ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

A SMART INTRUSION DETECTION SYSTEM AGAINST CYBER ATTACKS IN UNMANNED AERIAL VEHICLES

Ph.D. THESIS

Burcu SÖNMEZ SARIKAYA

Department of Computer Engineering

Computer Engineering Programme

JULY 2025

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

A SMART INTRUSION DETECTION SYSTEM AGAINST CYBER ATTACKS IN UNMANNED AERIAL VEHICLES

Ph.D. THESIS

Burcu SÖNMEZ SARIKAYA
(504182515)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Prof. Dr. Şerif BAHTİYAR

JULY 2025

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

İNSANSIZ HAVA ARAÇLARINDA SİBER SALDIRILARA KARŞI AKILLI SALDIRI TESPİT SİSTEMİ

DOKTORA TEZİ

Burcu SÖNMEZ SARIKAYA
(504182515)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Şerif BAHTİYAR

TEMMUZ 2025

Burcu SÖNMEZ SARIKAYA, a Ph.D. student of ITU Graduate School student ID 504182515 successfully defended the thesis entitled "A SMART INTRUSION DETECTION SYSTEM AGAINST CYBER ATTACKS IN UNMANNED AERIAL VEHICLES", which she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Şerif BAHTİYAR** Istanbul Technical University	.....
Jury Members :	**Assoc. Prof. Dr. Yusuf YASLAN** Istanbul Technical University	.....
	Prof. Dr. Muhammed Ali AYDIN Istanbul University Cerrahpaşa	.....
	Assoc. Prof. Dr. Gökhan SEÇİNTİ Istanbul Technical University	.....
	Asst. Prof. Dr. Özgür Can TURNA Istanbul University Cerrahpaşa	.....

Date of Submission : **10 June 2025**
Date of Defense : **29 July 2025**

To my spouse and my daughter,

FOREWORD

I would like to express my heartfelt gratitude to my supervisor, Prof. Dr. Şerif BAHTİYAR, for his invaluable guidance, mentorship, and unwavering support throughout this thesis and the entire doctoral journey. His profound knowledge, constructive feedback, and dedicated mentorship have not only shaped the quality of this work but also played a pivotal role in my academic and personal development. I am especially grateful for his continuous encouragement during challenging times, his efforts to introduce me to the essence of academic culture, and his positive communication, which made me appreciate my profession even more deeply. Thanks to his support, I was able to experience this demanding doctoral process as a joyful and fulfilling learning journey. The lessons I have learned from him will undoubtedly guide and inspire me as I continue on my academic path with enthusiasm and confidence.

I am also deeply grateful to the esteemed members of my thesis committee, Prof. Dr. Muhammed Ali AYDIN and Assoc. Prof. Dr. Yusuf YASLAN, for their generous time, thorough evaluation, and constructive suggestions that greatly enhanced the quality of this research.

This study was generously supported by Turkcell İletişim Hizmetleri A.Ş., to whom I extend my sincere appreciation.

I would also like to thank my fellow researchers at the Cyber Security and Privacy Research Lab at Istanbul Technical University for creating a positive and energetic work environment. Their support, collaboration, and friendship have been invaluable throughout this journey.

I am deeply grateful to my dear friends, Handan, Buket, and Büşra, who encouraged me to embark on this doctoral journey and continually motivated me throughout. I sincerely thank them for their support and friendship throughout my life.

Finally, I would like to thank my beloved family for their endless love, support, and understanding. Their presence has been my greatest source of strength throughout this journey.

July 2025

Burcu SÖNMEZ SARIKAYA
(Computer Engineer, M.Sc.)

TABLE OF CONTENTS

ABBREVIATIONS

ADR	: Attack Detection Rate
AI	: Artificial Intelligence
CTGAN	: Conditional Tabular Generative Adversarial Network
DQN	: Deep Q-Network
DDPM	: Denoising Diffusion Probabilistic Model
DRL	: Deep Reinforcement Learning
FAR	: False Alarm Rate
FP	: False Positive
FN	: False Negative
GCS	: Ground Control Station
GAN	: Generative Adversarial Network
GPS	: Global Positioning System
IDS	: Intrusion Detection System
IMU	: Inertial Measurement Unit
KL	: Kullback-Leibler
KS	: Kolmogorov-Smirnov
LSTM	: Long Short-Term Memory
MDP	: Markov Decision Process
RTH	: Return-To-Home
RL	: Reinforcement Learning
SDV	: Synthetic Data Vault
SEQ	: Packet Sequence (MAVLink)
SVM	: Support Vector Machine
TP	: True Positive
TN	: True Negative
UAV	: Unmanned Aerial Vehicle
VAE	: Variational Autoencoder
XGBoost	: Extreme Gradient Boosting

SYMBOLS

$\boldsymbol{x}$	: Real input data sample
$\boldsymbol{z}$	: Latent noise vector or input for generative models
$\boldsymbol{G(z)}$	: Output of generator given noise input
$\boldsymbol{D(x)}$	: Discriminator output for input data x
$\boldsymbol{p_{data}}$	: Distribution of real data
$\boldsymbol{p_g}$	: Distribution of generated data
$\boldsymbol{P, Q}$	: Probability distributions used in divergence/distance calculations
$\boldsymbol{D_{KL}(P\|Q)}$	: Kullback-Leibler Divergence between P and Q
$\boldsymbol{W(P,Q)}$	: Wasserstein distance between distributions P and Q
$\boldsymbol{D}$	: Discriminator in GAN-based models
$\boldsymbol{G}$	: Generator in GAN-based models
$\boldsymbol{\gamma}$	: Joint distribution used in Wasserstein optimization
$\boldsymbol{\hat{x}}$	: Interpolated sample between real and generated data
$\boldsymbol{\lambda}$	: Gradient penalty coefficient
$\boldsymbol{\nabla_{\hat{x}} D(\hat{x})}$	: Gradient of discriminator wrt. input $\hat{x}$
$\boldsymbol{s_t}$	: State at time t in RL setup
$\boldsymbol{a_t}$	: Action taken at time t
$\boldsymbol{r_t}$	: Reward at time t
$\boldsymbol{Q(s,a)}$	: Q-value for state-action pair (s,a)
$\boldsymbol{\alpha}$	: Learning rate in RL or weighting factor in loss functions
$\boldsymbol{\gamma}$	: Discount factor in RL
$\boldsymbol{T}$	: Total number of timesteps in an episode
$\boldsymbol{\Delta_b(a_t)}$	: Battery cost of action a_t
$\boldsymbol{v_t}$	: UAV velocity at time t
$\boldsymbol{\alpha_t}$	: UAV altitude at time t
$\boldsymbol{p_j, p_s}$	: Probability of jamming and spoofing attacks
$\boldsymbol{g_t}$	: GPS integrity score at time t
$\boldsymbol{b_t}$	: Battery level at time t
$\boldsymbol{V(s)}$	: Value function for state s
$\boldsymbol{\mathscr{T}(s_t, a_t)}$	: State transition function given current state and action
$\mathscr{L}$	: Loss function

LIST OF TABLES

Page

LIST OF FIGURES

Page

A SMART INTRUSION DETECTION SYSTEM AGAINST CYBER ATTACKS IN UNMANNED AERIAL VEHICLES

SUMMARY

Unmanned Aerial Vehicles (UAVs) have become integral to numerous civilian and military operations due to their versatility and ability to operate in challenging environments. However, their reliance on communication networks, sensors, and software makes them susceptible to cyber-attacks, which can compromise security, safety, and mission objectives. Among the most significant threats are jamming and Global Positioning System (GPS) spoofing attacks, which can disrupt communications and manipulate navigation, potentially leading to system failure or hijacking. Protecting these systems requires robust security measures, particularly Intrusion Detection Systems (IDS).

Traditional IDS approaches for UAVs face several limitations. Many produce high false alarm rates, struggle with real-time detection, and cannot adapt quickly to evolving threats. Furthermore, UAVs have inherent constraints in power, computation, and storage, limiting the complexity of onboard security algorithms. A critical challenge is the scarcity of comprehensive datasets containing diverse attack scenarios, which is essential for training effective machine learning-based IDS. Training on imbalanced or incomplete data often leads to poor detection accuracy, especially for novel or rare attacks.

This thesis addresses these challenges by proposing a novel, hybrid Intrusion Detection System (IDS) framework for UAV security. This framework innovatively integrates Generative Artificial Intelligence (GenAI) for attack simulation and Deep Reinforcement Learning (DRL) for adaptive, real-time defense. The primary goal is to enhance IDS robustness against GPS spoofing and jamming attacks by tackling data scarcity and improving adaptability under dynamic threat conditions.

The first core component is a synthetic attack data generation module. This module utilizes a suite of advanced generative models: Variational Autoencoders (VAE), Gaussian Copula, Conditional Tabular Generative Adversarial Network (CTGAN), and Denoising Diffusion Probabilistic Model (DDPM). These models are employed to generate realistic and diverse GPS spoofing and jamming attack data based on real flight logs. This research systematically evaluates these models to determine their effectiveness in simulating attacks specific to UAV systems, a novel contribution compared to prior work focusing on single models or scenarios.

The generated synthetic data, alongside real data, is used in two ways. Firstly, it facilitates a comprehensive evaluation of traditional machine learning-based IDS, specifically using XGBoost, to understand their performance against both known and GenAI-generated attacks. A multi-train cross-testing strategy is employed to

assess the generalization capabilities of IDS trained on different combinations of real and synthetic data, revealing how synthetic data impacts performance under various conditions.

Secondly, and crucially, the augmented data enriches the training environment for the DRL-based adaptive detection module. This module features a Deep Q-Network (DQN) agent designed for real-time threat mitigation. By modeling UAV mission execution as a Markov Decision Process (MDP), the DQN agent learns optimal defense policies through interaction with telemetry states, which include both real and synthetic attack indicators. This integration of GenAI-based data augmentation and DRL-based adaptive defense represents a significant advancement in UAV security.

The DRL agent operates within a custom simulation environment. The state representation is multi-dimensional, capturing critical indicators such as GPS integrity, jamming and spoofing probabilities (derived from XGBoost outputs), motion dynamics, and battery level. The action space includes responses like alerting the Ground Control Station (GCS), switching to Inertial Measurement Unit (IMU) navigation, throttling communications, or initiating a Return-to-Home (RTH) procedure. The reward function is meticulously designed to encourage accurate attack identification, penalize false alarms, and promote energy-efficient decisions.

To evaluate the quality of the generated synthetic data, we employ a multi-dimensional framework combining statistical similarity metrics (Kullback–Leibler Divergence, Wasserstein Distance, Kolmogorov-Smirnov Statistic) with IDS classification performance. This provides a comprehensive assessment of both the fidelity and the practical impact of the synthetic data. For the DRL module, evaluation focuses on Attack Detection Rate (ADR), False Alarm Rate (FAR), resource efficiency (battery consumption, action costs), and learning stability (reward convergence, RTH activity, mission completion rate).

Our experimental results demonstrate the effectiveness of GenAI models, particularly DDPM and CTGAN, in generating high-fidelity synthetic attack data. These models produced data that closely matched the statistical properties of real attacks and proved most effective in challenging the XGBoost-based IDS, significantly degrading its performance compared to training on real data alone. This highlights both the potential of GenAI for IDS testing and the vulnerability of static IDSs to sophisticated, synthesized attacks.

The DRL-based IDS exhibited strong learning capabilities and effective real-time defense. The DQN agent achieved a high Attack Detection Rate (ADR) of approximately 95% during training, demonstrating its proficiency in identifying threats using learned patterns. It also learned to invoke safety protocols like RTH when necessary. However, the results also indicated a high False Alarm Rate (FAR), suggesting a tendency towards an aggressive defense policy that requires further tuning. Despite this, the agent showed good reward convergence and improved mission completion rates, indicating successful learning and a balance between security and operational continuity.

The discussion of our findings emphasizes that the fidelity of generative models is directly correlated with their ability to effectively test and potentially evade IDS.

DDPM emerged as the most potent model due to its ability to capture subtle anomalies in UAV data. We also address the practical deployability of the framework, noting that while GenAI training is computationally intensive, it is an offline process, and the deployed IDS (XGBoost or the DRL agent's underlying network) can be lightweight enough for resource-constrained UAVs.

However, the study acknowledges several limitations. The synthetic data may not capture all real-world complexities, the evaluation relies on a single dataset, and the generative models can be computationally demanding and lack interpretability. Furthermore, the framework currently assumes static adversaries, whereas real-world attackers adapt. Ethical considerations surrounding the dual-use nature of attack generation tools are also discussed, highlighting the need for responsible development and deployment while underscoring the benefits of using synthetic data for privacy-preserving research.

In conclusion, this thesis successfully demonstrates that GenAI models can create high-fidelity synthetic UAV attack data capable of challenging existing IDS. It introduces a hybrid framework combining GenAI and DRL to create a more robust, adaptive, and intelligent IDS for UAVs. The results confirm the potential of this approach to address data scarcity and improve detection capabilities against evolving cyber-physical threats.

Future work will focus on integrating the framework into adaptive IDS architectures with closed-loop feedback, allowing generative models to be refined based on detection outcomes. We also aim to explore model compression techniques for lighter deployment, enhance model interpretability, and incorporate mechanisms for continuous learning against adapting adversaries. Validating the system through field deployments on live UAVs is a key priority to assess its effectiveness under real-world operational constraints. This research paves the way for more secure and resilient UAV operations by leveraging the power of GenAI and DRL.

İNSANSIZ HAVA ARAÇLARINDA SİBER SALDIRILARA KARŞI AKILLI SALDIRI TESPİT SİSTEMİ

ÖZET

İnsansız Hava Araçları (İHA'lar), zorlu koşullarda çalışabilme yetenekleri ve çok yönlülükleri sayesinde birçok sivil ve askeri operasyonda vazgeçilmez hale gelmiştir. Ancak, bu sistemlerin iletişim ağlarına, sensörlere ve yazılımlara olan bağımlılıkları, onları siber saldırılara karşı savunmasız bırakmakta; bu durum güvenlik, emniyet ve görev başarısını ciddi şekilde riske atmaktadır. En önemli tehditler arasında, iletişimi kesintiye uğratan karıştırma (jammer) ve seyir sistemlerini manipüle eden Küresel Konumlama Sistemi (KKS) aldatma (spoofing) saldırıları yer almaktadır. Bu tür saldırılara karşı korunmak, özellikle Saldırı Tespit Sistemleri (STS) aracılığıyla sağlam güvenlik önlemleri gerektirir.

Geleneksel STS yaklaşımları, İHA'lar için bazı sınırlamalar taşımaktadır. Bu sistemler çoğu zaman yüksek yanlış alarm oranlarına sahiptir, gerçek zamanlı tespit konusunda yetersiz kalırlar ve gelişen tehditlere uyum sağlamakta zorlanırlar. Ayrıca, İHA'ların sınırlı güç, işlem ve depolama kapasitesi, karmaşık güvenlik algoritmalarının araç üzerinde çalıştırılmasını kısıtlamaktadır. Etkili makine öğrenmesi tabanlı STS'lerin eğitimi için gerekli olan çeşitli saldırı senaryolarını içeren kapsamlı veri kümelerinin eksikliği, önemli bir sorundur. Dengesiz veya eksik veriyle yapılan eğitim, özellikle yeni ve nadir saldırı türlerinin tespitinde düşük doğrulukla sonuçlanmaktadır.

Bu tezde, yukarıda belirtilen zorluklara çözüm getiren yenilikçi ve hibrit bir İHA güvenliği Saldırı Tespit Sistemi (STS) çerçevesi önerilmektedir. Bu çerçeve, saldırı simülasyonu için Üretici Yapay Zekâ (ÜYZ) ve uyarlanabilir, gerçek zamanlı savunma için Derin Pekiştirmeli Öğrenme (DPÖ) yöntemlerini bütünleştirerek önemli bir yenilik sunmaktadır. Temel hedef, KKS aldatma ve karıştırma saldırılarına karşı STS dayanıklılığını artırmak, veri eksikliğini gidermek ve değişken tehdit koşullarında sistemin uyum yeteneğini geliştirmektir.

Sistem iki temel bileşenden oluşmaktadır. İlk olarak, sentetik saldırı verisi üretim modülü yer almaktadır. Bu modül; Varyasyonel Otokodlayıcı (VO), Gauss Kopula, Koşullu Tablosal Üretici Rekabetçi Ağ (KT-ÜRA), ve Gürültü Giderici Yayılım Olasılık Modeli (GGYOM) gibi gelişmiş üretici modeller kullanarak, gerçek uçuş günlüklerine dayalı gerçekçi ve çeşitli KKS aldatma ve karıştırma saldırı verileri üretmektedir. Bu tezde, bu modeller sistematik biçimde değerlendirilerek, İHA sistemlerine özgü saldırı senaryolarının simülasyonu açısından etkililiği incelenmiştir; bu, önceki çalışmaların çoğunlukla tek model ya da tek senaryo odaklı olmasından farklı olarak önemli bir katkıdır.

Üretilen sentetik veriler, gerçek verilerle birlikte iki şekilde kullanılmaktadır. Birincisi, geleneksel makine öğrenmesi tabanlı STS'lerin performansını değerlendirmek

amacıyla, özellikle XGBoost algoritması kullanılarak gerçekleştirilmektedir. Farklı gerçek ve sentetik veri kombinasyonlarıyla çoklu eğitim ve çapraz test stratejisi uygulanmış; böylece STS'lerin genelleme yetenekleri çeşitli koşullar altında analiz edilmiştir.

İkinci ve daha önemli kullanım ise, zenginleştirilmiş verilerin DPÖ tabanlı uyarlanabilir tespit modülünün eğitim ortamına entegre edilmesidir. Bu modülde, bir Derin-Q Ağı (DQA) ajanı, gerçek zamanlı tehdit hafifletme amacıyla görev yapmaktadır. İHA görev yürütmesi, Markov Karar Süreci (MKS) olarak modellenmiş ve DQA ajanı, hem gerçek hem de sentetik saldırı göstergelerini içeren telemetri durumları üzerinden savunma politikalarını öğrenmiştir. Bu ÜYZ tabanlı veri artırımı ve DPÖ tabanlı adaptif savunmanın entegrasyonu, İHA güvenliğinde önemli bir ilerleme teşkil etmektedir.

DPÖ ajanı özel olarak tasarlanmış bir simülasyon ortamında çalışmaktadır. Durum temsili çok boyutludur ve GPS bütünlüğü, XGBoost çıktılarından türetilen karıştırma ve aldatma olasılıkları, hareket dinamikleri ve pil durumu gibi kritik göstergeleri içermektedir. Aksiyon alanı; Yer Kontrol İstasyonu'nu (YKİ) uyarma, Ataletsel Ölçüm Birimi (AÖB) navigasyonuna geçiş, iletişim kısıtlama veya Eve Dön (ED) protokolünün başlatılması gibi eylemleri içermektedir. Ödül fonksiyonu ise doğru saldırı tespitini teşvik ederken, yanlış alarmları cezalandırmakta ve enerji açısından verimli kararları desteklemektedir.

Üretilen sentetik verilerin kalitesini değerlendirmek için, istatistiksel benzerlik metrikleri (Kullback–Leibler Sapması, Wasserstein Mesafesi, Kolmogorov-Smirnov İstatistiği) ile STS sınıflandırma başarımı birleştirilerek çok boyutlu bir çerçeve kullanılmıştır. DPÖ modülü için ise saldırı tespit oranı (STO), yanlış alarm oranı (YAO), kaynak verimliliği (pil tüketimi, eylem maliyeti) ve öğrenme istikrarı (ödül yakınsaması, ED aktivitesi, görev tamamlama oranı) gibi kriterler temelinde değerlendirme yapılmıştır.

Deneysel sonuçlar, özellikle GGYOM ve KT-ÜRA modellerinin yüksek doğrulukta sentetik saldırı verisi üretebildiğini göstermiştir. Bu modeller, gerçek saldırıların istatistiksel özelliklerine yakın veriler üretmiş ve XGBoost tabanlı STS performansını ciddi ölçüde düşürerek gerçek veriye kıyasla daha zorlu testler sunmuştur. Bu, ÜYZ'nin STS testinde taşıdığı potansiyeli ve sabit yapılı STS'lerin sofistike sentetik saldırılara karşı ne kadar savunmasız olduğunu ortaya koymaktadır.

DPÖ tabanlı STS ise etkili öğrenme yeteneği ve başarılı gerçek zamanlı savunma performansı sergilemiştir. DQA ajanı, eğitim süresince yaklaşık %95'lik bir saldırı tespit oranı elde etmiş, böylece tehditleri öğrenilmiş desenler üzerinden başarıyla tanımlamıştır. Ayrıca gerektiğinde ED gibi güvenlik protokollerini devreye almayı öğrenmiştir. Bununla birlikte, yüksek bir yanlış alarm oranı gözlemlenmiş ve bu durum, savunma stratejisinin fazla agresif olduğunu ve daha fazla ayarlama gerektirdiğini ortaya koymuştur. Yine de, ödül yakınsaması ve artan görev tamamlama oranları, öğrenmenin başarılı olduğunu ve güvenlik ile operasyonel süreklilik arasında denge kurulduğunu göstermektedir.

Elde edilen bulgular, üretici modellerin doğruluğu ile STS'leri test etme ve aşma kabiliyetleri arasında doğrudan bir ilişki olduğunu ortaya koymuştur. GGYOM modeli,

İHA verilerindeki ince anomalileri yakalayabilmesi sayesinde en etkili model olarak öne çıkmıştır. Ayrıca, çerçevenin pratik uygulanabilirliği de ele alınmıştır: ÜYZ eğitimi yoğun hesaplama gerektirse de bu süreç çevrimdışıdır; dağıtımı yapılan STS (XGBoost veya DPÖ ajanı) ise kaynak kısıtlı İHA'lar için yeterince hafif yapıda tasarlanmıştır.

Bununla birlikte, bu çalışmanın bazı sınırlamaları mevcuttur. Sentetik veriler tüm gerçek dünya karmaşıklıklarını yansıtmayabilir; değerlendirmeler tek bir veri kümesine dayanmaktadır ve üretici modeller hesaplama açısından yoğun ve yorumlanabilirlik açısından sınırlı olabilir. Ayrıca, önerilen çerçeve şu an için sabit saldırgan varsayımına dayanmaktadır; oysa gerçek dünyada saldırganlar sürekli olarak stratejilerini değiştirir. Son olarak, saldırı üretim araçlarının çift kullanımlı (çift yönlü amaç taşıyan) doğasıyla ilgili etik konular da tartışılmış; sentetik verilerin gizliliği koruyan araştırmalar için taşıdığı faydalar vurgulanırken, sorumlu geliştirme ve uygulamanın önemi vurgulanmıştır.

Sonuç olarak, bu tez, ÜYZ modellerinin yüksek doğrulukta sentetik İHA saldırı verisi üretebileceğini ve mevcut STS sistemlerini zorlayabileceğini başarıyla göstermiştir. ÜYZ ve DPÖ bileşenlerinin entegrasyonuyla oluşturulan hibrit STS çerçevesi, İHA'lar için daha dayanıklı, uyarlanabilir ve akıllı güvenlik sistemlerinin geliştirilmesine katkı sağlamaktadır. Elde edilen sonuçlar, bu yaklaşımın veri eksikliği sorununu çözme ve gelişen siber-fiziksel tehditlere karşı tespit yeteneğini artırma potansiyelini doğrulamaktadır.

