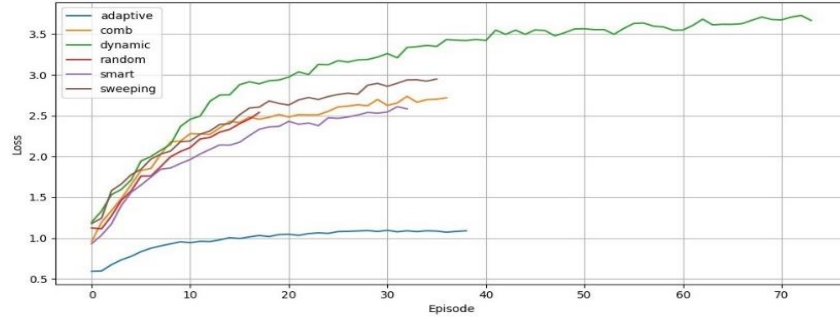


**Figure 4.37:** Critic1 Loss of SAC Under Different Jamming Types

Under the adaptive jammer, the SAC algorithm achieved one of its most successful results with a drop rate below 25%. In the comb, dynamic, random, and smart jammer environments, the drop rate has generally remained high (~around 50%), indicating that exploration-focused strategies in jamming scenarios may lead to stability issues. In the sweeping jammer scenario, the drop rate decreased over time, providing a moderate level of success (**Figure 4.38**).

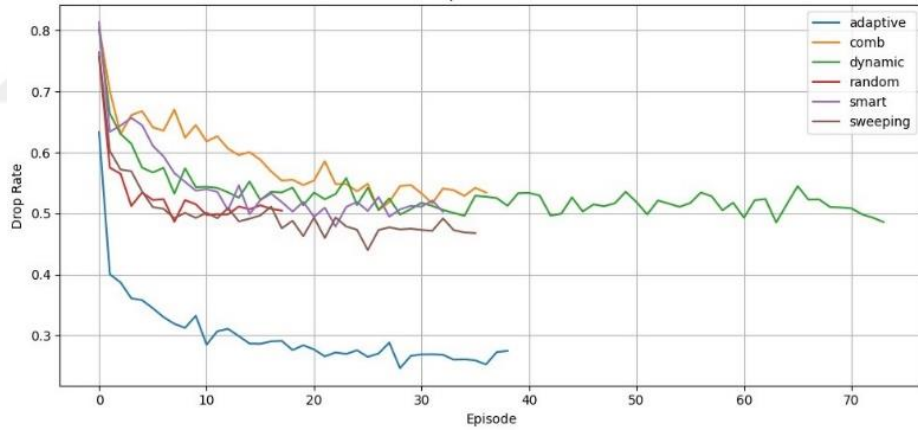


**Figure 4.38:** Drop Rate of SAC Under Different Jamming Types

The highest throughput was observed under the adaptive jammer (~4400). Under the random jammer, the performance is also relatively successful (~3750). In comb, dynamic, smart, and sweeping jammer scenarios, throughput is generally low (in the range of ~3600–3900). This shows that the high drop rate and critical deviations directly reflect on transmission efficiency. The entropy-based exploration strategy provides an advantage in stable environments but can cause significant performance losses against chaotic jammers (**Figure 4.39**).

In conclusion, the SAC algorithm has shown stable and high performance, especially in predictable threat environments such as adaptive jammers. However, it has struggled with high drop rates and low throughput values in more chaotic scenarios such as dynamic, smart, and random. SAC's entropy-based exploration strategy provides an advantage in stable environments but causes instabilities in situations with high variability.