Gelecek çalışmalar, bu çerçevenin kapalı döngü geri besleme mekanizmalarına sahip uyarlanabilir STS mimarilerine entegre edilmesine odaklanacaktır. Böylece üretici modeller, tespit sonuçlarına göre dinamik olarak iyileştirilebilecektir. Ayrıca, daha hafif dağıtım için model sıkıştırma teknikleri, model yorumlanabilirliğinin artırılması ve adaptif saldırganlara karşı sürekli öğrenme mekanizmalarının entegrasyonu da hedeflenmektedir. Gerçek İHA'lar üzerinde yapılacak saha testleri ile sistemin pratik operasyonel koşullar altındaki etkinliği de değerlendirilecektir. Bu araştırma, ÜYZ ve DPÖ'nün gücünden faydalanarak daha güvenli ve dayanıklı İHA operasyonlarına kapı aralamaktadır.

1. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are aircraft that are remotely operated by human operators or autonomously controlled by onboard computers. They have become increasingly popular in various civilian and military applications due to their versatility, cost-effectiveness, and ability to operate in hazardous or inaccessible environments. As with any connected device, UAVs are vulnerable to cyber attacks that can compromise their functionality, data security, and safety. UAVs use a combination of communication networks, sensors, and software systems. UAVs collect and process data, control their flight, and carry out specific missions. These systems generate and exchange large amounts of sensitive information, such as location data, video footage, sensor data, and control commands. Attackers can exploit vulnerabilities in these systems to gain unauthorized access, manipulate data, disrupt communications, hijack systems' control, or cause physical damage. Thus, attacks may compromise the security goals of UAVs, which are known as confidentiality, integrity, and availability [1]–[3]. For instance, jamming attacks like Denial-of-Service (DoS) and Global Positioning System (GPS) spoofing, replay, routing, eavesdropping, and false data injection are such examples, which disrupt UAV availability by affecting communications [4]–[6].

UAVs interact with many networks, therefore, security solutions of the networks should be addressed to cope with the grand challenge. Emerging technologies such as blockchain, software-defined networks (SDN), machine learning, fog, and edge computing have been explored in [7,8] as part of the solution architectures. Considering the resource constraints of drones, some blockchain-based systems [9]–[11] may provide a solution instead of using all properties of heavy resource-intensive chains. SDN technology may ensure network reliability by allowing the controller to closely monitor data traffic in UAVs [12]. Fog computing may be used to maintain computing capabilities near drones without exceeding their capacity [13].

The communication requirements of UAVs depend on the environment, which determines their security requirements. The complex communication environment of UAVs extends the attack surface. Therefore, it is necessary to apply multiple security mechanisms simultaneously to ensure their security. This circumstance necessitates the design of new security systems. The emerging security requirements also increase the communication cost of UAVs. Thus, the main challenge is to create a new attack detection system that supports dynamic environmental conditions for UAVs.

Jamming and GPS spoofing attacks, in particular, have emerged as potent threats to UAV communication protocols such as MAVLink, leading to serious degradation in system performance. Jamming attacks are intentionally used to disrupt or block communication signals. Such attacks directly compromise system availability and pose serious challenges to UAV security [14]. GPS spoofing attacks, which alter the trajectory of autonomous UAVs, represent another critical threat to UAV systems [15,16]. Security measures have been proposed to mitigate jamming and GPS spoofing threats, such as encryption, authentication protocols, and IDS. Specifically, Intrusion Detection System (IDS) plays an important role in protecting UAV systems by monitoring data traffic and identifying unauthorized or malicious activities. Machine learning-based IDSs have shown promising results in addressing signal spoofing and jamming attacks by analyzing data from UAV flight and sensor readings [17].

The emergence of Generative Artificial Intelligence (GenAI) introduces both opportunities and risks in UAV security. On one hand, GenAI models can generate realistic synthetic attack data. This synthetic attack data can be used to augment training datasets, improve generalization, and enhance IDS robustness against previously unseen attack patterns. On the other hand, these same models can be leveraged by adversaries to craft highly deceptive, undetectable attacks, emphasizing the need for proactive and adaptive detection mechanisms.

Existing IDS countermeasures for UAVs have several limitations, which prevent to have secure UAV systems. For example, many IDS systems produce excessive false alarms, leading to unnecessary operational disruptions. IDS systems may struggle to provide real-time threat and GenAI-based attack detection. On the other hand, UAV threats evolve rapidly, and existing IDS systems often fail to adapt to new and unknown attack vectors. UAVs typically operate with a limited power supply and a communication bandwidth, which may restrict the deployment of advanced IDS algorithms. Moreover, UAVs have constrained computational and storage capabilities that limit IDSs from implementing complex algorithms onboard.

Training IDS systems, particularly those based on machine learning, requires extensive data that includes diverse attack scenarios. Such data are often unavailable or incomplete for UAV environments. Training on imbalanced datasets with significantly fewer attack samples than normal operations leads to a poor detection accuracy for rare threats. These highlight the need for more robust, adaptable, and resource-efficient systems that may address the diverse and evolving threat landscape.

Reinforcement learning (RL) is a subfield of machine learning that brings a reward-driven behavior from psychology to artificial intelligence that creates a powerful alternative to regular controllers [18]. RL stands out from usual control methods since it only requires a goal definition to work and no prior knowledge about the environment or system is required. Actions to reach the goal are chosen by RL algorithms based on positive or negative outcomes of previous interactions with the environment.

RL maps an appropriate action to each state. After performing the assigned action, positive or negative feedback increases or decreases the chance of the action being chosen again. Updating mappings iteratively, RL finds best actions for each state to reach the goal as soon as possible. Exploitation of experiences to select better actions among candidates eliminates the requirement of the system or environment models that are given in advance [19].

Deep reinforcement learning (DRL) is a promising technique that will enhance the security of UAVs and communication networks. DRL helps UAVs to learn from experiences and make decisions based on environmental feedback. On the other hand, DRL poses several technical and ethical challenges that need to be addressed. Interpretation of learned policies, data privacy, and the potential for adversarial attacks are possible [20]. For example, several RL applications require data accesses, sensing the environment and collecting data, resource allocations for wireless connectivity, UAV-enabled mobile edge computing data, localization data, trajectory planning, and network security in multi-UAV wireless networks. Nevertheless, the absence of specific security models that support RL-based solutions for UAV security requires further research.

In this context, the integration of generative models with DRL-based detection architectures presents a promising direction for building next-generation UAV security systems. Building on these advancements, this thesis proposes a generative intelligence-driven intrusion detection framework for UAV security, designed to integrate synthetic attack data generation with deep reinforcement learning to enhance detection robustness. The framework comprises two core modules: 1) a synthetic attack data generation module utilizing advanced generative models namely Variational Autoencoders (VAE), Gaussian Copula, Conditional Tabular Generative Adversarial Network(CTGAN), and Denoising Diffusion Probabilistic Model (DDPM) to augment IDS training with diverse and realistic threat scenarios, and 2) a DRL-based adaptive detection module that enables real-time decision-making through continuous environmental interaction. By combining offline generative training with online reinforcement-driven adaptation, the proposed hybrid framework delivers scalable, resource-aware, and resilient protection against both conventional and emerging cyber-physical attacks in autonomous UAV systems.

The framework comprises three interrelated modules: 1) a generative module that leverages state-of-the-art models such as Variational Autoencoders (VAE), Gaussian Copula, Conditional Tabular Generative Adversarial Network(CTGAN), and Denoising Diffusion Probabilistic Model (DDPM) to generate diverse and realistic synthetic attack data, 2)A lightweight traditional IDS module based on tree-based classifiers (e.g. XGBoost) to evaluate GenAI-based attacks, 3) a DRL-based IDS module that enables real-time, adaptive defense through continuous learning from environmental feedback. By synthesizing these components, the proposed hybrid framework addresses both data scarcity and dynamic threat adaptation, offering a scalable, resource-efficient, and resilient solution for securing autonomous UAV systems against evolving cyber-physical attacks.

1.1 Purpose of the Thesis

The primary purpose of this thesis is to design, implement, and evaluate a hybrid intrusion detection system (IDS) for UAV security that integrates generative AI-based attack simulation, conventional machine learning classification, and deep reinforcement learning (DRL) for adaptive defense. By combining synthetic data generation using VAE, Gaussian Copula, CTGAN, and DDPM with a lightweight XGBoost-based classifier and a Deep Q-Network (DQN) agent, the framework aims to address key limitations in current UAV IDS systems—namely, data scarcity, static detection capabilities, dynamic threats, limited energy/battery, UAV action costs, and limited real-time adaptability. The proposed system is designed to operate under dynamic threat conditions, enabling UAVs to detect and respond to both real and synthetic GPS spoofing and jamming attacks in an energy-aware, and resource-efficient manner. We will pursue research to answer the following questions:

1. **How can generative AI techniques (VAE, Gaussian Copula, CTGAN, DDPM) be leveraged to address data scarcity and augment training for GPS spoofing and jamming detection in UAVs?**

   Expected research outcomes:

   (a) Generation of augmented and realistic attack data to enrich IDS training in low-data regimes.

   (b) Comparative evaluation of generative models for attack data fidelity and diversity.

   (c) Multi-dimensional assessment of generated data, including statistical similarity (KL Divergence, Wasserstein Distance, KS Statistic) and performance-based validation through cross-testing with traditional IDS classifiers.

2. **How accurately can a traditional machine learning-based IDS, such as XGBoost, detect both real and GenAI-generated attacks, and what limitations emerge under evolving adversarial conditions?**

   Expected research outcomes:

   (a) Comparative performance analysis of IDS trained with real, synthetic, and hybrid datasets.

   (b) Evaluation of False Alarm Rate (FAR) and detection reliability under varied attack distributions.

3. **How can a Deep Q-Network-based IDS be designed to operate under limited UAV resources while ensuring adaptability, attack resilience, and mission assurance?**

   Expected research outcomes:

   (a) Design of a resource-aware DQN-based IDS with energy-efficient state transitions.

   (b) Evaluation of mission-critical metrics such as mission complete rate and Return to Home Activity (RTH).

(c) Evaluation of attack detection metrics such as attack detection rate and false alarm rate.

(d) Design of energy-efficient action selection under constrained UAV resources.

(e) Design of multi-objective reward design under real and synthetic attacks.

4. **What is the impact of integrating synthetic attack data with reinforcement learning on the learning stability, reward convergence, policy entropy, and generalization performance of the DQN agent?**

   Expected research outcomes:

   (a) Improved generalization of DQN under unseen threat scenarios.

   (b) Analysis of training stability through reward convergence curves.

   (c) Quantitative evaluation of policy entropy and exploration–exploitation balance.

5. **How does the proposed hybrid GenAI + DRL IDS framework perform compared to conventional IDS solutions in terms of detection accuracy, energy efficiency, and operational resilience in UAV systems?**

   Expected research outcomes:

   (a) Comprehensive benchmarking across ADR, FAR, resource efficiency, and RTH.

   (b) Validation of hybrid framework under real-world and synthetic scenarios.

   (c) Analysis of benefits in resource-constrained and mission-critical UAV operations.

Contributions of this thesis include:

1. **Development of a multi-model GenAI-based synthetic attack generation framework**, specifically designed for GPS spoofing and jamming scenarios in UAV systems.

2. **Generation of augmented, realistic, and statistically validated adversarial attack data** to mitigate data scarcity and enhance the robustness of intrusion detection systems.

3. **Establishment of a controlled adversarial testing environment**, enabling systematic evaluation of DQN-based IDS sensitivity to adversarial threats and supporting the design of more resilient detection mechanisms.

4. **Design of a resource-aware, mission-conscious DQN-based IDS**, integrated with GenAI-enhanced training data to support adaptive, context-aware, and robust threat response in real-time UAV operations.

5. **Proposal of a hybrid evaluation protocol and metric design**, enabling comprehensive assessment of detection accuracy, energy efficiency, mission continuity, and overall system resilience under dynamic adversarial conditions.

6. **Design and evaluation of novel hybrid IDS architectures**, incorporating mission-aware decision logic and energy-efficient policies to ensure operational sustainability and threat adaptability.

In addition, this thesis highlights the critical role of integrating Generative AI and reinforcement learning to address UAV-specific constraints such as limited computational capacity, dynamic threat conditions, and the need for rapid decision-making. By enabling both data augmentation through synthetic attack generation and adaptive defense through real-time policy learning, the proposed approach bridges the gap between offline threat modeling and online operational resilience in UAV security systems.

1.2 Thesis Structure

The organization of this thesis is outlined as follows: In Chapter 2, we review the related works in the field of UAV security, with a particular focus on recent advances in generative artificial intelligence, DRL-based intrusion detection systems for attack data generation, and detection of attacks in UAVs. Chapter 3 provides an overview of an effective GenAI framework for generating realistic attack data in UAVs. Chapter 4 presents a detailed examination of the DRL-based intrusion detection system framework for UAVs under real and synthetic attacks, which aims to improve resource efficiency, robustness, energy efficiency, adaptability. In Chapter 5, describes the experimental setup and methodologies used to test our framework. Chapter 6 presents the results obtained from the experiments. Chapter 7 explores the implications of our findings and observations while suggesting potential directions for future research. Finally, Chapter 8 provides a comprehensive summary of our contributions to the field and discusses them in depth.

2. LITERATURE REVIEW

The growing reliance on Unmanned Aerial Vehicles (UAVs) in critical civilian and military operations has prompted increasing research attention toward securing UAV communication and control systems against cyber-physical threats.

At the same time, the emergence of Generative Artificial Intelligence (GenAI) models has introduced new opportunities for simulating adversarial conditions under data-scarce environments. These models have been explored in recent work for augmenting IDS training with synthetic data, yet their use in UAV-specific intrusion scenarios remains relatively underexplored. Parallel to this, Reinforcement Learning (RL), especially in the form of Deep Q-Learning (DQN), has gained traction as a promising paradigm for adaptive intrusion detection in dynamic environments.

This section reviews existing research across three key areas: (1) UAV and Security Concerns, (2)generative model-based attack simulation for UAV, and (3) reinforcement learning-based adaptive IDS architectures. Through this synthesis, we aim to identify critical research gaps and motivate the design of a hybrid GenAI+DRL framework for UAV security.

2.1 UAV and Security

2.1.1 UAV systems and communications

The development of UAV technology has led to the creation of different types of drones with different shapes and weights. This section examines the general system of UAVs and their communication infrastructures that are vulnerable to cyber-attack. The investigation focuses on the high-level structure and fundamental components of a UAV system. These include the communications infrastructure of UAVs, the Ground Control Station (GCS), and satellite communications technologies.

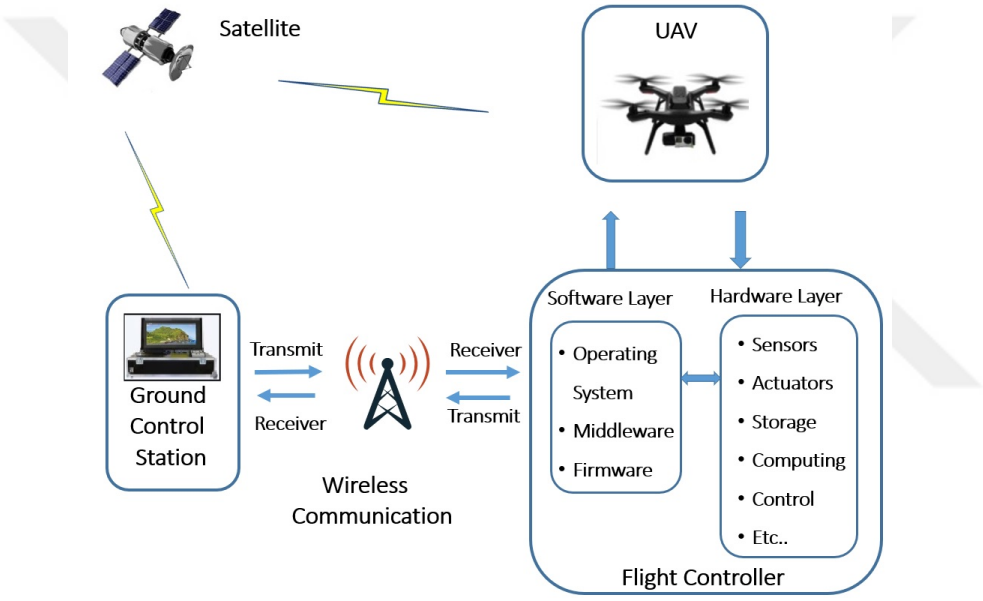

Figure 2.1 : General architecture of an UAV system.

In general, an UAV system consists of an unmanned aircraft, a ground control station and a communications data link [21,22]. The unmanned aircraft represents the core of an unmanned aerial system and consists of an airframe, a propulsion system, a flight controller, a precision navigation system, and a sense-and-avoid system [23]. In this research, we consider security of UAVs so we consider only the corresponding building blocks that will create a vulnerability in the system. For instance, an aircraft's flight controller, communication lines, and sensors are blocks of UAVs that may contain vulnerabilities. The high-level architecture of a UAV system and its main elements are shown in Figure 2.1.

The flight controller is the central processing unit of a drone that is responsible for stabilizing the aircraft during flight and collecting data from sensors. At the hardware level, the flight controller configuration includes rechargeable batteries, actuators, and various sensors, such as GPS and accelerators, as well as a wireless communication module. The software architecture consists of three layers, namely, firmware, middleware, and operating system. The firmware issues machine code instructions, the middleware manages communication between services and the operating system, which is often an RTOS that handles real-time data processing [24].

The flight controller makes communications with the ground control station easy. The controller processes commands and translates them into actions for actuators. Telemetric signals are transmitted to GCS via multiple channels. The flight controller may integrate sensors or may communicate with external units to enhance data acquisition capabilities, making it a critical component of the UAV system's distributed embedded architecture [23].

UAVs are not a fully self-acting artificial intelligence products, but they are a part of the whole of the communication infrastructure. For this reason, an UAV has to communicate with other UAVs and ground control stations to complete its mission. The communication lines in UAVs are shown in Figure 2.2. Two types of methods are used for the communication of UAVs [25]. The first method is to check the movement of an UAV with signal communication. The other is the transfer of data related to the task that the UAV should complete with data communication.

An UAV collects, processes, and transmits sensitive data according to its usage area. Therefore, the security of data that may be related to strategic operations, environmental surveillance, and communication is very important. The expanding application domains of UAVs, the huge communication options, and many implementations with different infrastructures, such as wireless sensor networks, mobile ad hoc networks (MANETs), Ad-Hoc Network (VANET), Flying Ad-Hoc Network (FANET), and GPS make UAVs open to security attacks [26]–[29]. In addition, UAVs are used to communicate with each other with less energy-consuming EMA-style protocols. Bluetooth, Zigbee, and WiFi protocols are used for short distances between UAVs and GCS. WiMAX and Cellular protocols are used for long distances. GPS coordinates are used for the communication of UAVs with satellites, usually WiMAX and Cellular communications [30].

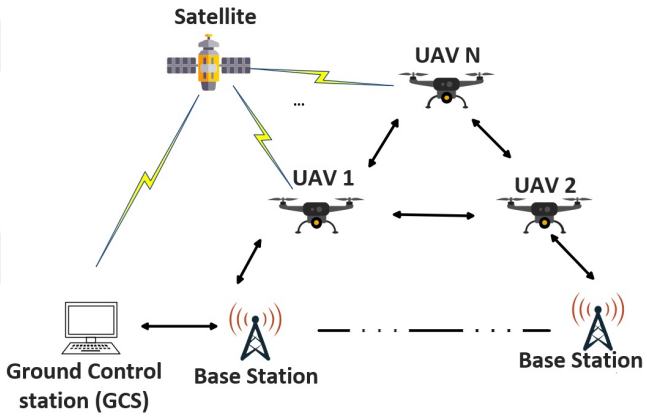

Figure 2.2 : The communication links of UAVs.

UAVs have different network technologies for communicating with multiple UAVs (UAV-to-UAV, U2U) and between UAVs and other systems (UAV-to-Infrastructure: U2I), such as GCS, WSN, and ground reactors as shown in Figure 2.2. Ad-hoc network is recommended as a good solution. [31] contains key issues related to UAV communication networks that include characteristics of existing ad hoc networks to classify UAV networks by topology. [32] focuses on communication between an UAV and a GCS. They are classified technologies for communication links, such as Zigbee wimax, IEEE 802.xx, by the range and the data rate. The most important problem in ensuring cooperation between Multi-UAVs is the communication infrastructure of UAVs that requires network supports and service components. A categorization of security risks and solutions related to FANETs and UAV communications is presented according to the four OSI layers in [29].

The most significant attack vector of UAVs is its communication. Specifically, it is critical for UAVs to secure wireless channels that are vulnerable to attacks. Obviously, UAVs communicate with each other and through the ground control station via wireless channels that are open to various attacks. In fact, launching an attack on UAVs is very easy. Special and critical UAV information can be accessed by unauthorized users. On the other hand, due to the lack of security mechanisms in the applications of satellite-connected UAVs, legal access to the essential services may be prevented, which is a kind of denial of service (DoS) attack. Thus, UAVs require a security mechanism to preserve the confidentiality, integrity, and availability of whole system in a UAV. Communications of UAVs are done by using data transfer or, more specifically, by using signal communications. The main security requirements for data transfer in a communication are integrity and confidentiality. The security requirement for signal processing is accessibility, which is sometimes known as availability.

Attacks on UAVs may endanger confidentiality, integrity, and availability of data [33] that are security requirements of the UAV system cyber security threat model [34]. There are also cyber-physical attacks on UAVs that have targets on the UAV network layers [35]. Attackers targeting confidentiality have unauthorized access to the system by capturing sensitive data or sensor data. To this end, the attacker can listen to the communication between the UAV and the GCS, then transmit data to other malicious actors. It may threaten security by collecting intelligence about the task of UAVs. Such attacks are referred as eavesdropping attacks in the literature [36].

Encryption methods are used to prevent attacks. For instance, data on wireless communication lines must be encrypted [37]. Attacks may disrupt the integrity of data in the communication network, which causes unauthorized modification of components and tasks of UAVs. Examples of such attacks targeting UAV data integrity are replay attacks [38]. To prevent these attacks, hash functions, such as a message authentication code, are used that check whether the data are corrupted.

UAVs are subject to different attacks according to changing environments, such as the network structure, coverage areas, and communication channels. Specifically, a UAV is open to different types of cyber attacks, such as jamming, spoofing, and replay attacks, due to the changing communication environments of the UAV and wireless channels. Malicious attacks may interrupt communications to reduce the availability of UAVs. Recently, jamming and GPS spoofing attacks on the wireless sensor network have been used to interrupt communications among UAVs. In order to prevent these attacks, UAVs need a variable and adaptive security mechanism. In particular, it is very important to ensure security in the communication area, which is critical to complete the task of UAVs. Therefore, jamming and GPS spoofing attacks, which consider the availability of UAVs and cause communication breakdowns, are investigated by many researchers [39].