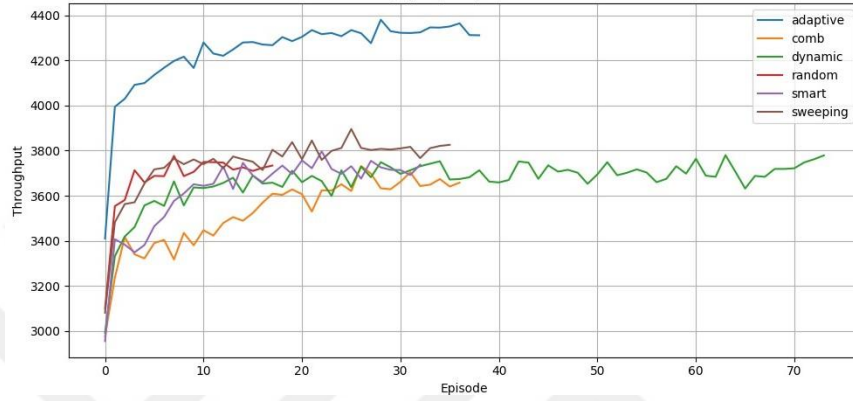


**Figure 4.39:** Throughput of SAC Under Different Jamming Types

For all types of jammers, actor loss of SingleAgentTD3 is steadily decreasing; this indicates that the agents are undergoing an effective policy learning process. Under sweeping and dynamic jamming, actor loss showed the fastest decline and dropped to around 45. Under the adaptive jammer, the actor loss has fluctuated more and remained around 5; this indicates that this environment is more resistant to learning. The highest critic loss values were observed under the random jammer (~around 10); this indicates that the unpredictability makes it difficult to calculate the Q-value. In the adaptive jammer environment, the critic losses remained lower and more stable in the range of ~6-7, which indicates a reliable learning process. Under the smart and sweeping jammer, the critic loss has shown a slow and controlled increase. Under comb, smart, and sweeping jammers, the drop rate is quite low (~5%), which indicates that agents are very successful in channel and target selection in these environments. Under adaptive and dynamic jammers, the drop rate is slightly higher (~12-20%). In a random jammer environment, the drop rate is higher (~25%), indicating that the algorithm struggles against the unpredictability of the environment.

In the environments of comb, smart, and sweeping jammers, throughput has reached the highest levels (~4900–4950), which are the scenarios where this algorithm works

most efficiently. In the adaptive jammer environment, although the throughput is high, the curve is more irregular (around  $\sim 4700$ – $4800$ ). Under a random jammer, the throughput value is lower ( $\sim 4370$ ), indicating a decrease in decision quality (**Figure 4.40**). SingleAgentTD3 has been extremely successful, especially in comb, smart, and sweeping jammer environments. Despite its decentralized structure, it has drawn attention with high throughput and low drop rate values.

In high-entropy environments like random jammers, the increase in critical loss and the decrease in throughput have revealed the limitations of this algorithm. The SAR algorithm is a lightweight approach that operates with a rule-based deterministic decision-making mechanism without any learning process. With these features, it offers low computational cost and high interpretability compared to learning-based agents. Since the SAR algorithm does not perform learning, there are no Actor, Critic1, and Critic2 networks. Therefore, the values in all the loss graphs are zero, which is as expected and is a structural characteristic of the algorithm.

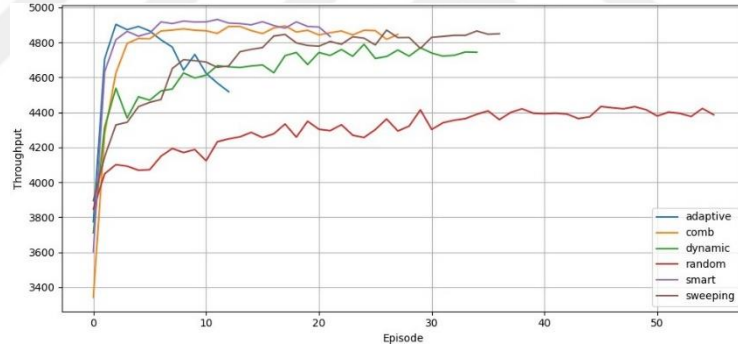


**Figure 4.40:** Throughput of SingleAgentTD3 Under Different Jamming Types

The deterministic structure of SAR can be quite effective, especially in predictable jammer environments: The lowest drop rate ( $\sim 0\%$ ) was achieved under the adaptive jammer. This suggests that rule-based decisions can be quite effective against certain attack patterns. In sweeping, random, and smart jammer environments, the drop rate has generally hovered around  $\sim 15$ – $17\%$ . Under the dynamic jammer, a stable value of around  $\sim 12\%$  has been observed. Drop rate fluctuations are at a low level, indicating that SAR has a stable but limited decision-making structure (**Figure 4.41**).

The throughput performance of the SAR algorithm is also noteworthy. The highest throughput value ( $\sim 5000$ ) has been achieved in the adaptive jammer environment. Under sweeping and random jammers, throughput has also remained at high and stable

values around 4800. In the smart and comb jammer scenarios, throughput remained stable at around 4600. In the dynamic jammer environment, throughput is slightly lower ( $\sim 4700$ ), but it is still comparable to the learning algorithms. Indirectly, performance can be inferred from the drop rate and throughput values.

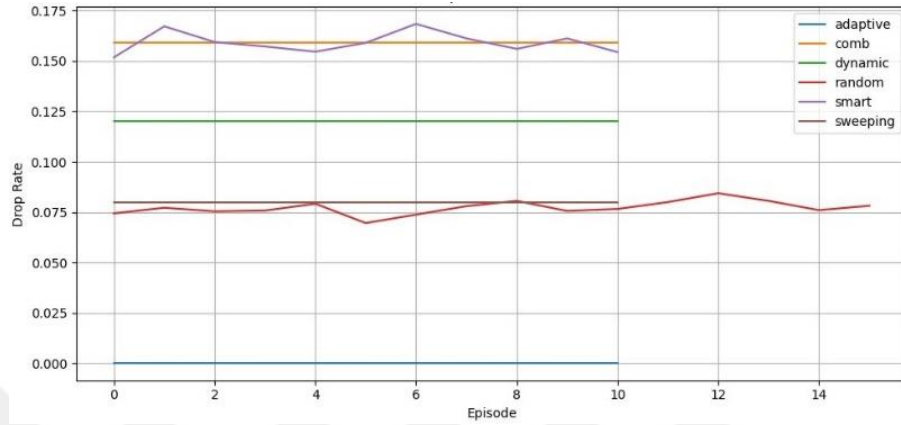


**Figure 4.41:** Drop Rate of SAR Under Different Jamming Types

Although the SAR algorithm does not involve learning, it has produced quite successful results in more predictable scenarios such as adaptive and random jammers. Especially under the adaptive jammer, it has stood out with the highest throughput and the lowest drop rate values. However, its lack of being learning-based limits its performance in complex scenarios.

In this regard, SAR can play an important role in system design as a low-cost, understandable, and robust baseline algorithm.

The random agent is a basic comparison agent that does not contain any learning algorithm and makes all its decisions completely randomly. Therefore, no significant change is observed in either the loss. However, this agent serves an important "control" role in highlighting the value of learning algorithms and analyzing the difficulty level of the system.

The random agent does not have actor and critic networks, so all loss values remain constant at zero. This situation indicates that, as expected, it is not a systematic learning process and the weights are not being updated. The absence of loss lines or their presence as a flat line at zero in the graphs is a natural consequence of this structure.

The random agent's random decision-making mechanism severely weakens communication security, resulting in the highest drop rate value (~145%) against the Adaptive jammer; the fact that this value is over 100 indicates that the same packet was dropped multiple times and that the system is operating unstably. In comb, dynamic, smart, and sweeping jammer scenarios, the drop rate has generally remained high in the range of 60–85%. These ratios indicate that random decisions lead to serious instability and high error rates in the system (**Figure 4.42**).

In terms of data transmission capacity, the Random agent also falls short. Under the adaptive jammer, the throughput value is only around 1300. This is the lowest value among all the tested agents. In the sweeping and random jammer scenarios, the throughput value has reached levels of ~3500–3600, but this is still quite low compared to the 4500–4900 range achieved by the learning agents. Random decisions seriously reduce the overall efficiency of the network by preventing effective channel and target selection.

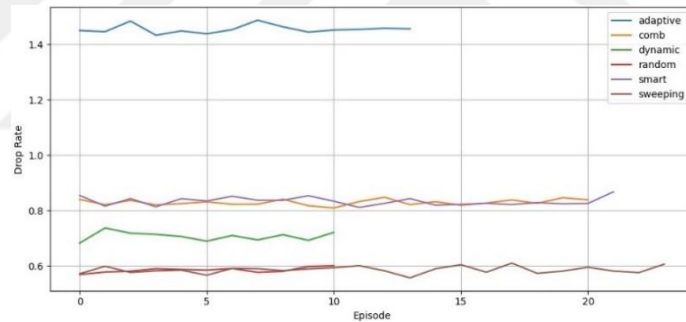


**Figure 4.42:** Drop Rate of Random Algorithm Under Different Jamming Types

This situation necessitates making inferences based on other metrics. The random agent has been the lowest-performing agent in the system. High drop rate, low throughput, zero loss, and reward curves indicate that this agent should only be used as a reference (baseline). The fact that this non-learning structure becomes almost useless, especially in adaptive or chaotic jammer environments, strongly demonstrates why learning agents should be preferred.

If we want to make a general inference, TD3 and SingleAgentTD3 have been the most effective learning agents in the experiment, achieving high success even in the most challenging jammer scenarios. SAC, particularly in stable scenarios like adaptive, has performed very well, while in complex environments like dynamic, it has experienced

critical deviations. SAR has drawn attention as a low-cost baseline agent with high throughput and low drop rate values achieved without learning. DDPG has not been able to reflect its significant potential due to the deficiency of the critic2 architecture; this situation necessitates the reconfiguration of the algorithm for further evaluations. The random agent has shown the lowest success, but it has served as an important reference for comparison with other algorithms.



## 5. CONCLUSION

In this thesis, DRL-based anti-jamming approaches have been developed and systematically evaluated to ensure the secure communication of UAV swarms in dynamic and hostile environments against various jammer threats. In the three-phase experimental structure, the problem space was gradually expanded: In the first experiment, agent behaviors were evaluated under an adaptive jammer on a single channel; in the second experiment, realistic elements such as a multi-dimensional action/observation space, varying jammer power per channel, and a TTL-based retransmission mechanism were added by increasing environmental complexity. In the final experiment, six different learning agents (TD3, SAC, DDPG, SingleAgentTD3, SAR, Random) were compared in a comprehensive benchmark simulation environment against six different types of jammers (adaptive, comb, sweeping, dynamic, smart, random).