In order to provide security against attacks on UAVs, an independent method of the environment is necessary to quickly detect attacks. At this point, deep-reinforcement methods provide remarkable options. DRL is suitable and the fastest route for an UAV that provides an independent environment for self-adaptation against attacks. An important summary of deep reinforcement applications in communication networks is given in [40]. Thus, understanding and optimizing UAV communications systems is crucial for achieving secure operations with UAVs in a variety of applications.

2.1.2 Security requirements of UAV

UAVs are becoming increasingly important in various industries, including military, agriculture, research, etc. UAVs must have strong security because they process sensitive data or operate in mission-critical infrastructures. The basic security requirements for UAVs are authentication, integrity, confidentiality, availability, identity anonymity, and self-stabilization.

Authentication is very important for all nodes in the infrastructure and messages passing through the UAV network. This requirement ensures that only authenticated nodes may participate in the routing process. An attacker can spoof a legal node, gain access to confidential information, and even interfere with network operation without authentication [41]–[43].

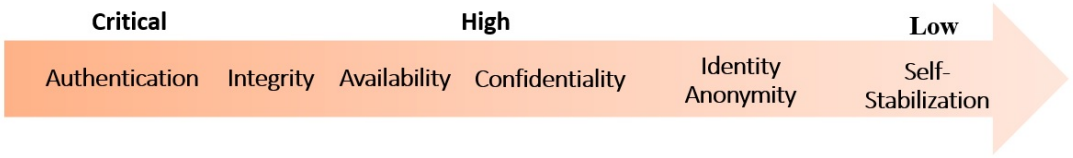

Figure 2.3 : Security requirements for UAVs.

Integrity includes the consistency, accuracy, and reliability of data packets. It ensures that the attacker does not alter data on the traffic or GPS coordinates by adding, deleting, or modifying the data transmission. If the integrity mechanism of UAV network is weak, the integrity of the entire UAVs may be at risk [41]–[43].

Confidentiality is the measure used to prevent sensitive information from being visible to a wrong node or malicious node. It assures that data is visible only to the right node. Confidentiality ensures that UAV network, payload traffic, command and control traffic, and sensitive information are not visible to unauthorized nodes. In the absence of a privacy mechanism, an attacker can launch attacks by improperly collecting sensitive information about the target [41]–[44].

Availability ensures that all services provided by an UAV system are always available to authorized entities or devices within an UAV system, even at the time of the attack. It ensures that the network has functional and useful information, such as command and traffic control during UAV network activity. In reality, this is difficult to achieve due to the limitations in delays and the critical nature of an UAV mission [41]–[43].

Identity anonymity ensures that an attacker cannot obtain the true identity of users from intercepted messages in the communication of an UAV network [42]. This requirement may be implemented as a part of authentication system within an UAV system. Both user authentication and packet authentication may be needed to be implemented for UAVs that have critical missions.

Self-stabilization in UAV network communication protocols must be able to automatically recover from any attack without the need of human intervention. This requirement ensures that malicious packets do not permanently damage the network [42]. DRL is a good candidate to provide self-stabilization requirement for UAVs.

The security requirements for UAVs contain specific challenges to be implemented, as the vulnerabilities associated with UAV operations are diverse. It is critical to ensure that only authorized people can access and control UAVs. Authentication mechanisms must be robust to prevent unauthorized access. The integrity of data and instructions transmitted between the ground control station and the UAVs is very critical. Interference with commands or data must be avoided. Availability for UAVs is also critical, which prevents DoS attacks. The prioritization of requirements can vary depending on specific use cases, regulations, and the operational context of the UAV. Regular risk assessments and updates on security measures are essential to counteract evolving threats. In general, prioritized security requirements of UAVs are shown in Figure 2.3.

2.2 Threats and Attacks on UAV

2.2.1 Security threats

While unmanned aerial vehicles offer innovative capabilities, they are prone to vulnerabilities. Understanding these vulnerabilities is critical to ensuring the security and reliability of UAV operations. This section provides an in-depth analysis of various threats and corresponding vulnerabilities in UAVs.

Wireless Link in UAV networks uses wireless connections to send and receive radio signals. In general, anyone can listen a frequency and receive signals with antennas configured for specific frequencies, such as GPS signals sent by GCS and UAVs. Specifically, GCS gives UAVs real-time control over the communications link to a satellite (uplink) and a command traffic. It is sufficient for attackers to simply detect the signal and generate a noise signal in radio communications. [45] contains an overview of applying machine learning techniques to tackle fundamental challenges in wireless communications within the Internet of Things (IoT) with the specific focus on the ad-hoc networking dimension.

A Wi-fi-based communication may have vulnerabilities. If a Wi-fi-based attack is carried out successfully and the control station does not react quickly, the UAV will not be able to receive any command for a certain period of time. This may cause an attacker to hijack an UAV. Due to data connectivity capabilities, wireless connections of UAVs often have lower bandwidth than wired networks. An attacker can exploit this feature by sending fake packets to UAVs. These packets may consume bandwidth and interfere with normal communication [46].

The use of radio transmission combined with a small size, a low cost, and a limited power makes the wireless link more susceptible to denial of service attacks. Wireless connectivity allows attackers to actively or passively eavesdrop on communication. Attacks on a wireless connection can come from all directions and the node can be captured, which may cause leakage of sensitive information [47].

Uncontrolled Environment in UAV networks does not have a central authority such as a switch management system to handle incoming and outgoing network packets, as in wired networks. Due to the lack of a key management system, inside and outside attacks on the network are possible. An attacker can send and receive data traffic while it is within the transmission range of an UAV [41].

Dynamic Topology is another problem that arises with an UAV network, where the environment prevents precisely to detect the enemy node. It is often difficult to distinguish between a malfunctioning node due to the dynamic topology of UAVs and a legal node that appears to be faulty because of the poor link quality. For instance, routing algorithms based on a topology prediction and a self-adaptive learning, which are specifically designed to adapt to the dynamic topology of the network, have been investigated in [48]. Future trends in security and privacy are discussed in addition to routing, connectivity, topology control, and energy efficiency. The difference in routing cost in both cases can be quite small. Therefore, such attacks are difficult to resolve [49].

Cooperation of routing algorithms for an UAV network require all nodes to participate in the discovery of the topology and the transmission of data. Security transactions between nodes are ignored. When a node searches for a route, it broadcasts the message regardless of the recipient's identity. Thus, an attacker could easily participate in the routing process and it can damage the UAV network topology by filtering or blocking the control traffic [41].

Limited Resources in UAVs represent limited information processing and storage capabilities, which are depending on the size of an UAV. By exploiting this vulnerability of the UAV, an attacker can launch an attack that aims to drain the UAV's resources, such as the battery. The implemented security solution directly dependents on the power and the storage capacity of UAVs. For example, the memory of an UAV must be large enough to hold all the variables needed to run asymmetric cryptographic algorithms. For low-power UAVs, this may be difficult because digital signature-based authentication methods require high computation power [50].

In the literature, there are solutions that consider vulnerabilities in different parts of UAV systems, such as routing, partition, connectivity, recovery, fault detection, dynamic mobility, unstable paths, power control, resource allocation, path planning, energy consumption, and communication overhead. For example, a k-means online learning routing protocol (KMORP) is specifically designed for ad hoc UAV networks with a Markov mobility model [51]. KMORP is particularly effective for specialized UAV networks as it swiftly adapts to dynamic changes in the network environment, such as UAV mobility, interference, and signal degradation. This approach guarantees optimal data transmissions and communications by expertly adapting to variations. KMORP utilizes clustering to reduce communication overheads between nodes, which help to alleviate the network's overall load.

An approach called Unmanned Aerial Vehicles (UAV)-assisted Network Segmentation Detection and Connection Restoration (UAV-NetRest) that covers all stages from a network fault detection to a partition resolution is presented in [52]. The speed of a fault detection is very important in order to avoid creating a security vulnerability. In the detection phase, relay node locations are divided and clusters are assigned to each UAV using k-means++ clustering. Depending on the number of failed nodes, UAV-NetRest dynamically adjusts the number of UAVs. In this way, network fault detection is performed quickly. The evaluation of the algorithm is based on the detection and the recovery time, the distance traveled by UAVs during the operation, the number of messages transmitted, and the deployment of the relay node.

AI-enabled routing protocols developed for UAV networks are used to detect and prevent threats [53]. For example, a topology predictive and self-adaptive learning-based routing algorithms to adjust the dynamic network topology are used in UAVs. Similarly, the cost of offloading for the user equipment (UE) and the energy efficiency of the UAV system is done by using Markov decision processes in [54]. It presents a Multi-Agent Reinforcement Deep Learning algorithm (MADRL) based on deep learning principles. MADRL aims to optimize power control, resource allocation, and UE association, so that it reduces energy consumption in the system.

The exponential growth of systems that use IoT devices like in IoT devices in health and UAV has led to use machine learning algorithms for better decision making. However, the security threads based on the usage of IoT devices remain [55]. Both IoT usage in health and in UAV have similar security threats. For instance, IoT devices are used in health systems as biomedical sensors, which have similar properties with sensors on UAV. Since sensors may build a wireless network, a secure wireless communication is essential [56]. Moreover, IoT based health and UAV systems may use machine learning algorithms for decision making, which algorithms are prone to data poisoning attacks. [57]. Furthermore, systematic poisoning attacks are possible for such critical systems [58].

Understanding and mitigating vulnerabilities require a holistic approach that combines technological innovation, security measures, and legal requirements. Ongoing research and developments are essential to stay ahead of emerging threats in the dynamic landscape of UAV technology.

2.2.2 Security attacks

UAV networks are exposed to different types of attacks. The goal of an attack to an UAV system is to absorb and control network traffic, interrupt the routing function, or inject malicious nodes. Various attacks in the environment of UAV networks have been described in the literature. For example, an overview of cyber attacks that may affect privacy, integrity, and availability of UAV systems is presented in [6]. Security of communication links between ground control station and drones is one of the main research focus [59]. Many articles in the literature focus on identifying security threats so corresponding attacks in drone communications and their corresponding countermeasures. The synthesis of security risks in various systems sets the stage for a discourse on key security challenges that occurred in unmanned systems [60], such as privacy and security challenges in the Internet of Drones (IoD).

A man-in-the-middle attack against a typical UAV used for critical applications is a common example for attacks in UAV systems [5]. GPS spoofing attacks to manipulate the trajectory of an autonomous UAV is another significant attack on UAV systems [16,61,62]. An attacker can compromise protocols using various attack strategies. Attacks and countermeasures for security protocols are highlighted, and a schematic diagram of possible attacks is provided in [63]. In general, when we evaluate all researches together in the literature, main attacks carried out on UAVs are GPS Spoofing Attacks, False Data Injection, Jamming Attacks, Routing Attacks, Eavesdropping, Side-Channel Attacks, Code Modification, Code Injection, Packet Sniffing, Replay Attacks.

False Data Injection Attacks: False data injection is a technique where an unauthorized person sends copied data to the UAV to take it control as shown in Figure 2.4. UAVs are exposed to malicious data and they may not distinguish between real and malicious data. To perform a false data injection, attackers typically take control of UAVs by injecting fake data into the UAVs' sensors using attackers' sensors.

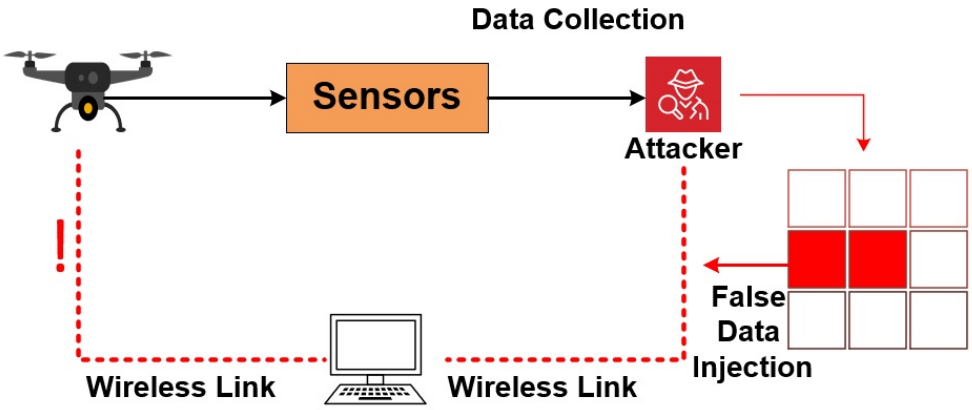

Figure 2.4 : False data injection attack on UAVs.

There are many countermeasures depending on the application of the UAV. In [64], a new algorithm is introduced to detect a data injection attack of a UAV bug. An adaptive neural network is used to detect errors injected into the sensors of the UAV. An integrated Kalman filter is used for in-line adjustment of the neural network weights. This inline setting makes intrusion detection faster and more accurate. Samy et al. introduce an NN-based system model for the fault detection process [65]. In this work, the system identification is achieved through the offline learning process, so the fault detection process does not need model data during the operation. A NN-based flaw detection design for failures of actuators on a satellite is presented in [66]. Shen et al. presented a neural network-based fault detection technique that takes into account the latency between a fault detection and a fault compliance [67].

Routing Attacks: An adversary attacks existing routing protocols to reduce the performance of an UAVs network or change its topology [41]. An attacker can degrade the performance of the UAV network by corrupting the routing algorithm or launching a DOS attack. An UAVs network's topology may be altered by adding non-existent nodes to routing tables, creating a fake route connection. In the literature, routing attacks may be explained within a controlling network traffic, interrupting routing functions, or injects malicious nodes. However, routing protocols may be exposed to different types of attacks. In literature, path discovery attacks, path maintenance attacks, and data transmission phase attacks based on their basic routing functions are analyzed as routing attacks [68]. [41] contains attacks on the routing protocol in a classified manner.

In these types of attacks, cryptographic methods and intrusion detection systems are used by an UAV on a communication line. For example, the Security Sensitive Temporary Routing (SAR) protocol uses security metrics in Route Request Packet (RREQ) [69]. It also uses a shared secret to generate a symmetric encryption key. A location-based routing solution to provide authentication and privacy is possible [70]. An anomaly-based detection mechanism that learns from the statistical analysis of different RREQ packet rates and calculates the threshold instantly is proposed in [71].

Eavesdropping Attacks: In an eavesdropping attack, the attacker passively or actively intercepts network communications to access secret information on an UAV network, such as node ID numbers, routing updates, or application-sensitive data. An attacker can use this private information to compromise network nodes, alter routing, or reduce application performance. Active eavesdropping aims to attack the main channel by reducing the channel capacity. Active spy transmits interference noise and simultaneously and independently captures spy signals [72]. Some eavesdroppers relay on the interference of a legitimate receiver, while others intercept the nosy.

Encryption is the standard defense against eavesdropping attacks. Due to limited processing power of some UAV systems in wireless communications, they cannot efficiently handle the standard encryption methods used in typical wired networks [73].

Side Channel Attacks: A side-channel attack refers to the disclosure of useful information about the internal execution of a system, either with transmitted data or via alternative routes. This type of attack obtains information indirectly by exploiting information leakage. Examples of side-channel attacks on UAVs include acquisition and analysis of meteorological information, power consumption, electromagnetic dissipation, acoustic signal analysis, and residual data. Defense against side-channel attacks includes the use of asynchronous processing units and mechanisms that help reduce electromagnetic emissions. The temporal information of the Unmanned Autonomy Systems communication channel is strongly associated with sensitive internal states, such as the number of UAVs.

Two possible solutions can be used to counter side-channel attacks [74]. Unmanned Autonomy Systems may randomize the temporal information of packets by generating redundant packets within random time intervals. While this solution may eliminate the loss of confidential information from the communication channel, it may, in turn, reveal the existence of the Unmanned Autonomy Systems communication channel, which may allow other attacks, such as signal interference. On the other hand, Unmanned Autonomy Systems can use traffic transformation to statistically make Unmanned Autonomy Systems traffic patterns indistinguishable from the traffic patterns of popular network applications, such as web clients or messengers [75]. Additionally, side-channel resistant implementations on devices in a UAV may help to have better security. For instance, Edwards Curve Digital Signature Algorithm (EDDSA) based on the Ed448 targeting the ARM Cortex-M4-based STM32F407VG microcontroller on IoT devices in a UAV system may help to prevent side channel attacks on UAV systems [76]. Since current systems have been expected to be broken by the advent of quantum computing, post-quantum solutions are expected to be implemented for critical systems, such as cryptographic accelerators for digital signatures that provide high-performance Ed25519 architecture [77]. Integrated countermeasures to high-performance post-quantum computing and their optimized version are expected to be emerging implementations on IoT devices for UAV systems [78].

Our research has also highlighted the growing use of machine learning techniques in detecting and mitigating side-channel attacks. Specifically, time-driven cache attacks targeting AES have been shown to leak timing patterns that can be effectively identified using machine learning algorithms trained on side-channel features [79]. Moreover, intelligent side-channel feature selection can further improve detection accuracy, demonstrating the potential of ML-enhanced cybersecurity strategies for embedded UAV systems [80].

Code Injection Attacks: After gaining access to the UAV networks and control units, an attacker can inject malicious code into the control units. Malicious payloads such as viruses, Trojan, and spyware may also infiltrate anti-malware software with this approach. UAV controllers may also inject code when one or more UAV components or subsystems are incompatible, in hopes of improving their vehicle's performance or misleading regulatory examination.

The defense against code injection attacks may be applying an intrusion detection system. In this case, the access control system should only grants permissions to authorized personnel. Additionally, UAVs may not include vehicle owners under certain circumstances that case need to be carefully considered [81].

Replay Attacks: A replay attack may destroy cyberphysical systems, including UAVs, even without knowing the external structure. The replay attack is basically shown in Figure 2.5. Replay attacks are the simplest attack that can be implemented for malicious purposes. An attacker records and covers transmitted data from a network perspective, compromising closed-loop capabilities of cyber-physical systems and degrading system performance.

A countermeasure against replay attacks is a recording horizon control law to address the attack on the communication network between the controller and the actuators [82]. Under bandwidth constraints, a secure fusion prediction scheme for cyber-physical systems against replay attacks has been proposed in [83]. Based on this detection measure, a stochastic play approach has been developed to reduce the loss of the control performance [84].

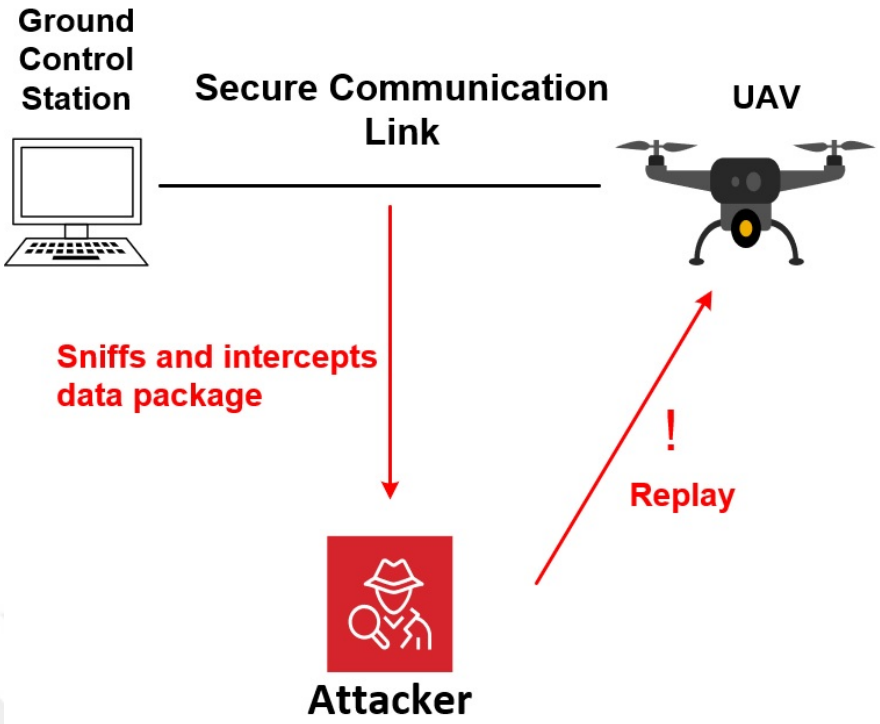

Figure 2.5 : A replay attack on an UAV.

Packet Sniffing Attacks: A packet sniffer may intercept and log traffic transmitted over a communication link. This type of attack is derived from a tool that is commonly used to diagnose network-related problems. An attacker can use a packet sniffer to spy on unencrypted data in packets to gather information.

Possible defenses against packet monitoring include the application of encryption techniques to protect the confidentiality of packets in transit, as well as distribution techniques to secure and authenticate sent and received communication signals [81].

Jamming Attacks: In UAV systems, jamming attacks are defined as electromagnetic energy emitted in a deliberate direction to a communication system to interrupt or prevent the transmission of the signal [85] as shown in Figure 2.6.

A jammer may be anything, such as from being a single specially equipped transmitter to all jamming stations. The goal of broadcast jammers is to reduce availability in a secure system. A comparative analysis of the various jamming techniques is discussed in [86].

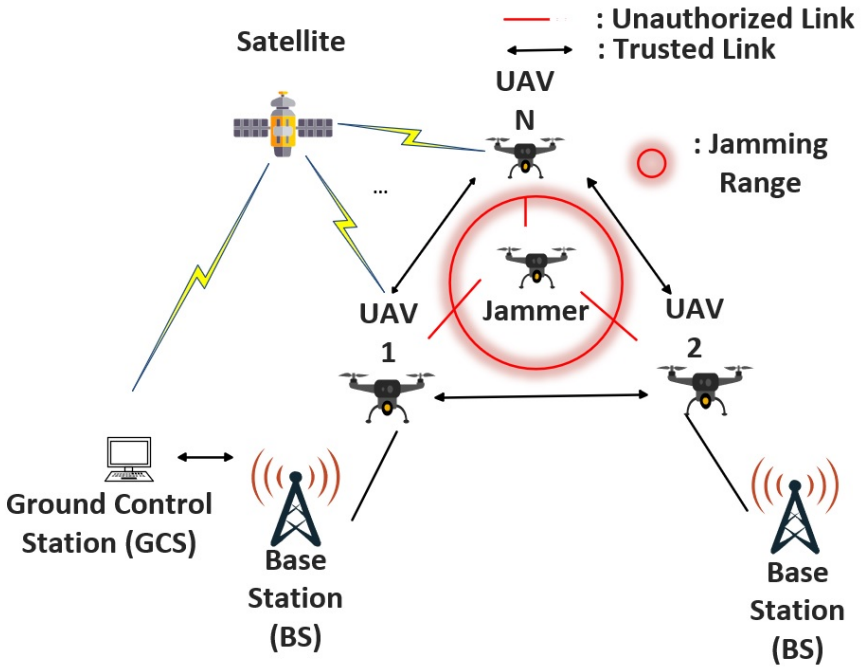

Figure 2.6 : A jamming attack on an UAV.

Classical countermeasures are block jamming attacks, moving away from the disrupted field hopping frequency, distributed attack detection, distributed secure predictions, and statistical approaches in wireless sensor networks [87,88]. Especially in VANETs, classical countermeasure methods are not always applicable due to high mobility and the wide network topology.