This study goes beyond commonly used methods based on discrete action spaces, such as QL and DQL, and proposes improved policy gradient-based deep learning architectures suitable for this complex communication problem that requires continuous and multi-dimensional action spaces. The existing DDPG, TD3, and SAC algorithms have been adapted to the distributed communication environment of UAV swarms; they have been structured to simultaneously optimize multidimensional decisions such as channel selection, target UAV identification, relay usage, and transmission power. In this regard, the study presents a unique and advanced application in terms of both architectural and environmental complexity.

In light of the findings, the TD3 algorithm has achieved the highest success in most jammer scenarios with its stable learning process, twin critic architecture, balanced action production, , and low drop rate rates. For example, in the sweeping jammer scenario, TD3 provided both reliable and efficient communication with a low packet loss rate of 6% and an approximate throughput value of 4880. Furthermore, the consistency of the actor and critic loss values suggests that learning is happening in a regulated manner. SingleAgentTD3, with its decentralized structure, has been effective against sophisticated threats like smart and sweeping, but has occasionally experienced fluctuations in learning quality due to critic instability. In smart jammer environment, it has a drop rate of only 3% and a throughput of 4950. However,

significant critic losses raised concerns about its long-term reliability ( $\sim 8$ ). SAC has shown stable performance against some types of jammers with its entropy-supported exploration mechanism; however, fluctuations in the evaluation stability of the critic network have been observed in variable environments. The SAC algorithm, while effective in stable threat environments such as adaptive jammers with approximately 4400 throughput and a drop rate below 25%, has experienced performance loss in high-entropy environments such as dynamic jammers due to the instability of its critic networks. The actor loss curve of SAC, which drops to levels around -225, indicates a controlled but slow learning process. Although the DDPG algorithm offers rapid learning capability, it has been exposed to excessive optimism in Q-value predictions due to the lack of a dual evaluation structure in TD3, which has made the learning process unstable and fluctuating. The SAR algorithm, despite not having any learning mechanism, has achieved high successes such as 5000 throughput in adaptive and random jammer environments and has provided highly competitive results with a drop rate close to 0%. SAR stands out as a low-cost and deterministic solution, especially in predictable threat scenarios where learning is not required; the Random agent, on the other hand, has shown the lowest performance across all metrics, functioning merely as a baseline agent for control purposes with results such as a drop rate reaching 145% in an adaptive jammer environment and only 1300 throughput.

Additionally, both centralized and decentralized architectures were comparatively evaluated in the study; while coordination advantages were obtained in centralized structures, features such as flexibility and scalability were highlighted in distributed structures.

Within the scope of this thesis, a DRL architecture has been developed, capable of optimizing multidimensional decision processes encompassing channel selection, power control, relay usage, and target UAV identification, with improvements made on existing algorithms. This architecture has been successfully implemented in a realistic simulation environment where environmental complexity and decision spaces have increased; the agents' adaptation capabilities, energy usage, throughput, fairness, and success rates against different jammer strategies have been thoroughly analyzed through various metrics. Additionally, another significant outcome of the study is the development of an open-source and extensible benchmark infrastructure. This system not only facilitates researchers in comparing different algorithms but also lays the groundwork for more advanced applications.

In future studies, trends such as training models with real-world communication data, integrating task-oriented multi-objective learning approaches, developing ultra-efficient agents for energy-constrained UAV systems, and testing larger-scale distributed architectures with federated/cooperative learning methods are emerging. In this context, the thesis work provides a solid foundation for the development of UAV communication systems with learning, flexible, and autonomous decision-making capabilities against dynamic and unknown jamming threats, both in academic and defense/civil applications.



## REFERENCES

- [1] T. Seymour and A. Shaheen, "History of wireless communication," *Review of Business Information Systems*, vol. 15, no. 2, pp. 37–42, 2011.
- [2] L. Kleinrock and J. Silvester, "Optimum transmission radii for packet radio networks or why six is a magic number," *Proc. IEEE Natl. Telecomm. Conf.*, pp. 4.3.1–4.3.5, December 1978.
- [3] A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proc. IEEE*, pp. 56–73, January 1987.
- [4] J. H. Yeh, J. C. Chen, and C. C. Lee, "WLAN standards," *IEEE Potentials*, vol. 22, no. 4, pp. 16–22, Oct.–Nov. 2003.
- [5] J. Jubin and J. D. Tornow, "The DARPA packet radio network protocols," *Proc. IEEE*, pp. 21–32, January 1987.
- [6] M. B. Pursley, "The role of spread spectrum in packet radio networks," *Proc. IEEE*, pp. 116–34, January 1987.
- [7] F. A. Tobagi, "Modeling and performance analysis of multihop packet radio networks," *Proc. IEEE*, pp. 135–55, January 1987.
- [8] A. F. M. S. Shahen Shah, A. N. Qasim, M. A. Karabulut, H. Ilhan, and M. B. Islam, "Survey and performance evaluation of multiple access schemes for next-generation wireless communication systems," *IEEE Access*, vol. 9, pp. 113428–113442, 2021.
- [9] B. Leiner, R. Ruther, and A. Sastry, "Goals and challenges of the DARPA Glomo program (global mobile information systems)," *IEEE Pers. Commun. Mag.*, pp. 34–43, December 1996.
- [10] R. Ramanathan and R. Rosales-Hain, "Topology control of multihop wireless networks using transmit power adjustment," *Proc. IEEE Infocom Conf.*, pp. 404–13, March 2000.
- [11] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.
- [12] I. F. Akyildiz, W. Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey," *Computer Networks*, vol. 50, no. 13, pp. 2127–2159, Sep. 2006.
- [13] J. Mitola and G. Q. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13–18, Aug. 1999.
- [14] Q. Zhao and B. M. Sadler, "A survey of dynamic spectrum access," *IEEE Signal Process. Mag.*, vol. 24, no. 3, pp. 79–89, May 2007.

- [15] T. S. Rappaport *et al.*, “Millimeter wave mobile communications for 5G cellular: It will work!,” *IEEE Access*, vol. 1, pp. 335–349, 2013.
- [16] H. Pirayesh and H. Zeng, “Jamming attacks and anti-jamming strategies in wireless networks: A comprehensive survey,” *IEEE Commun. Surveys Tuts.*, vol. 24, no. 2, pp. 767–809, 2022.
- [17] Y. Liu, Z. Qin, Y. Cai, Y. Gao, G. Y. Li, and A. Nallanathan, “UAV communications based on non-orthogonal multiple access,” *IEEE Wireless Communications*, vol. 26, no. 1, pp. 52–57, Feb. 2019.
- [18] Y. Zeng, R. Zhang, and T. J. Lim, “Wireless communications with unmanned aerial vehicles: Opportunities and challenges,” *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, May 2016.
- [19] L. Gupta, R. Jain, and G. Vaszkun, “Survey of important issues in UAV communication networks,” *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1123–1152, 2nd Quart., 2016.
- [20] A. Rahmati *et al.*, “Dynamic interference management for UAV-assisted wireless networks,” *IEEE Trans. Wireless Commun.*, vol. 21, no. 4, pp. 2637–2653, Apr. 2022.
- [21] L. Zhou, S. Leng, Q. Liu, and Q. Wang, “Intelligent UAV swarm cooperation for multiple targets tracking,” *IEEE Internet Things J.*, vol. 9, no. 1, pp. 743–754, Jan. 2022.
- [22] K. N. Vaishnavi, S. D. Khorvi, R. Kishore, and S. Gurugopinath, “A survey on jamming techniques in physical layer security and anti-jamming strategies for 6G,” in *Proc. 2021 28th Int. Conf. Telecommun. (ICT)*, 2021, pp. 1–6.
- [23] H. Pirayesh and H. Zeng, “Jamming attacks and anti-jamming strategies in wireless networks: A comprehensive survey,” *IEEE Commun. Surveys Tuts.*, vol. 24, no. 2, pp. 767–809, 2022.
- [24] K. Karishma, P. Yadav, and R. S. Tomar, “Experimental Analysis of DSSS-based Anti-Jamming in Wireless Sensor Networks,” *International Journal of Computer Applications*, vol. 175, no. 12, pp. 15–21, 2022.
- [25] Y. Yan, M. Li, and C. Wang, “Spatial Anti-Jamming Strategy Based on Zero-Forcing in MIMO Systems,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1561–1575, Mar. 2019.
- [26] R. G. Gallager, “Low-density parity-check codes,” *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, Jan. 1962.
- [27] I. S. Reed and G. Solomon, “Polynomial codes over certain finite fields,” *J. Soc. Indust. Appl. Math.*, vol. 8, no. 2, pp. 300–304, 1960.

- [28] K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy, "Denial of service attacks in wireless networks: The case of jammers," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 2, pp. 245–257, Second Quarter 2011.
- [29] B. Wang, Y. Wu, K. J. R. Liu, and T. C. Clancy, "An anti-jamming stochastic game for cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 4, pp. 877–889, Apr. 2011.
- [30] Y. Zhou, Y. Fang, and Y. Zhang, "Securing wireless sensor networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 3, pp. 6–28, 3rd Quart. 2008.
- [31] H. Davaslioglu and Y. E. Sagduyu, "DeepWiFi: Cognitive WiFi with Deep Learning," *IEEE Transactions on Mobile Computing*, vol. 19, no. 7, pp. 1477–1490, July 2020.
- [32] O. Puñal, J. Gross, and A. Wolisz, "Jamming Detection in Wireless Networks: A Machine Learning Approach," in *Proceedings of the IEEE International Conference on Communications (ICC)*, Montreal, QC, Canada, 2021, pp. 1–6.
- [33] F. Zhang, J. Li, and Z. Li, "A TD3-based multi-agent deep reinforcement learning method in mixed cooperation-competition environment," *Neurocomputing*, vol. 411, pp. 206–215, 2020.
- [34] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Urbana-Champaign, IL, USA, 2005, pp. 46–57.
- [35] C. Zhou, Y. Fang, and Y. Zhang, "Securing wireless sensor networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 3, pp. 6–28, 2008.
- [36] M. P. H. Hanawal, M. J. Abdel-Rahman, and M. Krunz, "Joint adaptation of frequency hopping and transmission rate for anti-jamming wireless systems," *IEEE Transactions on Mobile Computing*, vol. 13, no. 5, pp. 1020–1033, May 2014.
- [37] Y. Liu, X. Liu, and Y. Chen, "Reinforcement learning in multiple access control: A multi-agent deep reinforcement learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2289–2302, Oct. 2020.
- [38] Rajendran, S., Abbas, R., & Fitzek, F. H. P. (2018). Deep learning models for wireless signal classification with distributed low-cost spectrum sensors. *IEEE Wireless Communications and Networking Conference (WCNC)*, 1-6.
- [39] Zhang, W., Yang, N., Ding, Z., & Li, H. (2020). Deep reinforcement learning for dynamic spectrum access in wireless networks: Model-based and model-free comparison. *IEEE Access*, 8, 120468–120479.
- [40] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2nd ed.). MIT press.