GPS Spoofing Attacks: The Global Positioning System (GPS) is typically used to provide UAV navigation information and time information. However, GPS is subject to many deliberate threats. For example, an attacker can mislead a GPS receiver with false GPS signals as shown in Figure 2.7. First, a reliable navigation system is required for new autonomous systems based on the standalone UAV. GPS sensors are the most widely used navigation sensors for high-performance flights in UAV navigation systems. In military applications, GPS signals are encrypted to prevent unauthorized use and spoofing. However, the current civil GPS signal is transparent and easily accessible worldwide, making GPS-guided civilian infrastructures vulnerable to various frauds or broadcast jammers [89]–[91]. There are also experimental studies to examine the GPS signal generation process and effects of spoofing signals on UAVs [92].

The recommended countermeasure is to prevent spoofing attacks on civilian UAVs is to encrypt GPS signals. However, this is quite costly since it requires significant upgrading of the infrastructure. There are various anti-spoofing techniques to detect a false signal. The control of WLAN interaction points, a cross-check detection control, an activation of an alarm system when the ratio between the signal strength and the noise exceed a certain threshold is one of them [93].

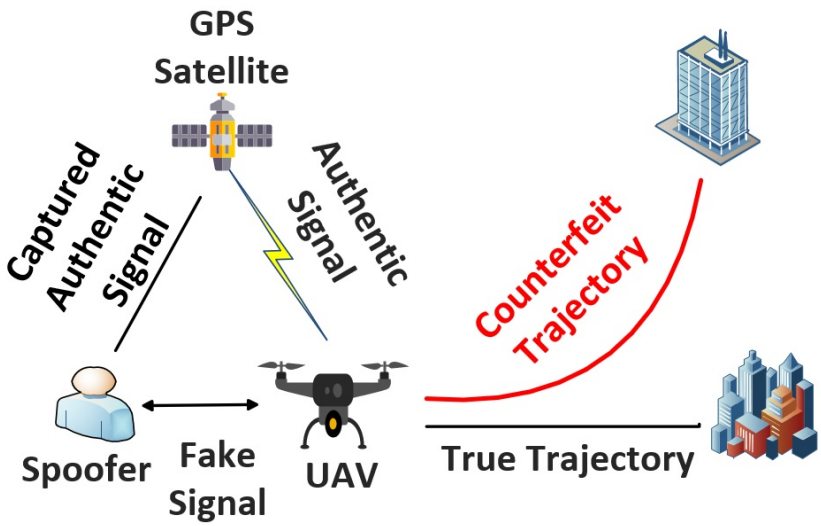

Figure 2.7 : A GPS spoofing attack on an UAV.

There are more advanced attacks on GPS that are relatively difficult to detect. For example, a receiver-based spoof system is more complex, which consists of a GPS receiver that is combined with a phishing transmitter. The generated signal is synchronized with real GPS signals [37]. Research is carried out to confirm whether the error between the UAV model estimator and the real signal is fake or not [94]. Threshold-based methods often increase the complexity of software and hardware. Moreover, the detection of a spoofed signal is not guaranteed when there is a very sophisticated spoof. Therefore, innovative and effective approaches are needed to secure GPS services in UAVs.

2.3 Traditional Defenses for UAVs

UAVs are vulnerable to a variety of attacks, including data interception, malicious data injection, jamming, and spoofing. These attacks may lead to a loss of control over the UAV, data theft, or damage to the UAV itself. Various defense techniques have been developed to prevent these attacks, including encryption methods, authentication mechanisms, and intrusion detection systems.

2.3.1 Encryption and authentication

Basic defense techniques against security attacks on UAVs are encryption and authentication. Encryption ensures that data is transmitted securely by encoding it in a way that only authorized users can decipher. Authentication ensures that only authorized users are allowed access to the UAV, its control systems, and its data. Cryptographic techniques are used to guarantee the confidentiality, integrity, and availability of the system. Specifically, symmetric cryptographic protocols are employed to safeguard sensitive information, such as text, audio, video, and images. In these protocols, both the sender and receiver have a shared key for both encryption and decryption purposes.

Asymmetric security protocols use a pair of distinct keys, namely a public key and a private key, for encryption and decryption of data by the sender and receiver, respectively. Unlike symmetric protocols, the confidentiality of the public key is not a critical issue, since encrypted data cannot be decrypted using the same key. Hence, a private key is always necessary for data decryption.

A lightweight authentication protocol is a security protocol designed to provide authentication of data or entities while minimizing the overhead on the system resources, such as memory and computational power. These protocols are particularly useful for resource-constrained devices such as UAVs, where traditional cryptographic protocols may be too heavy to be implemented. The goal of a lightweight authentication protocol is to provide sufficient security with minimal system resources. These techniques are commonly used to protect data transmitted between the UAV and the ground control station [95].

Encryption techniques in flying UAVs are also used to prevent attacks that cause hardware failures [63]. Thus, an enemy is prevented from obtaining confidential data. An authenticated encryption mechanism is used against eavesdropping attacks in similar cases [96]. Additionally, there are secure communication schemes against replay attacks [63,97].

Secure transmitter systems against hijacking, eavesdropping, and Distributed Denial-of-Service (DDoS) attacks are implemented using encryption methods. A System Development Life Cycle (SDLC) is used with a prototyping approach, which is implemented using three prototypes in [98]. The proposed scheme employs an Android application for a web platform and Advanced Encryption Standard (AES) algorithm for secure transmissions. Security tests have shown that the proposed scheme may resist attacks with a high success rate. The User Acceptance test has also confirmed that the application used for the secure transmission meets the user's requirements. In [99] proposed a method was proposed to verify the authenticity of data, which ensures that the data was sent by the original GCS and not by any unauthorized source. The proposed method uses asymmetric protocols to protect the integrity of data transmitted between different sensors or devices.

Hash algorithms are used in various parts of protocols to provide security for UAVs. For example, the verification of signatures after data is received by the recipient uses hash algorithms as in [100]. The UAV performs the verification process to ensure the confidentiality of data before doing an action. A 164-bit hash is generated using the SHA-1 algorithm on the sender side for verification of data. The generated hash is then encrypted using a public key and then sent to the recipient. The receiver uses a private key to decrypt the encrypted hash value. Then, it calculates the hash value from the data from the original message. The two checksums are then compared to verify the authenticity of the message.

Elliptic Curve Encryption is one of the significant cryptographic algorithm that is used in many protocols. In [101] proposed a lightweight recognition mode for authentication using Elliptic Curve Encryption (ECC) against UAV network attacks. The proposed approach is designed to protect against different types of security attack, such as message eavesdropping, fake identity, and replay attacks on a UAV network. The approach uses several security measures, such as an ECC digital certificate for UAV identification credentials, ECDSA for UAV identification and signature verification, and the Elliptic Curve Diffie-Hellman (ECDH) algorithm for session key negotiation during drone communication.

Lightweight security protocols are important for security of UAVs because the protocols consume a relatively small amount of energy. In [102] proposed a lightweight security protocol for IoD networks called Temporary Credential Based Anonymous Lightweight Authentication Scheme (TCALAS) that is limited to a single flight zone. This uses a threat model and an authentication. The protocol also includes a Ground Station Server (GSS), remote drones, a mobile device, and a control room. An upgraded version of the protocol is introduced in [103], which may provide security against various attacks and be implemented in multiple flight zones. The network model of the proposed scheme consists of a GSS, a control room, drones with related flight zones, and a drone user. The proposed scheme provides security features, such as anonymity and non-traceability of the user, mutual authentication, and robustness. This version of the protocol has also been shown to be faster to complete the entire authentication process.

Traditional security mechanisms use symmetric and asymmetric cryptography to implement security services. One of the significant threats to these services is the advent of quantum computing that requires post-quantum algorithms to be implemented and integrated to security services. UAV systems already use security services. Therefore, efficient implementations of post-quantum algorithms with different technologies, such as FPGA, are needed to protect UAV systems [104]. For example, high-speed NTT-based polynomial accelerators for post-quantum cryptography are required [105]. Another implementation may be on ARM Cortex-M4 [106]. A more detailed analysis of post-quantum computing is presented in [107], which covers quantum computing and post quantum computing for blockchain. All of these security services are expected to be implemented for UAV systems.

2.3.2 Intrusion detection systems

Intrusion Detection Systems (IDSs) are used to defend against security attacks on UAVs. These systems monitor the UAV's data traffic and alert the operator if any unauthorized or malicious activity is detected. IDSs may also monitor the UAV's physical environment, such as detecting any attempt to tamper with the UAV's hardware or software. In this paper, intrusion detection systems are examined in three categories.

1. **Rule based IDS:** Rule-based intrusion detection is performed. In this category, rules following the expected behavior of the UAV system are applied on UAVs to perform special tasks [108]. Rule based IDS sets specific rules and policies that define network behavior. Any deviation from these rules is detected as a potential intrusion.

2. **Signature based IDS:** This category involves the comparing network traffic with a known signature database or known attack patterns. If a match is found, an alarm is triggered and the intrusion is detected. A Bayesian-based technique for detecting insiders and removing malicious nodes from the network is presented in [109]. The technique includes activating an intrusion detection mode for various nodes and calculating the false behavior rate (MR) for nearby UAVs. If the threshold is lower than MR, the intrusion detection system starts monitoring nearby nodes and detects intrusions. The intrusion removal system determines the MR of a node, and if its threshold is less than the MR, the node is declared rogue and it is removed from the network.

3. **Anomaly-based/Machine Learning-based IDS:** This category involves detecting known and unknown attacks based on learning or filtering mechanisms. The working principle of Anomaly-based/Machine Learning-based IDS is identifying unusual or anomalous behavior in the network. It uses statistical analysis or machine learning algorithms to distinguish normal behavior and detect any deviation from the normal behavior. Machine learning (ML) algorithms have two phases, namely training and testing. The model is trained to predict future events using training data in the training phase. The testing phase uses various strategies to measure the model's accuracy. A summary of ML-Based Identities for UAV Security is given in Figure 2.8.

Each IDS method has advantages and disadvantages. For example, signature-based IDS is effective for detecting known attacks but it is weak against new or unknown attacks that change their patterns frequently. If one bit changes, the signature changes, too. Anomaly-based IDS may detect unknown attacks, but they may also suffer from false positives and false negatives. A more efficient IDS is to use a hybrid detection approach that combines two or more approaches for the accurate detection of unknown attacks [110]. A hybrid IDS can strike a balance between the two and is often the preferred approach for UAV security.

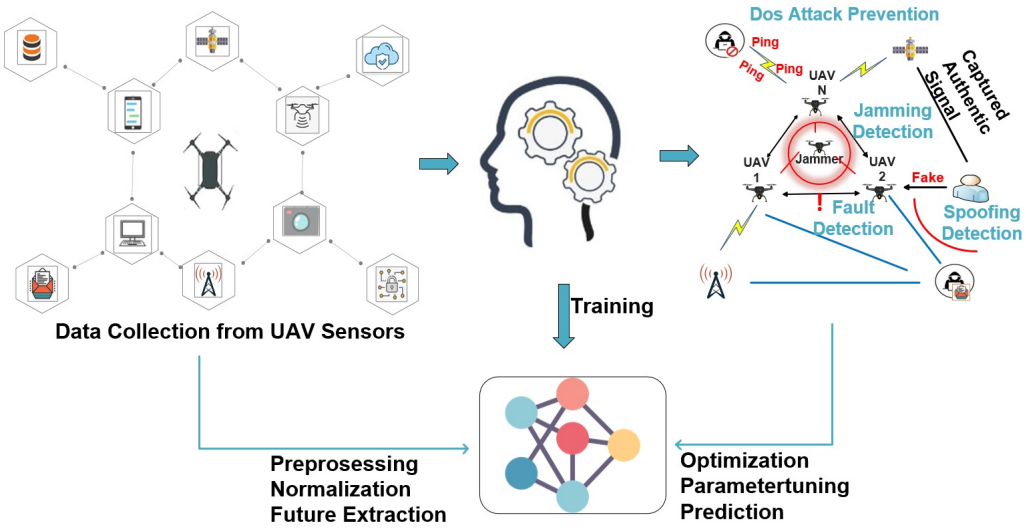

Figure 2.8 : A summary of ML-based IDS for UAV security.

Recently, ML-based IDS has become popular. For example, a machine learning-based IDS was proposed against GPS Spoofing attacks in [111]. The recommended method includes the one-class support vector machine and ML autoencoder algorithms. A method for detecting eavesdropping attacks on UAV communication is presented in [112]. The method uses K-means clustering and support vector machine (SVM) algorithms that may learn from existing data and make decisions for future samples. The technique consists of two phases. In the first phase, both parties send signals to the UAV. In the second phase, the UAV transmits the same signals to a third party to detect any deviation. A dataset is created and classified using machine learning algorithms to identify potential attacks. Also, feature reduction techniques combined with machine learning have been effectively explored for malicious traffic detection in recent studies [113].

ML-based IDS is used against signal spoofing and jamming attacks. In [114], Self-Teaching (STL) with a multi-class SVM is used to maintain a high true positive rate for IDS. It uses Deep-Q Network, a deep reinforcement learning algorithm for the drone dynamic route learning as a self-healing method in the IDS recovery phase. The proposed solution collects data from UAV components, such as flight logs and reading data from sensors. However, a real-world application of UAVs is difficult due to the limited power and a lack of enough computational resources.

A convolutional neural network (CNN) method is used to detect jamming signals in [115]. The UAV algorithm takes into account the weights and values of GCS, selects a relay power element based on Bit Error Rate (BER), and uses the boost and randomly selected values to send a message. While the algorithm may protect communications and jamming attacks, randomly selected relay power may increase the error rate, making it expensive.

Several machine learning-based security frameworks have been introduced in the literature to address a variety of security issues, such as malicious drone detection and DoS attack prevention for UAVs [116]. Some recent research has shown that federated learning techniques may be more effective than traditional machine learning algorithms. For example, there are reports on the development of radio frequency-based UAV authentication models using IoT networks [117].

A Lightly Distributed Detection Scheme aims to detect and mitigate flooding attacks in the Internet of Drones (IoD) environment that is explained in [118]. The scheme uses the concept of automatic count reports, where each drone counts the number of packages it sends in a given time interval and shares that report with other drones during contacts. Receiver drones store reports and send them to nearby ground stations to check consistency and detect flood attacks.

Specific attacks may require specific countermeasures. Protection against specific attacks is achieved through path planning, with illustrative examples in [119]. An airway-aware protection mechanism (IoD-JAPM) against jamming attacks (JA) on the Internet of Drones (IoD) is presented, including an analysis of airway availability and potential modifications on drone path planning. Specifically, IoD-JAPM effectively uses jamming attacks on all drones in the presence of a jammer. The approach contains three key stages, which are airway analysis, risk region (HR) discovery, and route planning generation.

A hybrid IDS model integrating spectral traffic analysis and a robust controller/observer for anomaly estimation within UAV networks is presented in [120]. The IDS is created against DDOS attacks. The hybrid method takes into account, in its initial phase, a statistical representation of the network's traffic. By analyzing the newly created signatures, anomalies may be determined with an appropriate model to accurately estimate the anomalous traffic.

The existing literature on attacks and countermeasures highlights that each attack vector focuses on a different aspect of UAVs, thereby undermining specific security requirements. Therefore, it is important to note that a single defense technique is not sufficient to protect UAVs from security attacks. A combination of these techniques must be applied to provide comprehensive security to UAVs. Moreover, the power consumption and battery usage must be evaluated according to the target of the attack to determine potential countermeasures. In addition, the defense techniques used should be regularly updated and tested to ensure that they are effective against the latest security attacks.

2.4 Generative Artificial Intelligence in UAV Security

UAVs are increasingly targeted by adversarial attacks, including spoofing, jamming, and data injection. Various GAN-based methods have been proposed to detect and mitigate these attacks and to enhance IDS performance. GUIDE [121] and G-IDS [122] systems utilize GANs to generate synthetic data, addressing the challenges of imbalanced datasets and improving IDS robustness. GAN-based models like SeqGAN and LeakGAN have demonstrated significant improvements in attack detection, which increase accuracy up to 37% for known attacks and 30% for unknown attacks.

While GANs have been used to enhance IDS, they are also leveraged to generate adversarial attacks. IDSGAN [123] is a framework designed to create malicious traffic that deceives IDS systems. By dynamically adapting to IDS models, IDSGAN may effectively bypass detection and challenge system robustness. Similarly, GIDS employs GANs to create synthetic data for in-vehicle network IDS, achieving high detection accuracy for known and unknown attacks [124]. Generative models are employed to address the scarcity of security-related data for UAVs, such as the use of Conditional Tabular GANs (CTGAN), Variational Autoencoders (VAE), and Gaussian Copulas to generate realistic jamming attack scenarios. These models enable the training of IDS on diverse and enriched datasets by improving detection capabilities [125]. In one of our previous studies, we proposed a novel framework that leverages Conditional Tabular GAN (CTGAN) to generate adversarial Denial-of-Service (DoS) attack data for UAV systems, addressing the challenge of data scarcity and enhancing the training of machine learning-based intrusion detection systems through realistic and diverse synthetic scenarios [126]. Moreover, GANs are combined with active learning to dynamically adapt IDS models, improving their effectiveness against evolving threats [124], [125].

Despite these advancements, the grand challenge, the lack of enough attack data, remains. Generative models often struggle with time-series data augmentation, as highlighted in MUVIDS and GUIDE implementations [127], [121]. While GAN-based IDS may detect novel attacks, they require extensive training data and computational resources, which limit their scalability for real-time applications [123,128].

Recent advances in generative modeling have significantly influenced the field of cybersecurity, particularly in the development of more robust Intrusion Detection Systems (IDS). Diffusion-based models have shown notable performance in generating realistic, high-dimensional network traffic data. For instance, Wang et al. [129] proposed a Denoising Diffusion Probabilistic Model (DDPM)-based intrusion detection system specifically for UAV networks, which demonstrated superior accuracy and robustness compared to traditional methods such as VAE and PCA. In addition, Yang et al. [130] introduced the Diff-IDS framework, a lightweight intrusion detection system that leverages a diffusion model with feature enhancement to handle imbalanced traffic data, outperforming state-of-the-art approaches on benchmarks such as CICIDS2017 and NSL-KDD. Although the study incorporates traditional augmentation techniques (e.g., image flipping), the diffusion model itself is used exclusively in a discriminative role, clearly separating generative and discriminative functions within the framework. A comparative study by Ammara et al. [131] found that CTGAN and CopulaGAN models yielded the highest fidelity and utility scores for generating network traffic data, effectively supporting IDS training. Chalé and Bastian [132] further confirmed the utility of synthetic cyber data generated via GANs and VAEs for training ML-based NIDS, demonstrating that hybrid datasets (real + synthetic) maintained detection accuracy while addressing data scarcity issues.

These contributions validate the strategic use of both Conditional GANs and DDPMs in our proposed framework and substantiate their potential to replicate complex adversarial behaviors while preserving data realism and security relevance.

While previous studies have effectively demonstrated the utility of GAN-based methods such as GUIDE [121], G-IDS [122], SeqGAN, and LeakGAN for enhancing IDS through synthetic data generation and adversarial training, our research extends beyond these approaches in several key ways. Unlike prior work that primarily focuses on a single generative model or application scenario, we systematically evaluate and compare multiple generative models—VAE, Gaussian Copula, CTGAN, and DDPM—to generate synthetic jamming and spoofing attacks specific to UAV systems.

In contrast to frameworks like IDSGAN [123] and Diff-IDS [130] that either simulate adversarial traffic or apply a single model to imbalanced datasets, our work introduces a multidimensional evaluation framework that integrates statistical similarity measures (e.g., KL Divergence, Wasserstein Distance, KS Test) with IDS classification metrics to assess the fidelity and impact of synthetic data comprehensively. Table 2.1 provides a structured comparison between our proposed framework and three notable GenAI-based IDS approaches: GUIDE, IDSGAN, and Diff-IDS. Each of these works addresses synthetic data generation for intrusion detection, but with different emphases and limitations.

Table 2.1 : Comparison of our framework with existing GenAI-based IDS approaches.

Feature	**GUIDE** [121]	**IDSGAN** [123]	**Diff-IDS** [130]	**Our Framework**
Generative Models Used	GAN (single model)	GAN (for adversarial sample generation)	Diffusion (DDPM – discriminative role only)	VAE, CTGAN, Gaussian Copula, DDPM
Application Domain	Time-series emulation	Adversarial traffic generation to bypass IDS	CICIDS2017 network traffic anomaly detection	UAV-specific IDS with jamming/spoofing focus
Attack Types	Flooding and unknown attacks (UAV-specific)	Adversarial samples to deceive IDS	Imbalanced network traffic (not UAV-specific)	Jamming and spoofing (UAV-specific)
Evaluation Strategy	Imbalance mitigation, TPR/FPR metrics	Bypass rate, adversarial success	Benchmark evaluation (accuracy, F1)	Multimodel cross-testing, statistical similarity + IDS metrics
Real+Synthetic Data Fusion	No explicit fusion	No	Not applicable (discriminative use only)	Yes – analyzed in various training-test configurations
Novelty in Modeling	Time-aware GAN for sequence data	IDS-evading sample generation via GAN	Lightweight diffusion model with feature enhancement	First UAV-focused DDPM generator, multi-modal threat generation, statistical fidelity assessment
Generalization Evaluation	Limited to imbalance settings	Focus on adversarial evasion	Focus on lightweight inference	Multi-train cross-testing with real/synthetic splits

Our inclusion of diffusion models (DDPM) for synthetic attack generation represents a novel contribution in UAV IDS research, building on recent advances but tailoring them specifically to the temporal and spectral complexities of UAV telemetry data. It also differs from previous studies in that it is explicitly applied to both jamming and spoofing attacks in the context of UAV systems, addressing a broader range of threat modalities that are often treated separately or overlooked in earlier research.

Importantly, our multi-train cross-testing strategy evaluates IDS generalization across different combinations of real and synthetic training sets, revealing the strengths and weaknesses of synthetic data under various deployment conditions. This novel integration of generative diversity, security-oriented data augmentation, and systematic cross-evaluation distinguishes our work from earlier GAN-based IDS research.

2.5 Reinforcement Learning-Based Intrusion Detection

The field of reinforcement learning (RL) has led to significant advancements in developing intelligent intrusion detection systems (IDS), as it facilitates the capacity for adaptive and context-aware decision-making in response to dynamic cyber-physical threats. Unlike traditional rule-based or supervised learning approaches, RL-based IDS can learn optimal defense policies through sequential interaction with the environment, making them particularly suitable for mission-critical and resource-constrained environments, especially UAV platforms.

In the literature, attacks have targets on all components of the UAV network infrastructure. Therefore, there is a need for environment-independent methods to detect various attacks quickly. Deep Reinforcement Learning methods have huge potential to detect attacks on UAV systems in a faster manner. DRL may also provide the most convenient countermeasures for UAVs by ensuring a self-adaptation that is independent from the environment and resists to many attacks. A summary of deep reinforcement learning applications in communication networks is presented in [40]. In this paper, we discuss the recommended reinforcement learning applications for detecting jamming and spoofing attacks targeting communication in UAVs.