- [41] Naparstek, O., & Cohen, K. (2019). Deep multi-user reinforcement learning for dynamic spectrum access in multichannel wireless networks. *IEEE Transactions on Wireless Communications*, 18(1), 310–323.
- [42] Y. Huang, H. Jiang, H. Hu, and Y. Yao, “Design of learning engine based on Support Vector Machine in cognitive radio,” *Proc. Int. Conf. Computational Intelligence and Software Engineering*, Wuhan, 2009. [DOI: 10.1109/CISE.2009.5363127]
- [43] S. Tumuluru, P. Wang, and D. Niyato, “A Neural Network Based Spectrum Prediction Scheme for Cognitive Radio,” *IEEE ICC Workshops*, 2010.
- [44] M. Khozeimeh and S. Haykin, “Self-organising dynamic spectrum management technique for cognitive ad-hoc networks,” *Journal of Experimental & Theoretical Artificial Intelligence*, 2012.
- [45] Q. Yu, “A survey of cooperative games for cognitive radio networks,” *Wireless Personal Communications*, vol. 73, no. 3, pp. 949–966, 2013. [DOI: 10.1007/s11277-013-1225-6]
- [46] Q. He, Z. Feng, and P. Zhang, “Reconfiguration decision making based on Ant Colony optimization in cognitive radio network,” *Wireless Personal Communications*, vol. 71, no. 2, pp. 1247–1269, 2013. [DOI: 10.1007/s11277-012-0872-3]
- [47] Y. Wang, Z. Zhang, L. Ma, and J. Chen, “SVM-based spectrum mobility prediction scheme in mobile cognitive radio networks,” *The Scientific World Journal*, vol. 2014, Article ID 395212, 2014.
- [48] K. L. A. Yau et al., “Application of reinforcement learning in cognitive radio networks: Models and algorithms,” *The Science World Journal*, vol. 2014, Article ID 209810. [DOI: 10.1155/2014/209810]
- [49] C. Pham et al., “Spectrum handoff model based on Hidden Markov model in Cognitive Radio Networks,” *ICOIN*, 2014, pp. 406–411. [DOI: 10.1109/ICOIN.2014.6799714]
- [50] Y. Saleem and M. H. Rehmani, “Primary radio user activity models for cognitive radio networks: A survey,” *J. Netw. Comput. Appl.*, vol. 43, pp. 1–16, 2015.
- [51] E. Morabit, M. Benahmed, and M. Benbouzid, “GA-based Spectrum Decisions in Cognitive Radio Networks,” in *Proc. Int. Conf. Wireless Networks and Mobile Communications (WINCOM)*, 2015.
- [52] M. Saad, A. Abdel-Rahman, and M. Khattab, “HMM for Industrial Spectrum Prediction,” in *Proc. IEEE Int. Conf. Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2016.
- [53] H. Tang and Y. Li, “Deep Recurrent Neural Network for Frequency Predictions,” *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 2, pp. 1067–1084, 2017.

- [54] M. Selim, R. D. Gitlin, and H. Arslan, "Spectrum monitoring for radar bands using deep CNN," arXiv preprint, arXiv:1705.00462, 2018.
- [55] H. Lee, B. Shim, J. W. Chong, and J. Choi, "CNN-based cooperative spectrum sensing," in Proc. Int. Conf. Ubiquitous and Future Networks (ICUFN), 2019, pp. 720–722.
- [56] M. A. Akbar, S. S. Qazi, M. A. Imran, and S. A. Khan, "Markov Based Channel Prediction Algorithm for CR Networks," in Proc. Int. Conf. Wireless and Mobile Communications, 2007.
- [57] S. Senthilkumar and C. Geetha Priya, "Hidden Markov model based channel selection framework for cognitive radio network," Comput. Electr. Eng., vol. 65, pp. 516–526, 2016. [DOI: 10.1016/j.compeleceng.2016.06.016]
- [58] K. Koushik, A. Jain, and R. A. Buguda, "Hardware Implementation of RL and Transfer Learning for Spectrum Handoff," in Proc. Int. Conf. Advances in Computing, Communications and Informatics (ICACCI), 2018, pp. 961–967.
- [59] V. MohanaPriya and G. Shalinie, "Energy-Efficient RBM-Based Secure Routing in CR Networks," Wireless Pers. Commun., vol. 97, pp. 2777–2799, 2017.
- [60] C. Salgado, L. F. M. Vieira, and M. A. M. Vieira, "Intelligent algorithm for spectrum mobility in cognitive wireless networks," Procedia Comput. Sci., vol. 83, pp. 278–283, 2016. [DOI: 10.1016/j.procs.2016.04.126]
- [61] Q. Wang, K. Ren, and P. Ning, "Anti-jamming communication in cognitive radio networks with unknown channel statistics," in Proc. IEEE 19th Int. Conf. Netw. Protocols (ICNP), 2011, pp. 393–402.
- [62] C. Chen, M. Song, C. Xin, and J. Backens, "A game-theoretical anti-jamming scheme for cognitive radio networks," IEEE Network, vol. 27, no. 3, pp. 22–27, 2013.
- [63] D. Yang et al., "Coping with a smart jammer in wireless networks: A Stackelberg game approach," IEEE Transactions on Wireless Communications, vol. 12, no. 8, pp. 4038–4047, 2013.
- [64] L. Xiao et al., "Power control with reinforcement learning in cooperative cognitive radio networks against jamming," The Journal of Supercomputing, vol. 71, no. 9, pp. 3237–3257, 2015.
- [65] M. K. Hanawal, M. J. Abdel-Rahman, and M. Krunz, "Joint adaptation of frequency hopping and transmission rate for anti-jamming wireless systems," IEEE Transactions on Mobile Computing, vol. 15, no. 9, pp. 2247–2259, 2015.
- [66] L. Jia, H. Zhu, Z. Zhang, X. Guan, and H. Chen, "A game-theoretic learning approach for anti-jamming dynamic spectrum access in dense wireless networks," IEEE Transactions on Vehicular Technology, vol. 68, no. 2, pp. 1646–1656, Feb. 2019.

- [67] S. Misra, R. Singh, A. Vaish, and P. K. Jana, "M-JAW: Mobility-based jamming avoidance in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5381–5390, May 2020.
- [68] M. A. Aref and S. K. Jayaweera, "A cognitive anti-jamming and interference-avoidance stochastic game," in *Proc. IEEE 16th Int. Conf. Cognitive Informatics & Cognitive Computing (ICCIIC)\**, Jul. 2017, pp. 520–527.
- [69] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 310–323, Jan. 2019.
- [70] X. Liu et al., "Anti-jamming communications using spectrum waterfall: A deep reinforcement learning approach," *IEEE Communications Letters*, vol. 22, no. 5, pp. 998–1001, May 2018.
- [71] Y. Li, Y. Shi, and K. B. Letaief, "On the performance of deep reinforcement learning-based anti-jamming method confronting intelligent jammer," *Applied Sciences*, vol. 9, no. 7, p. 1361, 2019.
- [72] X. Liu et al., "A heterogeneous information fusion deep reinforcement learning for intelligent frequency selection of HF communication," *China Communications*, vol. 15, no. 9, pp. 73–84, 2018.
- [73] L. Xiao et al., "Two-dimensional antijamming mobile communication based on reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9499–9512, Oct. 2018.
- [74] Y. Yuan, L. Mou, and X. Lu, "Scene recognition by manifold regularized deep learning architecture," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 10, pp. 2222–2233, Oct. 2015.
- [75] L. Zhang, H. Gao, Y. Chen, and J. Zhang, "Deep reinforcement learning-based modulation and coding scheme selection in cognitive heterogeneous networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 6, pp. 3281–3294, Jun. 2019.
- [76] Z. Zhang, Y. Liu, W. Wu, and H. Zhang, "Intelligent anti-jamming relay communication system based on reinforcement learning," in *Proc. 2nd Int. Conf. Commun. Eng. Technol. (ICCET)*, Nagoya, Japan, Oct. 2019.
- [77] F. Slimeni, B. Scheers, Z. Chtourou, and V. Le Nir, "Cooperative Q-learning based channel selection for cognitive radio networks," *Wireless Networks*, vol. 24, no. 7, pp. 4161–4171, 2019.
- [78] A. Das, M. S. Pathan, and C. S. Hong, "Q-learning based co-operative spectrum mobility in cognitive radio networks," in *Proc. IEEE 42nd Conf. Local Computer Networks (LCN)*, Singapore, Oct. 2017.
- [79] B. Zhang, Y. Wu, X. Yi, and X. Yang, "Joint communication-motion-antenna programming in UAV swarms against jamming via multi-dimensional deep

reinforcement learning,” IEEE Access, vol. 7, pp. 180542–180563, 2019. [DOI: 10.1109/ACCESS.2019.2958965]

[80] W. Zhang, W. Ding, and C. Liu, “A channel selection and power control method of UAV data link,” Journal of Beijing University of Aeronautics and Astronautics, vol. 43, no. 3, pp. 583–591, 2017.