In modern broadcast jammer prevention techniques, an attack-free self-learning protection system protects the communication with deep reinforcement learning. In the literature, broadcast jamming attacks have been discussed in different communication environments, such as Cognitive radio network (CRN), WSN, 5G, GPS, VANET, and MANET in UAV. Deep reinforcement learning techniques vary according to the area of usage in UAV systems. For example, DRL effectively decides power allocation, relay or not, channel selection, user selection, and resource allocation. Moreover, different reinforcement learning practices may adapt very quickly in natural environments.

Broadcast-busting attacks may have disruptive consequences for UAVs. Although there are solutions to prevent broadcast-busting attacks, they are inefficient in most cases [133]. DL methods may help to create efficient solutions, such as a dynamic game computational radio channel model against jamming attacks in Deep-Q-Network (DQN) communication. DQN can be used to control the power in a jamming attack targeting the communication [134]. In this case, devices decide to transmit energy by performing the selected action. The Signal to Interference Noise Ratio (SINR) value is measured in a time frame and it is calculated at the end of that time frame. The news channel is tuned in Universal Software Radio Peripherals (USRP). A reactive jammer calculates the utility to select the blocking power that is based on the final transmitted power of the transmitter. At each time slot, the disruptor observes the final SINR and then selects the parasitic power that can be exploited with a greedy strategy. The transmitter selects and adjusts the transmit power in each time slot and transmits data packets to the radio station. It selects the transmission power according to the DQN algorithm to improve the SINR of the transmitter and reduce the energy loss. DQN-based power control strategy improves communication efficiency compared to the Q-learning-based strategy.

Game theory and Q-learning are used to avoid smart jammers, who are targeting UAV communications in VANETs. For example, a game model against jamming is proposed in [135]. In the game, the UAV decides whether to transmit the message while choosing the smart jamming strength. The transmission decision of the UAV depends on the channel quality and Bit Error Rate (BER). An initial anti-disruptive transfer strategy, which is called Policy Hill Climbing Algorithm (PHC) algorithm, is used to obtain the optimum transfer strategy without knowing the VANET model and the compression model [136]. This model reduces the BER in VANET and increases the utility of the UAV compared to the Q-learning-based transmission strategy [137].

UAVs use 5G networks for their communications. Jamming prevention is critical in UAV communications with 5G networks. In this case, it is recommended to move the transmitter away from the faulty area to prevent broadcast jamming, which may interrupt the communication or simply creates communication problems. If 5G networks are used, it is recommended for the UAV to use a fast DQN, learning techniques, deep learning, deep reinforcement learning, and transfer learning together to decide the transmission power control mechanism without knowing the broadcast jammer and the 5G network model [138].

A game-based defense mechanism against GPS spoofing attacks for civilian UAVs is possible as in [139]. For example, a zero sum game using the Stackelberg methodology to represent strategic interactions between security agents deployed in UAVs and potential attackers targeting the Unmanned Aerial Vehicle Edge Computing (UEC) network is presented in [140]. This non-cooperative game requires security agents to protect UAVs and all communication links, which include U2U communication between UAVs and U2I communication between UAVs and infrastructure. It is possible to detect spoofing attacks faster and in more adaptive manner by using Q-learning [141].

Black Widow Optimization (BWO) algorithm, which is specially designed for UAV networks, is used with a DRL technique for optimization purpose in [142]. This approach aims to improve the intrusion detection performance of UAV network attacks. BWO algorithm is used for parameter optimization of the DRL technique, which provides better intrusion detection performance in UAV networks. The effectiveness of the technique is validated using NSL-KDD dataset. The DRL technique includes a reinforcement learning-based Deep Belief Network (DBN) developed for intrusion detection that allows the identification of intrusions into UAVs from networks. Furthermore, the BWO algorithm is used to determine the optimal values of the hyper-parameters within the proposed model. Analyses results show highly precise values which confirm the effectiveness of the approach.

In [143], a novel DRL-based approach for trajectory planning and interference rejection of an UAV in Internet of Things applications is presented. The radio environment is enhanced to suppress interfering signals and extend the desired signals using a re-configurable intelligent surface (RIS). Using Deep Deterministic Policy Gradient (DDPG) and Double Delay DDPG (TD3) models, the UAV autonomously learns its trajectory and RIS configuration based solely on changes in the received data rate. The simulation results show that the proposed DRL algorithms provide robust resistance to UAV interference, especially with the TD3 algorithm that provides faster and smoother convergence than the DDPG algorithm for larger RISs. Notably, this DRL-based approach offers important practical benefits by eliminating the need for explicit knowledge about RISs and disruptive channels.

Collision-free paths for multiple UAVs connected to cellular networks are explored by providing connectivity with Ground Base Stations (GBS) in the presence of a dynamic jammer in [144]. The problem is formulated as a sequential decision challenge in a discrete space, taking into account connectivity, collision avoidance, and kinematic constraints. Authors present an offline time difference (TD) learning algorithm that is combined with online SINR mapping as a solution approach. Specifically, an offline network is created and trained using the TD method to capture interactions between UAVs and the environment. Additionally, an online SINR mapping based DNN is designed and trained through supervised learning to capture the influence and changes caused by the block. The numerical results show that even without any mixer knowledge, the proposed algorithm achieves performance levels comparable to the ideal scenario with a perfect SINR map.

Secure communication between a UAV and a ground user in an urban environment is a significant requirement. was investigated. In [145], the stage includes several spies and a UAV jammer that generates artificial noise to disrupt spies' activities. The goal is to maximize stealth rates at the physical layer by co-opting the trajectory and transmission power of the UAVs. To handle the time-varying channel conditions, the problem is formulated as a Markov decision process. An advanced algorithm based on the double DQN is proposed to solve MDP. The simulation results show that the rapid convergence of the algorithm in various environments allows the UAV transmitter and UAV jammers to accurately determine the optimal locations to maximize information privacy rates. Furthermore, the performance comparison shows that the Double DQN (DDQN) algorithm outperforms the Q-learning and Deep Q-learning Network approaches. In another research, a secure evacuation system consisting of an end server, a ground control station, and a malicious eavesdropper UAV is presented [146]. The goal here is to maximize privacy by offering a UAV that may dynamically switch between scrambling and transfer modes. An algorithm based on a deep deterministic policy gradient (DDPG) is proposed to achieve the goal.

A physical layer security (PLS) in millimeter-wave rotary wing unmanned aerial vehicle communication using re-configurable intelligent surfaces is explained in [147]. Having multiple listeners and imperfect channel state information (CSI) may create a security problem in an UAV. To solve this problem, DRL is used to make real-time decisions in any time frame considering the dynamic UAV environment. To address continuous optimization variables, authors present a twin-twin-lag deep deterministic policy gradient (TTD3) approach, which maximizes the expected cumulative reward and improves the safe energy efficiency (SEE). The results demonstrate that the proposed method outperforms the traditional deep deterministic policy gradient twin DRL (TDDRL)-based approach.

A new framework with deep reinforcement learning for radio surveillance, in which a fixed-wing UAV is used to capture the radio fingerprint of a suspicious transmitter with the help of a benign Reconfigurable Intelligent Surface (RIS), is presented in [148]. The framework includes a newly designed TD3 model, which enables the UAV to learn both its trajectory and the RIS configuration that is based solely on the observed transmission rate of the suspicious transmitter. This research includes a fixed-wing UAV. Action and reward components are customized to suit the UAV's mobility constraint. Simulations demonstrate that the method offers the UAV reliable radio surveillance ability while it also keeps the desired distance from the UAV to the transmitter.

A covert communication system is used with intelligent reflective surfaces (IRS) in UAVs in [149]. The goal of the research is to improve the performance of confidential communications. Authors propose a new algorithm called Double Deep Q Network-based Orbit and Phase Optimization (TAP-DDQN). They can improve stealth communication performance by optimizing the 3D trajectory of the UAV and the phase configuration of the IRS. Through simulations, they demonstrate that TAP-DDQN outperforms reference solutions and leads to significant improvements in the stealth performance of the IRS-powered UAV stealth communication system.

In [150], authors present a framework that uses RL algorithms to detect intrusions in Mobile Unmanned Wireless Networks (MUWNs). The paper identifies three main categories of attacks on MUWNs, which are jamming, impersonation, and intrusion. Impersonation attacks involve attackers posing as UAVs within the MUWN to offer deceptive services or maliciously acquire data. Intrusions involve the direct upload of malicious software to the target MUWN. The authors present a case study which is based on DDPG algorithm to demonstrate how RL can be applied for intrusion detection.

Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm, known for its effectiveness in multi-agent situations, is proposed to overcome the obstacles presented by collaborative mixing and trajectory design in a scenario where multiple UAVs operate together [151]. UAVs fall into two categories. One category serves as aerial Base Stations (BSs) that transmit data to users, while the other one serves as a jammer that emits fake sounds to disturb listeners on the ground. The proposed system assumes that UAVs have information about the locations of both ground users and eavesdroppers. Training for both groups of UAVs is performed using MADDPG to dynamically change their position and maximize the combined safety ratio of all ground users. The safety ratio for a ground user is determined by subtracting the maximum acceptable signal-to-noise ratio for the entire ground eavesdropper from the received signal-to-noise ratio. Simulation results show that an efficient joint orbit design of UAVs is achieved by the MADRL method. To improve learning efficiency and convergence, the Continuous Action Attention MADDPG (CAA-MADDPG) method is also presented. The simulation results show that CAA-MADDPG outperforms MADDPG and provides better reward performance.

A comprehensive review of reinforcement learning algorithms employed in intrusion detection systems (IDSs) across diverse cybersecurity domains, including traditional networks, IoT environments, wireless systems, and cloud infrastructures, is presented in [152]. Numerous studies in the literature have employed traditional reinforcement learning (RL) algorithms, including Deep Q-Networks (DQN), Policy Gradient (PG), and Actor-Critic (AC), for the development of intrusion detection systems (IDS).

In the [153] study, the authors introduced a deep reinforcement learning (DRL)-based network intrusion detection system (NIDS) and conducted a comparative analysis of four reinforcement learning (RL) algorithms. DQN, Double DQN (DDQN), PG, and AC. The experimental results obtained from this study indicated that DDQN demonstrated the optimal performance. The RL components of this work were modeled using AWID [154] and NSL-KDD [155] datasets samples as states, labeling operations as actions, and a binary reward signal based on classification accuracy.

In the [156] work, the authors proposed a real-time IDS using DQN. In this environment, the defining factor is network traffic; the agent functions as the detection engine, and rewards are contingent on the accuracy of classification decisions.

For cloud-based environments, the study by [157] proposes a DRL-driven IDS comprising three distinct components: a host network (environment), an agent network, and an administration network. Feature vectors were utilized as the state-space, and logical XOR operations between decision vectors and classifications comprised the action space. The DQN agent was rewarded for accurate classifications.

In [158], a multi-UAV is used to receive signals from a ground station. Some UAV jammers near the target try to interfere with UAVs to reduce their received SINR. In this research, a UAV is considered as a unique system, such as an agent that is trained by RL algorithm, while jammers adjust their transmission powers and positions depending on predefined strategies. UAVs only use RSS and SINR to make decisions. A Q-learning-based model is trained by using RSS, SINR, and predicted trajectories of jammers, which are called knowledge-based RL. Although the location of the jammer is unknown, the agent can estimate the jammer trajectory based on the change of the RSS value with the distance and the flight inertia of the jammer.

A ground node sends confidential information to a legitimate UAV while a smart UAV eavesdropper is present that can adjust its position for optimal eavesdropping [159]. The legitimate UAV and the eavesdropper are both treated as conflicting agents. The goal of the legitimate UAV is to maximize the total security, while the eavesdropper aims to minimize security. The problem is redefined as a two-player zero-sum stochastic game (TZSG). Findings demonstrate that the legitimate UAV strategically chooses a communication link away from the Eavesdropper while the Eavesdropper seeks to intercept confidential information.

DRL helps UAVs to safely perform their duties by adapting themselves to the environment without the need for costly infrastructure changes. The DQN-based channel selection strategy may improve communication efficiency compared to the Q-learning-based strategy. In the literature, there are solutions against different attacks on UAVs using deep reinforcement methods, such as DQN, game theory, PHC, and actor-critic methods.

Reinforcement learning (RL) has emerged as a powerful paradigm for designing intelligent intrusion detection systems (IDS) due to its capacity for adaptive decision-making in dynamic threat environments. Unlike traditional rule-based or supervised learning methods, RL-based IDS frameworks are capable of learning optimal defense strategies through interaction with the environment, which is especially advantageous for mission-critical and resource-constrained domains such as IoT, cloud, and wireless networks. Several surveys have summarized the application of RL techniques—including Deep Q-Networks (DQN), Policy Gradient (PG), and Actor-Critic (AC)—in IDS development across various infrastructures [152,157].

A wide body of research has focused on evaluating and comparing RL algorithms in traditional network intrusion detection systems. For instance, [153] conducted a comprehensive evaluation of DQN, DDQN, PG, and AC on datasets such as AWID [154] and NSL-KDD [155], demonstrating that DDQN achieved superior performance. Similarly, [156] leveraged DQN for real-time classification of network traffic, where rewards were determined based on prediction accuracy, showcasing RL's suitability for high-speed detection scenarios. In another line of work, [157] proposed a DRL-driven IDS for cloud environments using XOR logic for the action space and feature vectors as states, while [160] enhanced encrypted traffic detection by combining DQN with adversarial sample generation and ResNet-based classification, achieving 99.94

Other works have introduced hybrid approaches integrating RL with deep learning architectures or optimization methods. For instance, the DAEQ-N model in [161] merges deep autoencoders with Q-learning in environments such as OpenAI Gym, enabling the model to adaptively classify traffic while minimizing false alarms. Similarly, [162] combines convolutional neural networks (CNNs) with Q-learning to detect low-rate DDoS attacks in cyber-physical social systems, reporting a 95.22% detection rate. These models highlight the flexibility of RL in tackling diverse security challenges across domains. Moreover, adversarial training has also been investigated, such as in the AE-RL framework of [163], which employs both DQN and DDPG agents to simulate attacker-defender interactions in dynamic threat environments.

Reinforcement learning has also gained significant traction in the UAV security domain, where real-time adaptability and resource constraints are key challenges. For example, [150] proposed a DRL-based intrusion detection framework tailored for aerial computing networks, demonstrating its superiority over conventional methods in onboard UAV deployment. Extending this idea, [164] developed a context-adaptive IDS leveraging multi-agent DRL and denoising autoencoders, evaluated across datasets such as NSL-KDD, UNSW-NB15, and AWID. Similarly, attention-based mechanisms have been incorporated into DRL frameworks as shown in [165], where agent-specific attention modules improve scalability and robustness in distributed IDS deployments.

More recent contributions have introduced cooperative and metaheuristic-enhanced frameworks. In [166], a two-layer Q-learning-based IDS for Internet of Drones (IoD) dynamically adjusted voting and auditing policies to balance detection accuracy and energy consumption. Meanwhile, [142] employed Black Widow Optimization to fine-tune a DRL-based Deep Belief Network for UAV intrusion detection, achieving a 98.9% accuracy rate. To enhance feature selection and attack generalization, Ren et al. proposed ID-RDRL [167] and MAFSIDS [168], incorporating recursive elimination and GCN-based feature extraction into multi-agent DRL pipelines.

Hybrid IDS frameworks combining CNNs, LSTMs, and reinforcement learning have also been proposed for UAV contexts. For instance, [169] fused CNN-LSTM feature extraction with deep Q-networks, achieving 99.99% precision and demonstrating resilience to evolving threats. Similarly, [170] introduced an adversarial reinforcement learning-based IDS for 6LoWPAN networks, which, although designed for IoT, shares relevance with UAV security due to its emphasis on lightweight operation and adaptability in lossy networks. Hierarchical RL has emerged as a promising extension for layered threat detection. Beyond single-agent solutions, hierarchical reinforcement learning (HRL) offers structured approaches for layered decision-making. A recent RF-based UAV detection study achieved 99.7% accuracy by designing a REINFORCE-driven HRL model that sequentially performs presence detection, type classification, and flight mode estimation [171]. Extending this concept to collaborative settings, Arranz et al. introduced a centralized swarm architecture that employs PPO-trained subagents, coordinated by a master controller, for distributed surveillance and tracking. This illustrates the potential of multi-agent RL frameworks for UAV networks [172].

Addressing UAV-specific constraints, [173] proposed a DQN-based IDS that includes a custom reward function to manage class imbalance and a periodic offline learning scheme to save energy, demonstrating superior performance in UAV environments. In the same direction, the AirEye platform developed by [174] combines a Raspberry Pi-based control unit, QR code recognition, and RL-guided navigation for energy-aware UAV exploration, validated in both physical and simulated environments.

Table 2.2 : Summary of limitations in RL-based IDS approaches for UAV security.

Study	RL Technique / Scenario	Reported Limitations
Han et al. (2017) [133]	DQN for jamming power control	High complexity due to SINR estimation; Greedy jamming limits generalization
Chen et al. (2018) [134]	DQN for adaptive SINR-based power control	Real-time energy control is limited by environment assumptions
Xiao et al. (2018) [135]	Game-based Q-learning in VANET	BER adaptation depends on static assumptions; limited environment diversity
Zhang et al. (2017) [139]	Game-based GPS spoofing defense	Assumes perfect knowledge of attacker behavior; limited scalability
Sedjelmaci et al. (2019) [140]	Stackelberg security game for UAV edge computing	Non-cooperative model oversimplifies attacker coordination
Praveena et al. (2022) [142]	DRL with BWO optimization	Optimization depends on hyperparameter tuning; lacks transferability to real UAVs
Hu et al. (2023) [143]	DDPG / TD3 with RIS for interference mitigation	Requires RIS deployment; learning overhead with larger surfaces
Wang et al. (2021) [144]	TD learning with online SINR mapping	Offline training dependency; supervised SINR mapping needed
Qian et al. (2023) [145]	DDQN for stealth trajectory and jamming	Limited interpretability of optimal stealth path decisions
Yoo et al. (2023) [146]	DDPG-based UAV switching mechanism	Sensitive to mode-switch threshold tuning; vulnerable to delayed transitions
Tham et al. (2023) [147]	TTD3 for secure mmWave UAV-RIS communication	Imperfect CSI severely affects optimization convergence
Yuan et al. (2023) [148]	TD3 with UAV-RIS for radio surveillance	Requires RIS calibration and UAV stability during monitoring
Bi et al. (2023) [149]	TAP-DDQN for covert comm. via IRS	IRS optimization complexity; real-world deployment unverified
Tao et al. (2021) [150]	DDPG for MUWN intrusion detection	Fixed attack categories; generalization to unseen attacks not tested
Zhang et al. (2020) [151]	MADDPG for cooperative secure comm.	Coordination overhead in multi-agent UAVs; training convergence issues
Louati et al. (2024) [152]	Survey of RL in IDS	Highlights general issues: reward design, environment realism, scalability
Lopez-Martin et al. (2020) [153]	Comparative analysis of DQN, DDQN, PG, AC	DDQN outperforms others, but real-time deployment not addressed
Hsu et al. (2020) [156]	DQN for real-time network IDS	Limited feature space; high false positive sensitivity
Sethi et al. (2020) [157]	DRL in cloud IDS	Logic-based actions limit adaptability; XOR logic impractical in UAVs
Li et al. (2021) [158]	Q-learning for signal-based jammer detection	Requires accurate trajectory prediction; limited to known patterns
Wen et al. (2022) [159]	Stochastic game for eavesdropping avoidance	2-agent assumption restricts multi-threat scenarios
Ren et al. (2022, 2023) [167,168]	Recursive feature elimination + DRL	High model complexity; increased training overhead
Khayat et al. (2025) [169]	CNN-LSTM + DQN hybrid IDS	Requires large labeled datasets; complex training pipeline
Soliman et al. (2022) [174]	RL-guided UAV navigation with QR codes	Hardware constrained; scalability untested
Salameh et al. (2023) [175]	Federated Q-learning for UAV collaboration	High communication load; assumes consistent battery levels
Strickland et al. (2024) [176]	GAN + DRL for minority class attacks	Sensitive to GAN quality; overfitting on minority class possible
Bouhamed et al. (2021) [173]	Energy-aware DQN for UAV IDS	Offline learning dependency; assumes stationary environment

Finally, several studies have explored cooperative UAV defense. In [175], a federated RL-based system allowed two UAVs to collaboratively track deceptive targets under battery constraints using Q-learning and SARSA. Similarly, [176] combined GANs and DRL to improve IDS performance on minority-class attacks using synthetic data augmentation, achieving improved recall and precision for underrepresented threats. These approaches showcase the growing trend toward integrating generative models with reinforcement learning to handle class imbalance and scarcity in UAV-based IDS systems.

Table 2.2 provides a comprehensive summary of the limitations observed in Reinforcement Learning (RL)-based Intrusion Detection System (IDS) approaches tailored for UAV security applications. It categorizes a wide range of studies based on the RL technique or scenario employed and outlines the key limitations reported in each case. Common challenges identified across the literature include dependency on offline training, lack of generalization to real-world UAV scenarios, limited transferability due to environment-specific tuning, and difficulties in acquiring large-scale labeled datasets. Furthermore, several methods assume perfect attacker knowledge or rely on predefined action categories, which may not hold in dynamic or adversarial settings. Hardware constraints, high communication loads, and high false positive sensitivity are also frequently cited issues. Overall, the table highlights that despite the growing interest in RL for UAV security, many proposed methods face significant barriers to practical deployment, emphasizing the need for more robust, scalable, and adaptable solutions in future research.

Collectively, the literature underscores the effectiveness of RL-based methods in enhancing UAV security by offering adaptability, robustness, and efficient resource management. However, challenges such as false alarms, policy overfitting, and real-world deployment constraints remain active areas for research and improvement.

Table 2.3 : Comparative analysis of state of the art in RL-based IDS frameworks for UAV and cyber-physical systems.

Study	RL Method	Environment	Dataset	Attack Type	Resource Efficiency	Generalization	Multi-Objective Reward
Lopez-Martin et al. [153]	DQN, DDQN, PG, AC	General NIDS	NSL-KDD, AWID	DoS, Probe, U2R, R2L	No	No	No
Hsu et al. [156]	DQN	Real-time IDS	Custom simulated	Generic attacks	No	No	No
Sethi et al. [157]	DQN	Cloud-based IDS	NSL-KDD	DoS, Probe	No	No	No
Yang et al. [160]	DQN + DCGAN + ResNet	Encrypted Traffic Detection	Encrypted traffic samples (IDX)	Encrypted malware	No	Yes	Yes
Khayat et al. [169]	DQN + CNN-LSTM Hybrid	IoT Network IDS	UNSW-NB15, CI-CIDS2017	Multi-class cyber attacks	Yes	Yes	No
Sethi et al. [165]	DQN + Attention + Multi-Agent	Enterprise IDS	NSL-KDD, CI-CIDS2017	Adversarial, Zero-day	Yes	Yes	No
Ren et al. [168]	Multi-Agent DRL + GCN	Cyber Network IDS	NSL-KDD	Various + Unknown attacks	No	Yes	No
Bouhamed et al. [173]	DQN	UAV Network IDS	Simulated UAV	General UAV Attacks	Yes	No	Yes
Soliman et al. [174]	RL + CNN (Navigation)	Autonomous UAV	Real flight test + Gazebo sim.	None (Nav-focused)	Yes	No	No
Masadeh et al. [177]	Q-Learning, SARSA	Target Detection in UAV	Simulated Search Area	Intruder detection	Yes	No	No
Strickland et al. [176]	DQN + GAN	NIDS	NSL-KDD	Minority Class Attacks	No	Yes	No
Wu et al. [166]	Two-layer Q-Learning	Internet of Drones	Simulated IoD data	Host/System attacks	Yes	No	No
AlKhonaini et al. [171]	REINFORCE HRL	UAV Detection	RF signals	UAV presence, type, mode	No	Yes	No
Arranz et al. (2025) [172]	PPO-based Hybrid DRL	UAV Swarm	Sim sensor/vision	Surveillance / tracking	Yes	Yes	No
Bouhamed et al. (2021) [173]	DQN	UAV IDS	Sim logs	Generic UAV attacks	Yes	No	Yes
Strickland et al. (2024) [176]	DQN + GAN	NIDS	NSL-KDD	Rare attacks	No	Yes	No
Our Framework (DQN + GenAI)	DQN	UAV Cyber Defense	UAV Attack Dataset [178]	Jamming, Spoofing	Yes	Yes	Yes

2.6 Positioning of the Thesis

This research introduces a comprehensive Generative Artificial Intelligence (GenAI) and reinforcement learning-enhanced intrusion detection framework specifically designed for securing Unmanned Aerial Vehicles (UAVs) against GPS spoofing and jamming attacks. While previous studies have effectively demonstrated the utility of GAN-based methods such as GUIDE [121], G-IDS [122], SeqGAN, and LeakGAN for enhancing IDS through synthetic data generation and adversarial training, our research extends beyond these approaches in several key ways. Unlike prior work that primarily focuses on a single generative model or application scenario, we systematically evaluate and compare multiple generative models (VAE, Gaussian Copula, CTGAN, and DDPM) to generate synthetic jamming and spoofing attacks specific to UAV systems. Furthermore, in contrast to frameworks like IDSGAN [123] and Diff-IDS [130] that either simulate adversarial traffic or apply a single model to imbalanced datasets, our work introduces a multidimensional evaluation framework that integrates statistical similarity measures (e.g., KL Divergence, Wasserstein Distance, KS Test) with IDS classification metrics to assess the fidelity and impact of synthetic data comprehensively.

Our inclusion of diffusion models (DDPM) for synthetic attack generation represents a novel contribution in UAV IDS research, building on recent advances but tailoring them specifically to the temporal and spectral complexities of UAV telemetry data. It also differs from previous studies in that it is explicitly applied to both jamming and spoofing attacks in the context of UAV systems, addressing a broader range of threat modalities that are often treated separately or overlooked in earlier research.

Importantly, our multi-train cross-testing strategy evaluates IDS generalization across different combinations of real and synthetic training sets, revealing the strengths and weaknesses of synthetic data under various deployment conditions. This novel integration of generative diversity, security-oriented data augmentation, and systematic cross-evaluation distinguishes our work from earlier GAN-based IDS research.

In addition to static classification using XGBoost, the framework introduces a Deep Q-Network (DQN)-based adaptive IDS module capable of real-time threat mitigation. The DRL component models UAV mission execution as a Markov Decision Process (MDP), allowing the agent to learn defense policies through interaction with telemetry states enriched by both real and synthetic data. This hybrid integration of GenAI-based data generation and DRL-based adaptive defense marks a novel direction in UAV security research, offering a scalable and resilient solution that addresses both the data scarcity and real-time response challenges prevalent in current IDS systems.

3. ATTACKS AND SCENARIOS

UAVs use a combination of communication networks, sensors, electronic receivers and transmitters, and software systems. UAVs collect and process data, share data over communication networks, control their flight, and perform specific tasks. These systems generate logs that contain substantial volumes of sensitive information, including location coordinates, video streams, sensor readings, and control commands. Attackers can exploit vulnerabilities in these systems to gain unauthorized access, manipulate data, disrupt communications, take control of systems, or cause physical damage.

Dataset contains the UAV's log records during the attack and the flights where no attack occurred. In the scenario implemented by Whelan et al., [178] the UAV must first perform a flight where no attack occurred. Benign flight can be performed in a controlled environment or in a non-hostile area such as a military base. After this flight is completed, it is recorded as benign flight logs. Their scenario used the Global Positioning System (GPS) for navigation of Unmanned Aerial Vehicles (UAVs) and the MAVLink protocol for communication with the Ground Control Station (GCS) [17]. These systems are vulnerable to certain threats, especially jamming and GPS spoofing. Spoofing disrupts communication or GPS signals, while GPS spoofing involves broadcasting false GPS signals to deceive the UAV's navigation system. Figure 3.1 illustrates a typical scenario involving a Ground Control Station (GCS) and an Unmanned Aerial Vehicle (UAV) communicating via the MAVLink protocol. The scenario involves a UAV that communicates with a GCS using the MAVLink protocol, a widely used messaging protocol for UAVs.

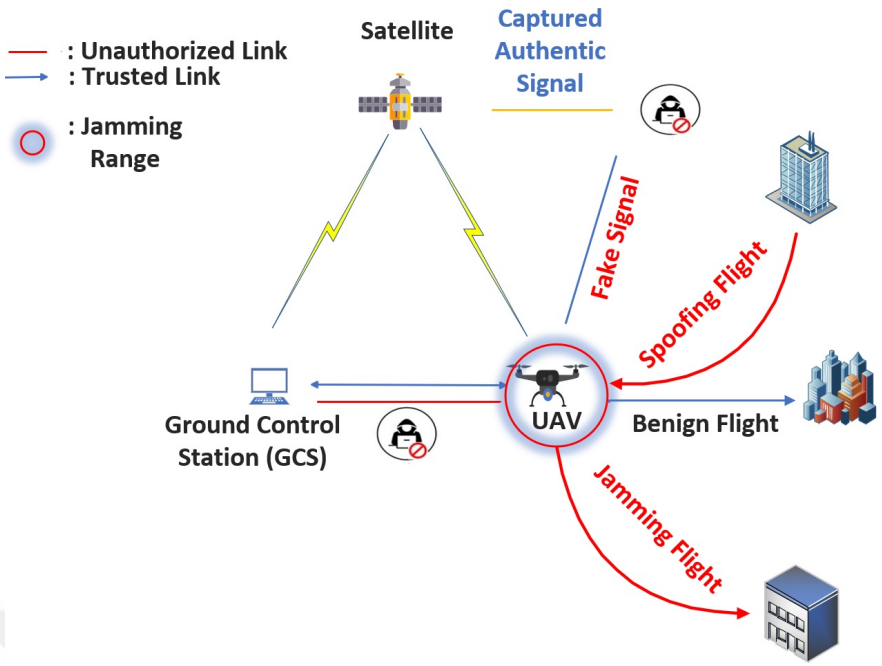

Figure 3.1 : An attack scenario.

MAVLink is a lightweight protocol designed for communication links with limited bandwidth and systems with constrained resources, which are used to carry telemetry and command and control data for numerous small unmanned aerial vehicles [179].

MAVLink protocol defines the mechanism on the structure of messages and how to serialize them at the application layer. These messages are then forwarded to lower layers, such as transport layer and physical layer, to be transmitted to the network. The advantage of MAVLink protocol is that it supports different types of transport layers and mediums thanks to its lightweight structure. It can be transmitted through WiFi, Ethernet, or serial telemetry low bandwidth channels.

Figure 3.2 describes the structure and meaning of each component in a MAVLink frame, which is a communication protocol widely used in UAVs. The first byte is the "Packet Start Sign" (STX), which has a fixed value of 0xFE and signals the beginning of a new packet. The "Payload Length" (LEN) indicates the size of the following payload, ranging from 0 to 255 bytes. The "Packet Sequence" (SEQ) is used for tracking the sequence of packets to help detect any packet loss. The "System ID" (SYS) and "Component ID" (COMP) identify the sending system and its specific component, allowing for differentiation between multiple platforms and components on the same network. The "Message ID" (MSG) denotes the type of message, which determines the payload's meaning and how to interpret it. The "Data" (Payload), ranging from 0 to 255 bytes, starts depends on the message ID. Lastly, the checksum ensures data integrity by computing a hash of the previous bytes, incorporating an extra parameter to enhance error-checking. This structured format helps MAVLink achieve reliable and organized communication in real-time applications [180].

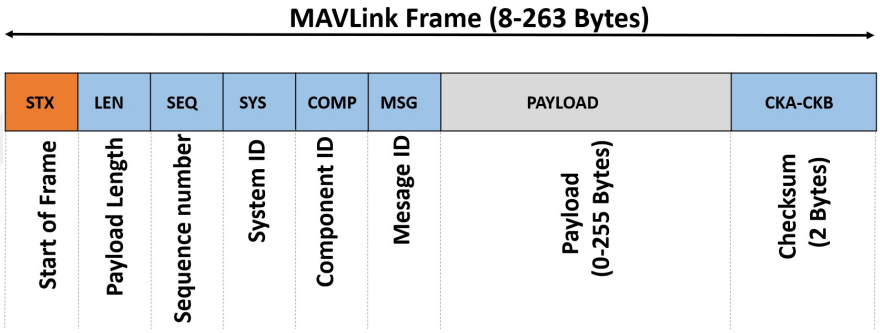

Figure 3.2 : MAVLink Protocol Data Frame.

MAVLink protocol has no privacy or authentication mechanism. Therefore, it does not provide any security and can be hacked quite easily. GCS communicates with drones over an unauthenticated and unencrypted channel. Since MAVLink message streams are sent without encryption, anyone with a suitable transmitter can communicate with the drone, intercept communication, eavesdrop, and inject commands. Thus, any attack can be launched easily. In [181] discusses the vulnerabilities of MAVLink protocol and proposes a security-integrated mechanism for MAVLink that ensures the protection of MAVLink messages exchanged between UAVs and GCSs by utilizing the use of encryption algorithms.

A Holybro S500 UAV equipped with a Pixhawk 4 flight controller was used to explore UAV intrusion detection under GPS spoofing and jamming attacks in [17]. A HackRF software-defined radio (SDR) performed the attacks, while a Keysight EXG signal generator provided controlled GPS signals simulating a location. The UAV first performed a harmless flight to collect basic data. During the GPS spoofing attack, the GPS-SDR-SIM generated false GPS data streams, causing the UAV to lose its balance and crash. Similarly, in the GPS jamming attack, white Gaussian noise corrupted the UAV's GPS, again leading to a crash. Observations from these harmless and harmless flights were recorded for further analysis. The log records of all flights were shared openly, shedding light on our work. We generated synthetic data for both benign and attack classes using generative AI, based on the data collected in the attack scenario described above. We then compared the statistical properties and intrusion detection performance of our generated dataset with the real dataset from [17].

4. A GEN-AI FRAMEWORK FOR GENERATING SYNTHETIC ATTACKS ON UAVS

In this chapter, we propose a framework to generate synthetic attack data for UAVs. The framework consists of two parts. Synthetic attack data is generated with generative models in the first part with models that are VAE, Gaussian Copula, CTGAN and DDPM. In the second part, an intrusion detection algorithm is applied to datasets generated in the first part. XGBoost classification is used to distinguish attack data and benign data. The proposed framework is shown in Figure 5.2.

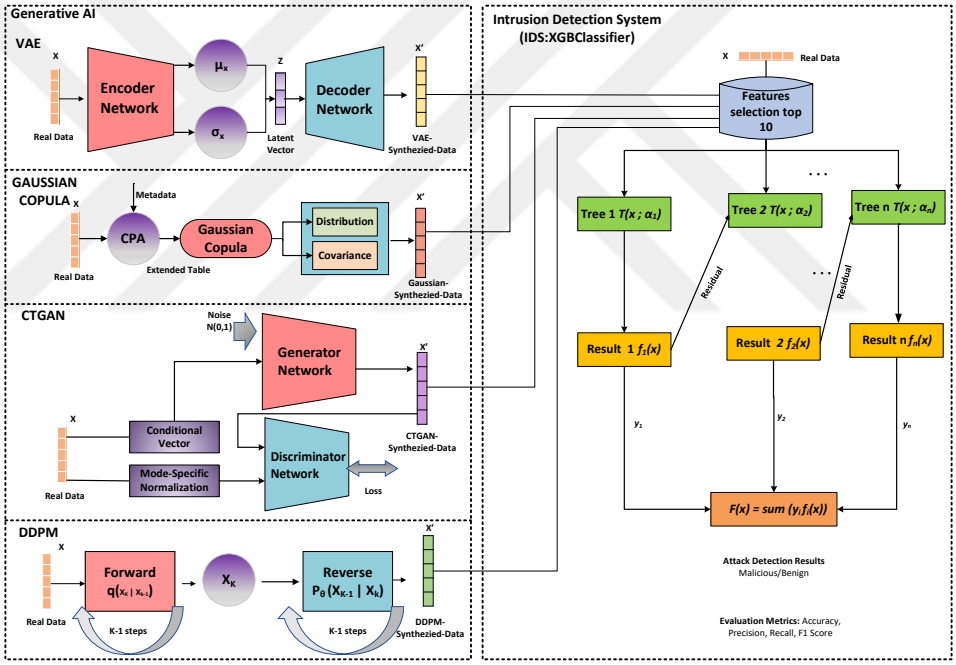

Figure 4.1 : Framework for Generating Synthetic Attack Data Using GenAI Models.

4.1 GenAI Models

To generate robust and diverse synthetic data for jamming and spoofing attacks, we employed four complementary generative models: VAE, Gaussian Copula, CTGAN, and DDPM. Each model contributes uniquely to capturing statistical structures, temporal behaviors, and multimodal feature dependencies in UAV attack scenarios.

4.1.1 Variational autoencoders

We investigated Variational Autoencoders as a method of generating synthetic attack data to train our models.

An autoencoder (VAE) is comprised of two principal components: the encoder and the decoder.

The encoder compresses the input data x into a latent representation z. The output of the encoder is a pair of parameters, namely the mean (μ) and the standard deviation (σ), which together define a Gaussian distribution. The latent vector z is sampled from this distribution, ensuring that the latent variables follow a continuous, typically normal, distribution. This allows the model to capture essential information in a lower-dimensional space.

The latent vector z represents the compressed knowledge of the input, holding its critical features in a reduced form. It acts as the input for the decoder, enabling data reconstruction.

The decoder reconstructs the input data from the latent vector z. Its objective is to minimize the reconstruction error, ensuring that the reconstructed data is as close as possible to the original input. This demonstrates the effectiveness of compressed representation in terms of accuracy.

4.1.2 Gaussian copula

We also used Gaussian copulas to generate synthetic attacks by identifying the dependency between variables in the dataset. A copula can be defined as a multivariate distribution, and the use of a Gaussian copula enables the accurate modeling of the correlations between different variables. In the initial phase, the conditional probability approach is employed to integrate metadata with the features in question. This phase employs contextual insights derived from the metadata to modify or enhance the data potentially. The result of this process is an "extended table," which is subsequently transformed using a Gaussian copula. The Gaussian copula, a statistical method designed to model dependencies among multiple variables, serves to prepare the data for analysis in a more sophisticated model. This advanced model is designed to examine the data's distribution and covariance, thereby facilitating a more profound understanding of the subject matter [182].

4.1.3 Conditional tabular GAN

We investigated the potential of Conditional Tabular GAN (CTGAN) as a way to generate synthetic attack data for training models. The conditional generator is an effective tool for the generation of synthetic rows, whereby conditioning is applied to a specific discrete column. By employing training-by-sampling, both the conditioning values and the training data are sampled based on the logarithmic frequency of each category [183]. This ensures that CTGAN effectively explores all potential discrete values, allowing the modelling of complex relationships in the data that are necessary to simulate realistic attack scenarios. CTGANs are particularly effective in handling a variety of conditions found in real-world data [184].

The process begins by sampling a condition, which is then passed to the conditional generator G along with a noise vector z drawn from a standard normal distribution $N(0,1)$ The generator creates a synthetic sample, which is compared against a randomly selected dataset sample that satisfies the same condition. This comparison is evaluated by the conditional discriminator D. To capture all potential correlations between the columns, both the generator and discriminator use fully connected neural networks with two hidden layers each. The generator employs batch normalization and the ReLU activation function, generating synthetic row representations through a mixture activation function after the hidden layers. The discriminator uses the leaky ReLU activation function and applies dropout in each hidden layer to enhance robustness. CTGAN enhances TableGAN (TGAN) by modifying the loss function and normalization techniques to handle imbalanced data, particularly for continuous data with non-Gaussian or multimodal distributions.

4.1.4 Denoising diffusion probabilistic model

Finally, we also used a Denoising Diffusion Probabilistic Model (DDPM) to generate synthetic attack data. In generally diffusion models are a type of latent variable model in machine learning that leverage Markov chains and variational inference to uncover the hidden structure of a dataset. These models learn the data distribution by progressively adding noise to the data and subsequently denoising it, allowing them to generate high-quality and diverse samples.

We used a diffusion model that operates in two main stages: forward process (diffusion) and reverse process (denoising).

For the DDPM [185,186] in the forward process, the original data $(X_0 \in \mathbb{R}^d)$ is progressively corrupted by adding Gaussian noise (ε)step by step, resulting in a sequence of latent variables $(X_1, \ldots, X_K)$. This process transforms the data into pure Gaussian noise, where the final state $(X_K \sim \mathcal{N}(0, I))$. Each transition in this sequence follows a Markov Chain. The reverse process then reconstructs the original data by iterative denoising of the Gaussian noise.

4.1.4.1 Algorithms of GenAI models

Algorithm 1 outlines the proposed GenAI model for generating synthetic jamming and spoofing attacks in UAVs. The algorithm consists of three main steps. In the first step, training a VAE model to generate synthetic attacks is accomplished. Then, we train a Gaussian Copula model to generate synthetic attacks. Finally, we train a CTGAN model to generate synthetic attack, where x_i represents the i-th feature and N is the total number of features and outputs the final generated synthetic attack data for each model.

In step 1, the Variational Autoencoder (VAE) uses the Kullback–Leibler (KL) divergence metric [187], which measures the difference between the learned latent distribution and a target Gaussian distribution. This encourages the latent space to align with the standard Gaussian prior $\mathcal{N}(0,1)$, promoting regularity and smoothness. Specifically, the KL divergence term quantifies how much the encoder's learned distribution $(q(z|x))$ deviates from the prior, as shown in Eq. 4.1.

Unlike standard autoencoders that map the input x to a single deterministic latent vector, VAEs instead map x to two separate vectors representing the mean (μ) and standard deviation (σ) of a Gaussian distribution. This probabilistic encoding ensures a smooth and continuous latent space, enabling better generalization and generative capabilities [188].

$$L_{\mathrm{KL}} = D_{\mathrm{KL}}(q(z|x) \,\|\, \mathcal{N}(0,1)) \tag{4.1}$$

$$L_{\mathrm{total}} = L_{\mathrm{KL}} + \|x - \tilde{x}\|^2 \tag{4.2}$$

Algorithm 1 Generative Artificial Intelligence for Generated Synthetic Attacks in UAVs

Input: Training data $x_1, x_2, ..., x_n \in X$
Output: VAE-synthezied-data, GaussianCopula-synthezied-data, CTGAN-synthezied-data, DDPM-synthezied-data
Step 1:Training a VAE model to generating synthetic attacks
Initialize decoder parameters ϕ, encoder parameters θ
repeat
  for $k = 1$ to N **do**
    Draw samples S from $\varepsilon \sim \mathcal{N}(0,1)$
    Compute latent variable: $z_{(k,s)} = h_\phi(\varepsilon^{(k)}, x^{(k)})$
  Compute reconstruction loss: $E = \|x - \tilde{x}\|^2$
  Compute KL divergence: $L_{\text{KL}} = D_{\text{KL}}(q(z|x) \,\|\, \mathcal{N}(0,1))$
  Combine losses: $L_{\text{total}} = L_{\text{KL}} + E$
  Update parameters ϕ, θ using Stochastic Gradient Descent
until Parameters ϕ, θ converge
Step 2: Training a Gaussian Copula model to generate synthetic attacks
for each feature $(x_i \in X)$ **do**
  Estimate the marginal distribution of each feature.
  Transform each feature into a standard normal distribution using CDF.
  Compute the covariance matrix of the transformed features.
  Generate synthetic samples from a multivariate normal distribution using the learned covariance matrix.
  Apply the inverse CDF transformation of each feature to map the synthetic data back to the original space.
Combine synthetic features.
Step 3: Training a CTGAN model to generate synthetic attacks
for each feature $(x_i \in X)$ **do**
  Sample noise vector from a standard normal distribution.
  Sample conditional vector representing feature distributions.
  Generate synthetic samples using the generator.
  Discriminate real and synthetic samples using the discriminator.
  Compute generator loss as Wasserstein Loss : $\mathscr{L}(D,G) = \mathbb{E}_{x\sim p_r}\big[D(x)\big] - \mathbb{E}_{z\sim p_g}\big[D(G(z))\big]$
Step 4: Training a DDPM model to generate synthetic attacks:
for each feature $(x_i \in X)$ **do**
  Add Gaussian noise at each step.
  Train a neural network to predict the noise.
  Minimize the loss function.
$L_{\text{simple}} = \mathbb{E}_{k,x_0,\varepsilon}\left[\|\varepsilon - \varepsilon_\theta(x_k, k)\|^2\right]$
  Sample
  Iteratively denoise using the reverse process
  Denormalize the generated data to match the original data distribution
Step 5: Output:
Return VAE-synthezied-data, GaussianCopula-synthezied-data, CTGAN-synthezied-data, DDPM-synthezied-data

Variational Autoencoder (VAE) loss function ensures the model to learn meaningful latent representations while accurately reconstructing input data. The 4.1 equation, calculates the Kullback-Leibler (KL) divergence, which measures how much the learned latent distribution deviates from the desired prior distribution, typically a standard normal distribution $\mathcal{N}(0,1)$. This regularization term enforces structure and continuity in the latent space, enabling smooth interpolation and meaningful sampling. Equation 4.2 combines the regularization term with the reconstruction loss that is represented as the squared difference between the input x and its reconstruction $\tilde{x}$. This combined objective ensures that the VAE not only produces reconstructions that are faithful to the original input but also learns a well-regularized and structured latent representation. The KL term shapes the latent space, while the reconstruction term preserves input fidelity. Together, these losses guide VAE during training, balancing accurate data reconstruction and a robust latent representation [189].

The training of a Gaussian Copula model to generate synthetic attacks is described in step 2. Each feature is transformed into a uniform distribution through the application of cumulative distribution functions (CDFs), and is then further standardized to a normal distribution through the use of the inverse of the standard normal CDF. A correlation matrix is then derived from the transformed variables in order to capture the relationships between them. The matrix is then employed to generate new samples that are drawn from a multivariate normal distribution. Subsequently, the samples are mapped back to the original variable scales through inverse transformations. Initially, they are smoothed with the normal CDF, and then reverted to the original distributions using the inverse CDFs of the fitted models [190], [191].

The training of a Gaussian Copula model to generate synthetic attacks is described in step 3.