[81] X. Wang, M. Lei, M. Zhao, and M. Li, “Cooperative anti-jamming strategy and outage probability optimization for multi-hop ad-hoc networks,” in Proc. IEEE 86th Vehicular Technology Conference (VTC-Fall), 2017.

[82] A. Li and W. Zhang, “Mobile jammer-aided secure UAV communications via trajectory design and power control,” China Communications, vol. 15, no. 8, pp. 151–161, 2018.

[83] X. Liu, Y. Xu, L. Jia, Q. Wu, and A. Anpalagan, “Anti-jamming communications using spectrum waterfall: A deep reinforcement learning approach,” IEEE Communications Letters, vol. 22, no. 5, pp. 998–1001, May 2018.

[84] L. Xiao, D. Jiang, D. Xu, H. Zhu, Y. Zhang, and H. V. Poor, “Two-dimensional antijamming mobile communication based on reinforcement learning,” IEEE Transactions on Vehicular Technology, vol. 67, no. 10, pp. 9499–9512, Oct. 2018.

[85] X. Liu, Y. Xu, Y. Cheng, Y. Li, L. Zhao, and X. Zhang, “A heterogeneous information fusion deep reinforcement learning for intelligent frequency selection of HF communication,” China Communications, vol. 15, no. 9, pp. 73–84, 2018.

[86] F. Yao and L. Jia, “A collaborative multi-agent reinforcement learning anti-jamming algorithm in wireless networks,” IEEE Wireless Communications Letters, vol. 8, no. 4, pp. 1024–1027, Aug. 2019.

[87] F. Slimeni, B. Scheers, Z. Chtourou, V. Le Nir, and R. Attia, “Cognitive radio jamming mitigation using Markov decision process and reinforcement learning,” Procedia Computer Science, vol. 73, pp. 199–208, Jan. 2015.

[88] F. Slimeni, B. Scheers, Z. Chtourou, and V. Le Nir, “Jamming mitigation in cognitive radio networks using a modified Q-learning algorithm,” in Proc. Int. Conf. Military Commun. and Information Systems (ICMCIS), 2015, pp. 1–7.

[89] F. Slimeni, Z. Chtourou, B. Scheers, V. Le Nir, and R. Attia, “Cooperative Q-learning based channel selection for cognitive radio networks,” Wireless Networks, vol. 25, pp. 4161–4171, Oct. 2018.

[90] G. Han, L. Xiao, and H. V. Poor, “Two-dimensional anti-jamming communication based on deep reinforcement learning,” in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Mar. 2017, pp. 2087–2091.

[91] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... & Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

- [92] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, Stockholm, Sweden, Jul. 2018, pp. 1587–1596.
- [93] Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning (ICML)*, 1861–1870.
- [94] M. Kofman and R. Lee, *Russian Performance in the Russo-Ukrainian War: A Preliminary Analysis*. Arlington, VA: CNA, 2022.
- [95] J. Watling and J. Bronk, *The Russian Air War and Ukrainian Requirements for Air Defence*. London: Royal United Services Institute (RUSI), 2022.
- [96] Proakis J. G., Salehi M., (2008), *Digital Communications*, 5th Edition, McGraw-Hill.
- [97] Goldsmith A., (2005), *Wireless Communications*, Cambridge University Press.
- [98] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for dynamic spectrum access in multichannel wireless networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 310-323, Jan. 2019.
- [99] T. S. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed., Prentice Hall, 2002.
- [100] A. Goldsmith, *Wireless Communications*, Cambridge University Press, 2005.
- [101] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.
- [102] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, 2018, pp. 1587–1596. [Online]. Available: arXiv:1802.09477.
- [103] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 1861–1870. [Online]. Available: arXiv:1801.01290.

## BIOGRAPHY

Ömer Faruk DİKMEN graduated from Istanbul Technical University with a degree in Astronautical Engineering in 1996. He later pursued an MBA in Business Administration at Istanbul University. From 1997 to 2002, he held mid-level managerial positions in various corporate companies. In 2002, he founded his own company and engaged in international trade, acquiring several overseas distributorships and managing operations until 2022. Following the global pandemic, he decided to return to academia and began his Master's studies in Computer Engineering at Gebze Technical University in early 2023.



## **PUBLICATIONS AND PRESENTATIONS FROM THE THESIS**

Dikmen, Ö. F., Çelebi, H. (2025). “An Intelligent Jamming Environment: Reinforcement Learning Approach”, 4th International Graduate Research Symposium (IGRS'25), May 14, 2025.