The CTGAN loss function was optimized using the Adam algorithm. For CTGAN, the training process leverages the Wasserstein loss function, which provides a stable and effective objective for generative adversarial learning in the equation 4.3. In this equation, D represents the discriminator, G denotes the generator, z is a sample from the generator's input noise distribution p_g, and x is a sample from the real data distribution p_r. This approach offers several advantages over the standard GAN loss: it provides more informative and smooth gradients, reduces training instability, and mitigates issues like mode collapse. Most importantly, it enables the generator to iteratively improve in producing synthetic data that closely approximates the real distribution in both marginal and joint feature spaces. By minimizing this divergence between real and generated distributions, the Wasserstein loss ensures that the synthetic data retains the structural and statistical properties of the real dataset.

$$\mathscr{L}(D,G) = \mathbb{E}_{x\sim p_r}\big[D(x)\big] - \mathbb{E}_{z\sim p_g}\big[D(G(z))\big] \tag{4.3}$$

The training of a DDPM model to generate synthetic attacks is explained in step 4. Finally, outputs for each model are recorded in last generated synthetic attack data. The loss function (in equation 4.4) L_{simple} is a fundamental component of DDPM models, designed to optimize the model's ability to reconstruct data by accurately predicting the noise introduced during the forward diffusion process. It is defined as the expected mean-squared error (MSE) between the true noise ε and the predicted noise $\varepsilon_\theta(x_k,k)$ where the prediction is performed by a neural network parameterized by θ [185], [192].

$$L_{\text{simple}} = \mathbb{E}_{k,x_0,\varepsilon}\left[\|\varepsilon - \varepsilon_\theta(x_k,k)\|^2\right] \tag{4.4}$$

Let (x_k) denote the input at time step (k), and let $(\varepsilon \sim \mathscr{N}(0,1))$ represent the Gaussian noise sampled from a standard normal distribution. The expectation $(\mathbb{E}_{k,x_0,\varepsilon})$ is computed over the timestamps (k), real data samples (x_0), and Gaussian noise (ε). By minimizing this loss, the model learns to predict the noise for any noisy input (x_k) at a given timestep (k). This is crucial for the reverse diffusion process, where the model progressively removes noise to reconstruct the original data.

4.2 Tree-based Intrusion Detection System

We tested generated synthetic attack data by using intrusion detection systems for UAVs. We use XGBoost algorithm for the intrusion detection system. XGBoost, a highly scalable tree boosting system, was introduced by Chen et. al. [193] as an extended version of the Gradient Boosting Decision Tree (GBDT) algorithm [194].

To reduce the dimensionality of the model while preserving informative features, we used the Gini-based gain importance, which measures the average gain across all splits where a feature is used. For each feature (x_i), the gain is calculated as in 4.5, where $T(x_i)$ is the set of trees where (x_i) is used, and $\Delta\mathscr{L}_t(x_i)$ is the reduction in the loss function (e.g., logistic/log-loss or squared error) when splitting on feature (x_i) at node t. Based on the computed gain scores, features were ranked in descending order of importance. The top ten most important features were selected and used to train the final IDS model, ensuring a balance between performance and computational efficiency.

$$\text{Gain\_Based\_Importance}(x_i) = \sum_{t \in T(x_i)} \frac{1}{|T(x_i)|} \Delta\mathscr{L}_t(x_i) \tag{4.5}$$

Algorithm 2 Feature Importance Extraction via XGBoost

1: **Input:** Raw feature set $\mathscr{X} = \{x_1, x_2, \ldots, x_n\}$ from training data
2: **Output:** Top-k features $\mathscr{X}_{\text{top}}$
3: Compute gain-based importance score $\text{Gain}(x_i)$ for each feature x_i
4: Sort all features in descending order of gain: $\text{Sort}(\mathscr{X}) \rightarrow \mathscr{X}'$
5: Select top-10 features: $\mathscr{X}_{\text{top}} \leftarrow \mathscr{X}'[1:10]$
6: Return $\mathscr{X}_{\text{top}}$

The pseudocode for the feature ranking and selection process is provided in Algorithm 2. According to the ranked list of features, the top ten features were selected for the training of the IDS model.

XGBoost operates under the principle of gradient boosting, which consists of three primary stages. The initial step is to identify a differentiable loss function that is appropriate for the problem at this stage of the process. One advantage of gradient boosting is its flexibility, in that new algorithms do not need to be derived for different loss functions. Instead, an appropriate loss function can be selected and integrated into the framework. Secondly, a weak learner is utilized to make predictions, with decision trees being the optimal choice. In particular, regression trees are utilized as weak learners, as they generate real-valued outputs for splits, thereby allowing their outputs to be aggregated. This additive property enables the model to iteratively enhance the accuracy of residuals. The trees are constructed in a greedy manner, frequently with constraints that guarantee they remain weak learners while maintaining computational efficiency. Thirdly, an additive model is constructed by sequentially combining the outputs of the weak learners in order to minimize the loss function. New trees are added to the sequence one at a time, with each tree's output refining the predictions of the previous ones. This process continues until the loss function is sufficiently optimized. XGBoost builds upon gradient boosting but introduces a significant enhancement. In contrast to traditional gradient boosting, where weak learners are added sequentially, XGBoost employs a multi-threaded approach.

A framework for evaluating attack detection using XGBoost with both real and synthesized datasets is shown in Figure 5.2. From each dataset, the top 10 most important features are selected for training purposes. The XGBoost is then used to model the data. The algorithm employs a sequence of decision trees, wherein each tree refines the residual errors of the previous one. The trees are then combined into an additive model, whereby the final prediction is the weighted sum of the outputs of all trees.

Algorithm 3 IDS for Synthetic and Real Attacks on UAVs

Set initial values for the parameters of the XGBoost classifier
for each instance f_i in F **do** $x_i \leftarrow$ SortFeatures(f_i) Sort features based on feature importance score $G(x_i)$
    $p_1 \leftarrow \arg\min_{s \in S} \mathscr{L}_{\text{gain}}(s, f_i)$ Select split point s with minimum gain-based loss
    Optimize objective target function to compute the tree depth and choose the best split point
    $M \leftarrow$ Optimizing(Choosing descriptor-point(p_i))
    Tree-leaves $\leftarrow$ Prediction-score$(M,$Tree-leaves$)$
    Bottom-up-Prune-negative-nodes $(M,$Tree-leaves$)$
    Repeat
    Repeat steps 3 and 4 until Length$(M) =$ Max-tree-depth
    Repeat steps 3 to 10 until cumulative training includes all trees in C
    Until
    Return M
    Step 4: Evaluation:
    Evaluate final M model using test features.

The framework is designed to predict whether an input sample is "malicious" or "benign." Its performance is evaluated using a range of metrics, including accuracy, precision, recall and F1 score. This configuration enables a comparison of the quality of synthetic data to support effective attack detection relative to the real dataset.

A method for training an intrusion detection system based on the XGBoost classifier, specifically designed to address synthetic and real attacks on UAVs is described in Algorithm 3. It takes the top 10 most important features as inputs, which are extracted from both synthetic and real datasets, along with hyperparameters such as the number of trees and other classifier settings. Initially, the parameters of the XGBoost classifier are set. For each feature in the dataset, the algorithm sorts the feature values, selects the optimal split point using a lowest-gain-first approach, and optimizes the objective target function to determine the tree depth and split point. Subsequently, the tree structure is updated by optimizing and pruning unnecessary nodes based on their contribution to the prediction task. This process iterates until the maximum tree depth is reached. The cumulative training is repeated for all trees in the dataset, refining the model. Finally, the trained model is returned and evaluated using test data, completing the IDS training process. This approach ensures an efficient and optimized classifier that effectively detects synthetic and real attacks.

5. DEEP REINFORCEMENT LEARNING BASED INTRUSION DETECTION SYSTEMS

This chapter presents the experimental framework for our Deep Q-Learning (DQN)-based intrusion detection system (IDS) for unmanned aerial vehicles (UAVs). The proposed architecture is designed to detect and mitigate both real and synthetic adversarial GPS jamming and spoofing attacks. A distinctive feature of our setup is the integration of synthetic attack scenarios generated using VAE, Gaussian Copula, CTGAN, and DDPM. These synthetic threats complement real-world attack telemetry to train and evaluate a robust reinforcement learning agent capable of adaptive decision-making.

Reinforcement learning describes machine learning algorithms that try to find optimal behavior in an environment that maximizes the cumulative reward. A common structure of reinforcement learning for UAVs is shown in Figure 5.1. The decision-making process controls the environment that is often considered a Markov Decision Process (MDP) [18]. An MDP consists of a set of states S, which may be finite, infinite, or continuous, a set of executable actions, which can be discrete or continuous, a transition probability, which describes the dynamics of an agent interacting with its environment, and a reward function at a given initial, final state after taking a certain action.

Classic RL algorithms have limitations when large-scale problems need to be solved with RL algorithms. For example, many classic RL algorithms require maintaining a value look-up table for each state or a state-action pair [18]. Thanks to the recent advances in computational power, Deep Learning (DL) has overcome this limitation. Most of the time, RL algorithms use Temporal Difference Learning and Q-learning for Actor-Critic models.

Temporal Difference (TD) learning consists of a transition model and a reward function with policy evaluation [195]. The state value function of MDP, $(V_\pi(s))$, is calculated as follows.

$$V_\pi(s) = E[\sum_{t=0}^{\infty} \gamma^t R(s_t)|s_0 = s, \pi] \tag{5.1}$$

In the standard dynamic programming approach, $V_\pi(s)$ is computed as follows.

$$\forall s: \;\; V(s) \leftarrow R(s) + \gamma \sum_{s'} P(s'|s,a)V(s') \tag{5.2}$$

A stochastic version of the state function is as follows.

$$\text{pick some state } s: \;\; \text{sample} \;\; s' \sim P(s'|s, \pi(s)) \tag{5.3}$$

$$V_\pi(s) \leftarrow \alpha(R(s) + \gamma V'_\pi(s)) + (1-\alpha)V_\pi(s) \tag{5.4}$$

where α is the step size. A possible step size is $\alpha_k = \frac{1}{k}$ for the k'th time for an update.

Convergence is the stochastic version of policy evaluation that converges to the true value function under certain assumptions. The following assumptions are sufficient conditions.

- every state is often visited infinitely.
- α satisfies $\sum_{k=0}^{\infty} \alpha_k = \infty$; $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$.

In practice, a reason to use TD for policy evaluation could be that we do not have the transition model available. The samples are then generated by executing the policy and by performing stochastic value function updates according to the states that are visited.

TD only considers policy *evaluation.* If we are interested in finding a near-optimal policy, we can use TD as the policy evaluation step in *policy iteration* as follows.

Iterate:

- Run TD to perform policy evaluation, which gives us $V_\pi(s)$, $\forall s$.

- Pick a new policy π, such that $\pi(s) = \arg\max\ [R(s) + \gamma \sum_{s'} P(s'|s,a)V(s')]$

Iterate:

$$\forall s,a : Q(s,a) \leftarrow R(s) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q(s',a') \tag{5.5}$$

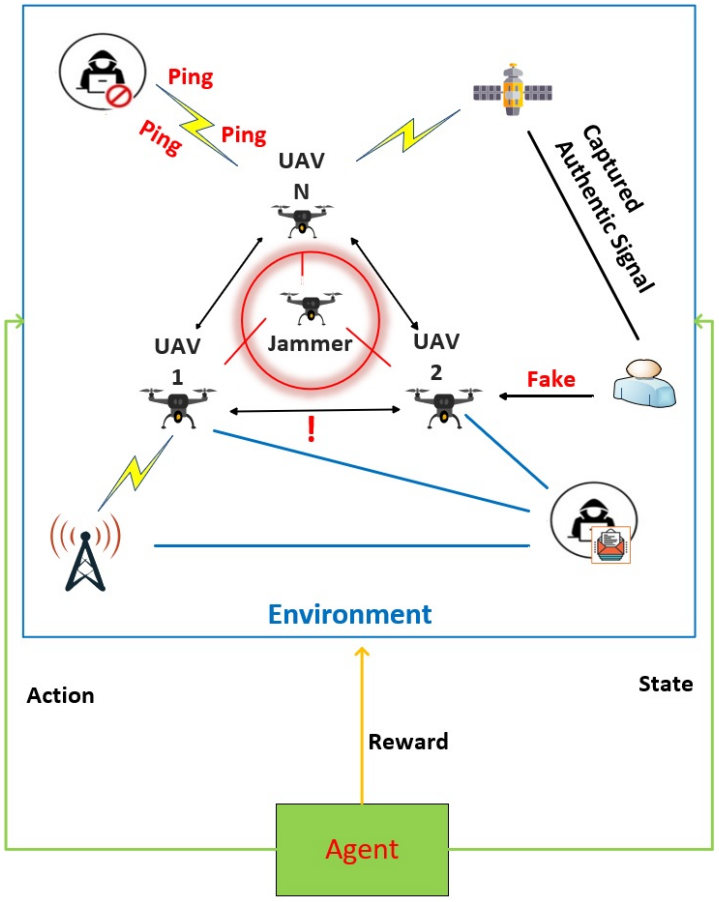

Figure 5.1 : General structure of reinforcement learning for UAV security.

Similarly to TD learning, we can run a stochastic version instead for k = 0, 1, 2, ... as follows.

$$Q(s,a) \leftarrow (1-\alpha_k)Q(s,a) + \alpha_k(R(s) + \gamma \sum_{s'} \max_{a'} Q(s',a')) \tag{5.6}$$

The stochastic version does not require a priori knowledge of the transition probability distribution P or the reward function R. It is sufficient to have access to a trace. The stochastic version may converge to the true Q function under the same assumptions as in TD.

- every state is often visited infinitely.
- α satisfies $\sum_{k=0}^{\infty} \alpha_k = \infty$; $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$.

In practice, one can ensure that all reachable states are visited infinitely often by using an ε greedy policy;

$$\text{with probability } \varepsilon : \text{choose an action at random}$$
$$\text{with probability } 1-\varepsilon : a = \arg\max_a Q(s,a)$$

Another popular way to choose the actions is computed as follows.

$$\text{pick action } a \text{ with probability} \frac{\exp(-Q(s,a)/T)}{\sum_b \exp(-Q(s,b)/T)} \tag{5.7}$$

In equation 5.7, T is the temperature and it determines how greedily the actions are being chosen. A natural choice decreases the temperature over time. Note that the temperature T and factor α need not be the same for all states. For instance, one could have these variables that depends on how often the current state s has been visited.

Q-Learning is one of the commonly used algorithms to solve the MDP problem. The actions are obtained for every state that is based on an action-value function. *Q(s, a)* is defined with the value of the state (s) and action (a) pair for attacks in communication networks. *R(s)* is the reward function of the current state. *P(s'|s, a)* is defined as the probability of the transition from the actual state-action pair to the next attacking state in a threat condition. *V(s')* is the value of the next state. *Q(s', a')* can be defined as the value of the next state (s') and next action (a').

$$Q(s,a) = R(s) + \gamma \sum_{s'} P(s'|s,a)V(s') \tag{5.8}$$

We can run dynamic programming, such as value iteration, by performing the Bellman back-ups in terms of Q function as follows [18].

$$Q(s,a) \leftarrow R(s) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q(s',a') \tag{5.9}$$

Deep Q Networks When states are discrete, Q-function can be easily formulated as a table. This formulation becomes harder when the number of states increases, and may be impossible when the states are continuous. In such a case, the Q function is formulated as a parameterized function of the states, actions pairs *Q(s; a;w)*. The solution is required to find the best setting for the parameter w. Using this formulation, it is possible to approximate Q-function by using a Deep Neural Network (DNN). In this setup, the problem will be an optimization problem. DNN may minimize the Mean Square Error (MSE) of Q-values using Gradient-based methods Stochastic Gradient Descent (SGD) [196].

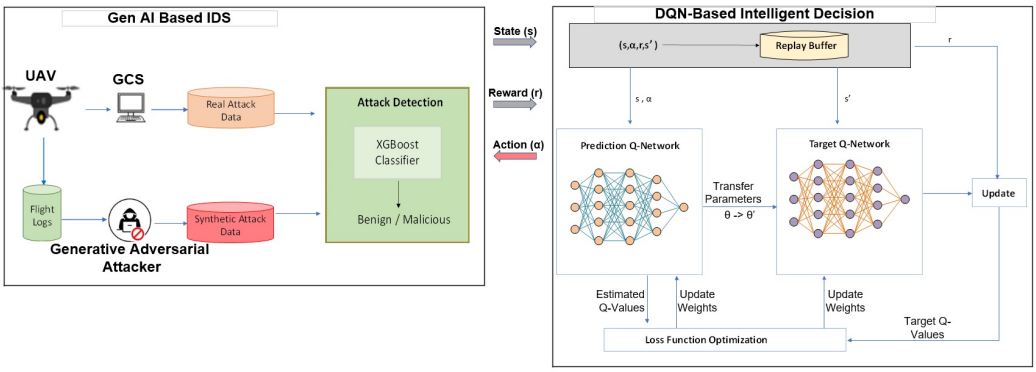

Figure 5.2 : Overall system architecture showing synthetic attack generation, XGBoost-based detection, and DQN-based reinforcement learning.

Figure 5.2 presents the overall architecture of the proposed UAV intrusion detection system, which integrates **Generative Adversarial Data Augmentation** with a **Deep Q-Network (DQN)-based Intrusion Detection System (IDS)** for adaptive threat mitigation. The system is composed of two primary stages: *Real and Synthetic Attack Data Detection* and *DQN-Based Intelligent Decision*.

In the first stage (left panel), real communication data between the UAV and Ground Control Station (GCS) is collected and labeled as *real attack data*. Simultaneously, a *Generative Adversarial Attacker* module synthesizes *synthetic attack data* by learning from flight log patterns. Both real and synthetic datasets are used to train an **XGBoost classifier**, which labels telemetry instances as either benign or malicious. This classification result forms part of the observation state used by the DQN agent.

In the second stage (right panel), the DQN agent interacts with the UAV environment modeled as a Markov Decision Process (MDP). At each time step, the agent receives a state s consisting of features such as GPS integrity, classifier outputs, motion dynamics, and battery level. Based on this input, the agent selects an action α (e.g., NoAction, Alert, IMU mode, Throttle, RTH) and receives a reward r according to the designed multi-objective reward function. The experience tuple (s, α, r, s') is stored in a **Replay Buffer** to enable off-policy learning.

The learning mechanism employs two neural networks: the **Prediction Q-Network**, which estimates the current Q-values and updates its parameters via loss minimization, and the **Target Q-Network**, which generates stable target Q-values and is periodically synchronized with the prediction network ($\theta \rightarrow \theta'$). This setup stabilizes the learning process and improves convergence.

Through continuous interaction and learning, the DQN agent develops effective real-time defense policies, including triggering critical safety maneuvers such as Return-to-Home (RTH) or communication throttling. This hybrid architecture significantly enhances the UAV's resilience to cyber-physical threats by combining the strengths of generative data augmentation and deep reinforcement learning.

The proposed UAV Intrusion Detection System (IDS) integrates deep reinforcement learning with generative data augmentation to enhance both security and adaptability during mission execution. At the core of the system lies a Deep Q-Network (DQN) agent augmented with an LSTM layer, enabling it to capture temporal dependencies in sequential UAV telemetry data. The system begins by extracting meaningful features from telemetry, such as GPS integrity, battery level, and motion dynamics, alongside threat probability indicators from a pretrained XGBoost classifier. To address data scarcity and improve generalization, synthetic attack samples generated by models such as VAE, Gaussian Copula, CTGAN, and DDPM are incorporated during training.

Figure 5.3 illustrates the full pipeline of this IDS architecture. The extracted state representation is passed through a fully connected (FC) layer of 128 units and then processed by a 128-unit LSTM layer, which helps retain contextual information over time. The final output layer provides a Q-value for each of the five available actions:

No Action, Alert GCS, Switch to IMU, Throttle Communication, and Return-to-Home (RTH). Actions are selected based on an ε-greedy exploration strategy. The agent is trained using a multi-objective reward function that accounts for detection accuracy, battery usage, action cost, GPS reliability, and mission progress. This holistic design empowers the UAV to respond adaptively to jamming and spoofing attacks in real-time, maintaining both operational continuity and cybersecurity resilience.

Algorithm 4 DQN Training Procedure for UAV Intrusion Detection

Require: Environment $\mathscr{E}$, Q-network Q, target network Q', replay buffer $\mathscr{D}$, learning rate α, discount factor γ, exploration rate ε

Initialize Q, set $Q' \leftarrow Q$, initialize buffer $\mathscr{D}$
for each episode $e = 1$ to N **do**
    Reset environment and get initial state s_0
    Initialize LSTM hidden state h_0
    for each step $t = 0$ to T **do**
        Select action a_t using ε-greedy policy on Q
        Execute a_t in $\mathscr{E}$; observe r_t, s_{t+1}, d_t
        Store $(s_t, a_t, r_t, s_{t+1}, d_t)$ in $\mathscr{D}$
        if $|\mathscr{D}| >$ batch size **then**
            Sample mini-batch from $\mathscr{D}$
            Compute target:

$$y_i = \begin{cases} r_i & \text{if } d_i = \text{True} \\ r_i + \gamma \max_{a'} Q'(s_{i+1}, a') & \text{otherwise} \end{cases}$$

            Perform gradient descent on loss $\mathscr{L} = (y_i - Q(s_i, a_i))^2$
        if d_t is True **then**
            break
    Periodically update $Q' \leftarrow Q$
=0

This transition model allows the DQN agent to learn policies that are robust to sensor noise, probabilistic threats, and energy constraints, while explicitly modeling the operational effects of defense actions.

The training procedure for the proposed Intrusion Detection System (IDS) is based on a Deep Q-Network (DQN) framework adapted for UAV environments. As outlined in Algorithm 4, the agent interacts with a custom-designed UAV environment that models both normal and adversarial network conditions, such as jamming and spoofing attacks.

At the start of each episode, the environment is reset and the initial state is observed. The agent selects actions using an ε-greedy policy to balance exploration and exploitation. After executing an action, the resulting transition comprising the current state, selected action, reward, next state, and done flag is stored in a replay buffer.

If the buffer contains enough samples, a mini-batch is randomly drawn to perform gradient-based updates on the Q-network. The target Q-values are computed using a separate target network to stabilize learning. The Q-network parameters are periodically copied to the target network. This iterative process allows the agent to learn optimal defense strategies that minimize false alarms and missed detections while preserving system resources such as battery life and communication bandwidth. The integration of LSTM in the Q-network further improves temporal awareness, allowing the agent to adaptively handle sequences of observations and detect time-dependent anomalies in UAV behavior.

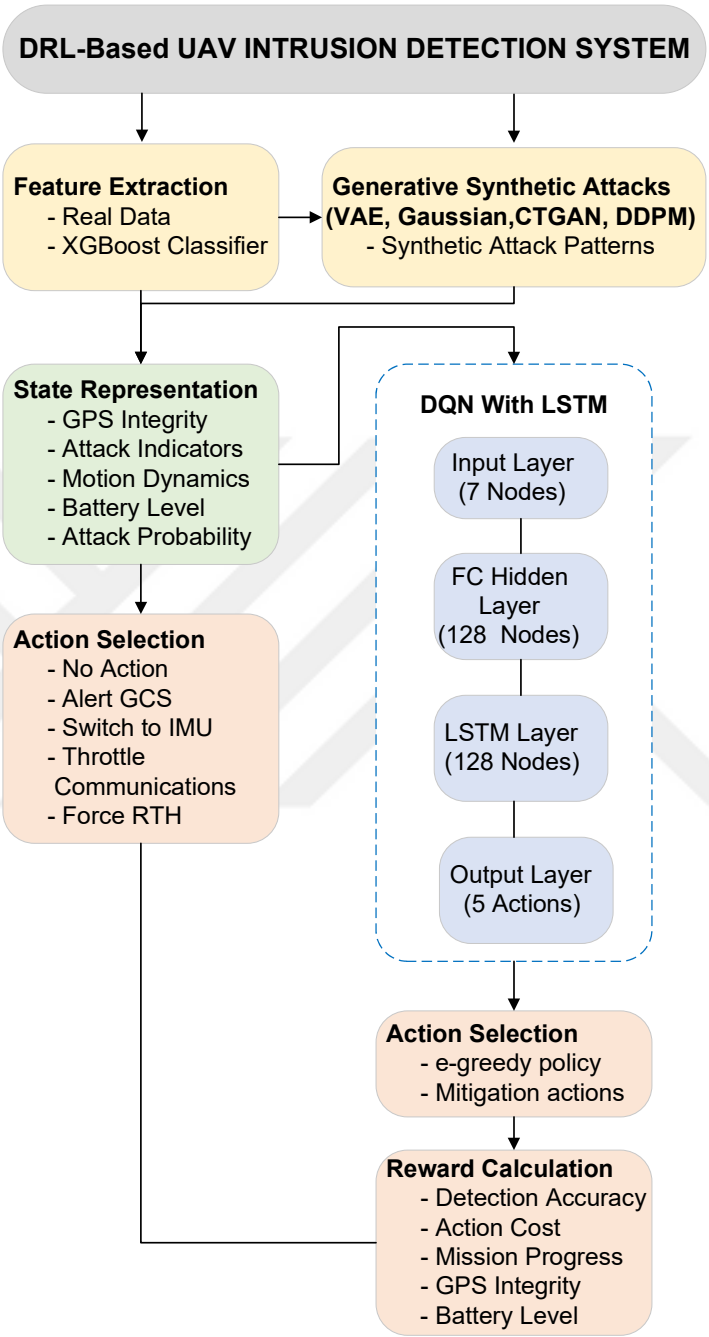

Figure 5.3 : Overview of the DRL-Based UAV Intrusion Detection System integrating generative synthetic attacks.

6. EXPERIMENTS

This chapter presents the experimental setup, implementation details, and evaluation methodologies used to assess the performance of the proposed hybrid intrusion detection framework for UAV security. The experiments are designed to validate the effectiveness of synthetic attack generation using Generative AI (GenAI) models and the adaptive threat mitigation capabilities of a Deep Q-Network (DQN)-based IDS under realistic UAV conditions.

Section 6.1 provides an overview of the datasets used, highlighting key features relevant to spoofing and jamming attacks. Section 6.2 details the configurations and architectural parameters of the generative models employed for synthetic data generation, including VAE, Gaussian Copula, CTGAN, and DDPM. Section 6.3 outlines the setup and tuning of the XGBoost-based IDS component. Section 6.4 describes the design, state representation, action space, and training procedure of the DQN agent for real-time intrusion detection. Finally, Section 6.5 defines the evaluation criteria and metrics used to measure both classification accuracy and reinforcement learning performance, including detection rates, resource efficiency, and policy stability.

6.1 Datasets and Features

UAV attack dataset in [178] provides a comprehensive repository of real communication data between UAVs and ground control stations, providing a valuable basis for simulating various attack scenarios, including GPS spoofing and jamming attacks. Moreover, the dataset includes both simulated and real flight records, and data from benign flights and flights under attack.

The dataset typically includes telemetry data, commands, and status messages exchanged between a UAV and a ground control station. Data covers both benign and malicious examples derived from real flight records, reaching a total of 6,446 rows and 84 columns for jamming attacks and 3,623 rows and 84 columns for GPS spoofing attacks. Table 6.1 summarizes the number of benign and malicious examples for each dataset, grouped by spoofing and jamming attack types. This table also presents a comparison of benign and malicious examples between real and generated synthetic datasets.

Table 6.1 : Comparison of benign vs. malicious samples across real and synthetic datasets.

Attack Type	**Dataset**	**Benign**	**Malicious**
Spoofing	Real Data	3124	498
	VAE Generated Data	3456	2804
	Gaussian Generated Data	2984	3236
	CTGAN Generated Data	3340	2860
	DDPM Generated Data	1761	1861
Jamming	Real Data	2561	596
	VAE Generated Data	2815	2485
	Gaussian Generated Data	2700	2600
	CTGAN Generated Data	2990	2310
	DDPM Generated Data	2890	2410

Figure 6.1 presents the distribution of benign and malicious samples in the jamming attack dataset, where 22.7% of the data is malicious and 77.3% is benign. Similarly, Figure 6.2 shows the distribution for the spoofing attack dataset, in which 13.7% of the data is malicious and 86.3% is benign. Both figures include comparisons of synthetic and real data, highlighting the malicious-to-benign sample ratios for each attack scenario.

We used XGBOOST classifiers to extract important features from the real dataset. The provided feature importance data explains how certain features play important roles in identifying jamming and spoofing attacks in GPS systems.

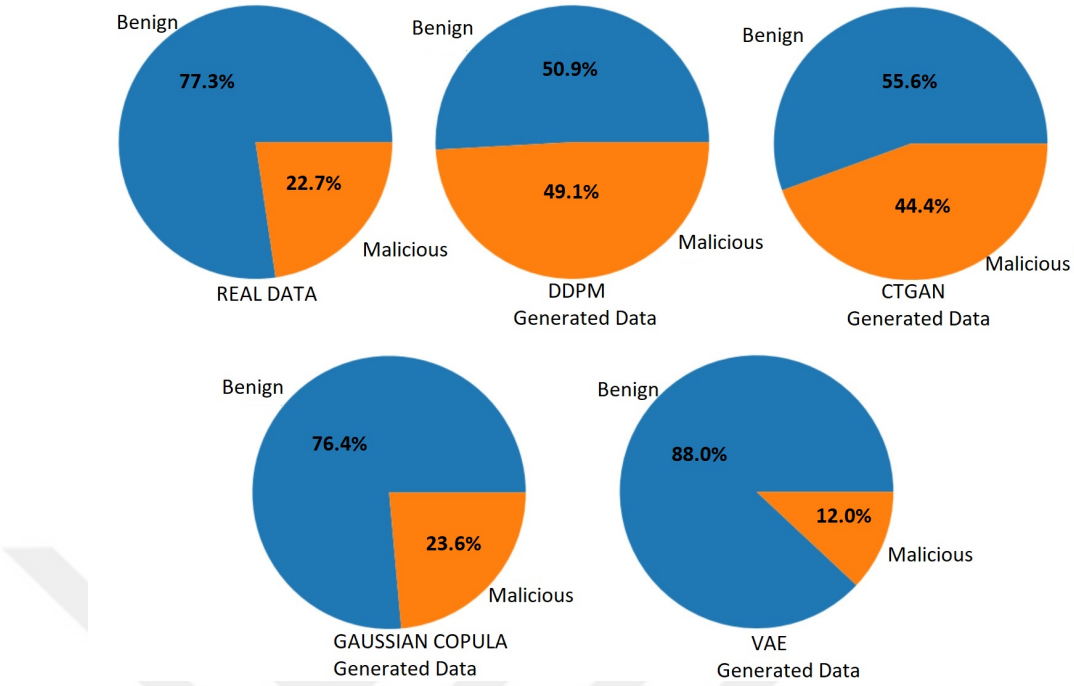

Figure 6.1 : Distribution of benign and malicious samples in jamming attack datasets.

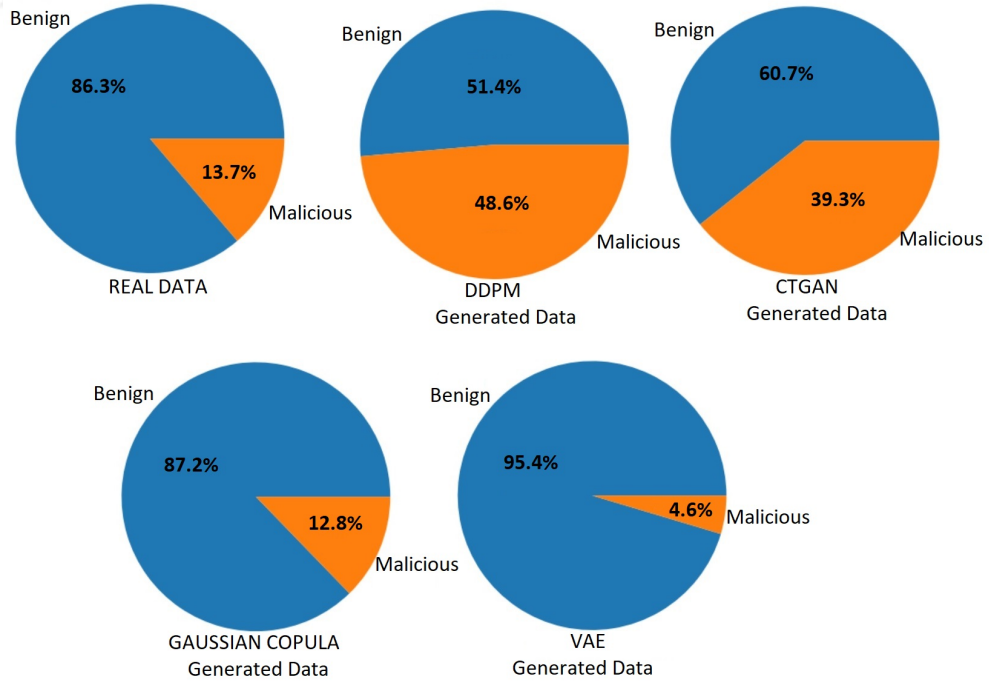

Figure 6.2 : Distribution of benign and malicious samples in spoofing attack datasets.

These two types of attacks have different impacts on GPS signal quality and navigation data, and the extracted features provide insights into the detection of these attacks and the generation of attacks with similar features to generative models. For example, jamming attack focuses on degrading GPS signal quality, which leads to a noise, a reduced satellite utilization, and a poor positional accuracy. Spoofing attack actively manipulates GPS data to mislead users, which results in impacts on location, speed, and orientation.

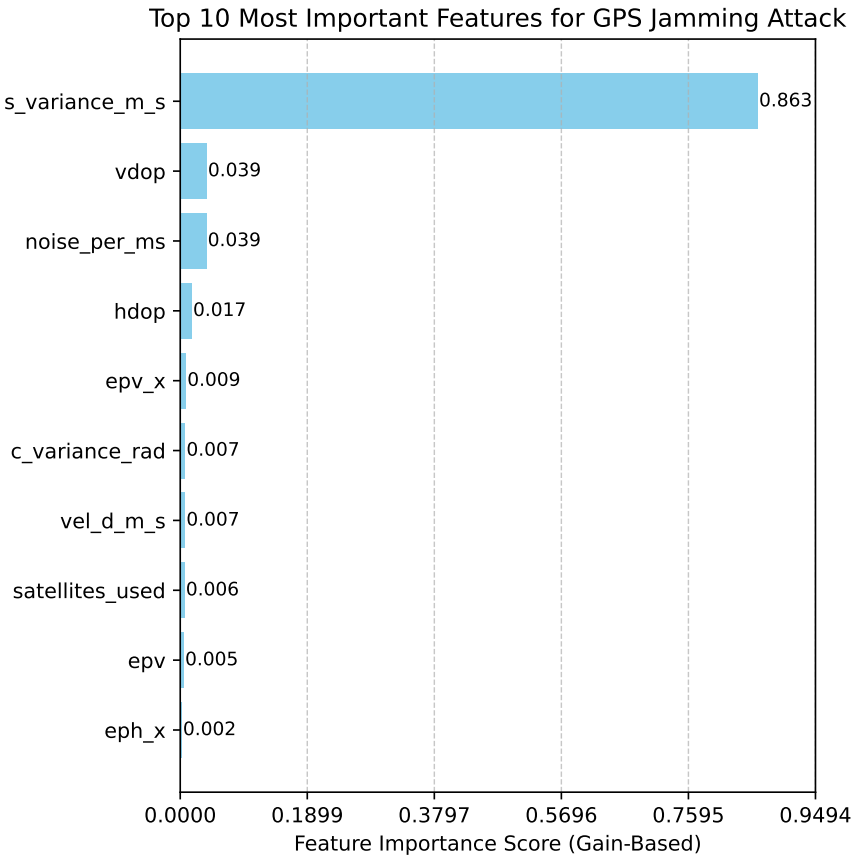

Figure 6.3 : Important features for jamming attacks.

Important features of GPS jamming attacks identified by the XGBOOST classifier are given in Figure 6.3. In the figure, *eph_x* represents the horizontal position accuracy of the GPS signal. *epv* indicates the vertical position accuracy. These features achieve the highest feature importance scores, highlighting their strong predictive power in identifying jamming. They measure the deviations in GPS position accuracy that jamming significantly increases.

Feature *satellites_used* denotes the number of satellites utilized in the GPS positioning system. The feature importance score of this feature indicates its importance in detecting a loss of usable satellites, which is sign of jamming. A reduction in the number of active satellites has a significant impact on the reliability of the GPS system. *vel_d_m_s* feature measures vertical velocity in meters per second. The high feature importance value of this feature suggests that jamming disrupts vertical velocity calculations, creating noisy data.

Feature *c_variance_rad* represents the variance of the path over the ground. The predictive power of this feature reflects the effect of obstructions on the stability of the computed trajectory. *hdop* and *vdop* features indicate the quality of the horizontal and vertical position solution. With significant feature importance values, these features show how jamming undermines the accuracy of GPS positioning by affecting satellite geometry. *noise_per_ms* feature measures the noise in the GPS signal. The high feature importance value of this feature indicates its effectiveness in capturing the increased signal noise associated with jamming. *s_variance_m_s* feature is the variance of the calculated speed. A high feature importance score for this feature indicates that it is useful for identifying irregular speed fluctuations caused by jamming.

In summary, if we consider effects of jamming features, jamming attacks degrade GPS signal quality by creating noise and reducing the number of effective satellites. Features such as *hdop*, *vdop*, and *noise_per_ms* effectively capture this degradation. Positional and velocity inaccuracies (*epv_x*, *epv*, *vel_d_m_s*) provide additional evidence of jamming attack.

Important features of GPS spoofing attacks identified by the XGBOOST classifier are given in Figure 6.4. In the figure, *eph* represents the horizontal position error similar to *eph_x* in jamming. This feature has a high feature importance score indicating its effectiveness in detecting positional inaccuracies.

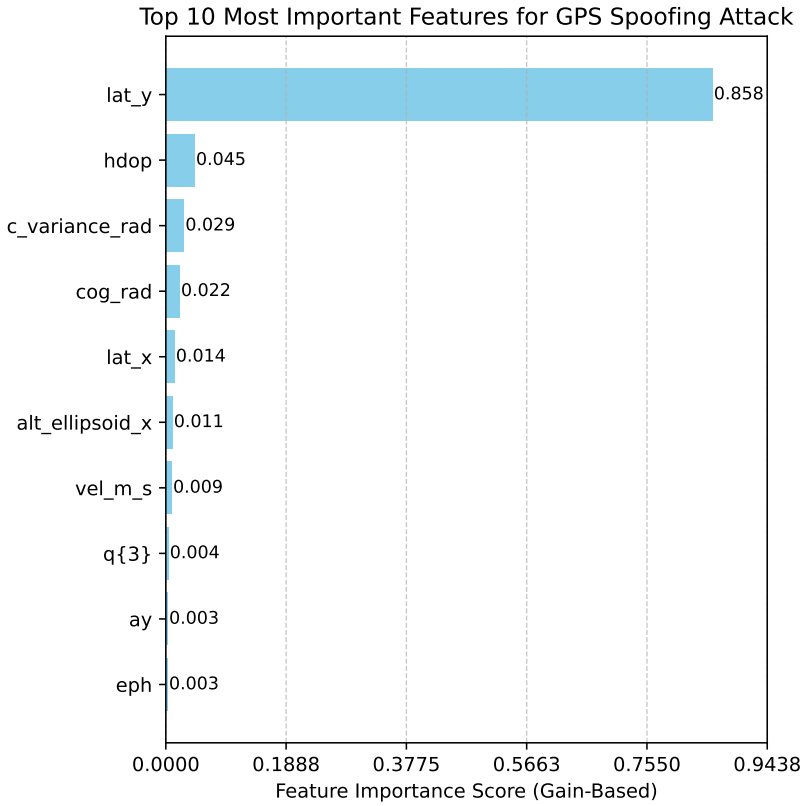

Figure 6.4 : Important features for spoofing attacks.

The feature *ay* represents the acceleration along the y-axis. $q\{3\}$ represents the 3D orientation in space. These features achieve high feature importance scores, demonstrating their ability to detect the inconsistencies in motion and orientation caused by spoofed signals. *vel_m_s* feature is the overall velocity in meters per second. The feature importance score here indicates its sensitivity to altered velocity data typical of spoofing attacks.

alt_ellipsoid_x feature represents the GPS altitude relative to a reference ellipsoid. This feature's feature importance score reflects its usefulness in detecting spoofed altitude values that differ from actual measurements. *lat_x* and *lat_y* are latitude coordinates. With remarkable feature importance scores, these features demonstrate their effectiveness in capturing false positional information injected by spoofing. *cog_rad* represents the direction of travel in radians. A high feature importance score for this feature emphasizes its role in detecting spoofing changes to the directional data. *c_variance_rad* feature measures course consistency. This feature captures inconsistency in course calculations caused by spoofing based on the feature importance score. *hdop*, like jamming, measures horizontal accuracy. Unlike jamming, spoofing can artificially reduce this value to create a false sense of precision that the feature successfully identifies.

In summary, when considering the impact of spoofing features on spoofing detection, features related to a position, a motion, and an orientation are the most important. Their high feature importance values confirm their ability to detect the manipulated data injected during spoofing attacks.

Analysis of feature importance scores reveals distinct patterns in the importance of features for detecting jamming and spoofing attacks. Jamming features emphasize the signal degradation and the noise, while spoofing features emphasize the positional and the navigational inconsistencies. This distinction allows us to create targeted detection strategies for these types of GPS attacks. Thus, extracted features are proven to be defining features of the attacks according to the feature importance score. These defining features helped us both in attack detection and in similarity tests of the generated synthetic data, similar to original data.

6.2 Generative AI Models Configuration

In this section, we discuss the architectural settings, hyperparameters, and baselines used for each generative model.

- VAE, we employ TVAE Synthesizer framework that uses a variational autoencoder [197]. It has different hyperparameters related to neural network like *batch_size*, *compress_dims*, and *decompress_dims*. These settings are specific to the neural network and are intended for optimizing the technical architecture and modeling process. *batch_size* parameter defines the number of data samples processed in each step with a default value of 10. Each model is trained for up to 300 *epochs*. *compress_dims* parameter specifies the size of each hidden layer in the encoder, with a default value of (128, 128). *decompress_dims* determines the size of each hidden layer in the decoder with default values (128, 128). *embedding_dim* parameter sets the embedding dimension size used by both the encoder and decoder, with a default value of 128. Regularization is controlled by *l2scale* parameter, which defaults to (1×10^{-5}). Lastly, $loss\_factor$ parameter acts as a multiplier for the reconstruction error, with a default value of 2.

- Gaussian Copula, we used Gaussian Copula Synthesizer, synthetic data vault (SDV), framework for the implementation [198]. First, the synthesizer learns the marginal distribution for each individual column, such as a beta distribution with parameters $(\alpha = 2)$ and $(\beta = 5)$. It uses this learned distribution to normalize the column values, transforming them into standard normal distributions with a mean of $(\mu = 0)$ and a standard deviation of $(\sigma = 1)$. Next, the synthesizer learns the covariance between each pair of normalized columns. These covariances are stored in an $(n \times n)$ matrix, where (n) represents the number of columns in the dataset.

- CTGAN, we employ fully-connected networks in both the generator and discriminator to capture all possible correlations among the columns. The generator and discriminator each utilize *two fullyconnected hidden layers*. In the generator, batch normalization and *ReLU activation functions* are applied. Following the two hidden layers, the generator produces synthetic row representations using a combination of activation functions. Scalar values are generated with a *tanh activation*, while mode indicators and discrete values are generated using a *Gumbel softmax function*. In the discriminator, each hidden layer employs leaky *ReLU activation functions* and incorporates *dropout* for regularization [199].

- DDPM, models are configured with different numbers of neurons [256, 512, 1024, 2048, 4096, 8192] and hidden layers [4, 6, 8, 10, 12]. Each model is trained for up to 3000 epochs with a minibatch size of 512. Adam optimizer is employed for training, with parameters ($\beta_1 = 0.9$) and ($\beta_2 = 0.999$), alongside a cosine learning rate scheduler. The model weights are initialized randomly following the method described in [200].

6.3 IDS Configuration

The XGBoost classifier (version 2.1.0) was hyperparameter-tuned via a grid search over the key tree parameters *learning_rate*, *max_depth*, and *min_child_weight*, as shown in Tables 6.2 and 6.3. For each attack type (jamming vs. spoofing) and each training-set variant, we selected the combination that maximized 10-fold cross-validation accuracy. All other parameters were left at their XGBoost defaults *booster type* is set to $'gbtree'$, which means the model uses tree-based methods. Since our classifier performs binary classification as malicious/benign, the objective function is $'binary : logistic'$. *verbosity* level is set to 1, thereby ensuring that any relevant warning messages are displayed and that *use_label_encoder* parameter is *enabled*, which applies label encoding to the target variable.

This grid-search procedure ensures that each IDS instance is tailored to the characteristics of its training data, improving both detection performance on in-distribution samples and robustness against synthetic attack variants.

Table 6.2 : Optimal hyperparameters for GPS jamming IDS.

Training Set	learning_rate	max_depth	min_child_weight
Real	0.10	3	1
Real + VAE	0.10	7	3
Real + Gaussian Copula	0.10	7	1
Real + DDPM	0.10	7	3
Real + CTGAN	0.10	7	1

Table 6.3 : Optimal hyperparameters for GPS spoofing IDS.

Training Set	learning_rate	max_depth	min_child_weight
Real	0.01	3	1
Real + VAE	0.10	7	1
Real + Gaussian Copula	0.10	7	5
Real + DDPM	0.10	7	3
Real + CTGAN	0.10	7	1

For the cross-testing experiments, each dataset variant was randomly split into 70% for training and 30% for testing. Stratified sampling was used to preserve the malicious/benign class ratio in both subsets. The XGBoost classifier was trained on the 70% split using the optimal hyperparameters from the grid search, and all performance metrics were evaluated on the held-out 30%.

6.4 DQN Configuration

To model UAV dynamics, threat interaction and decision-making dynamics, we implemented a custom OpenAI Gym environment. At each step, the environment processes either real or synthetic data and feeds the relevant telemetry features into pretrained XGBoost classifiers to estimate the likelihood of spoofing or jamming. These outputs, along with the internal UAV state, compose the observation provided to the DQN agent. Each simulation episode simulates a continuous UAV mission timeline, allowing the RL agent to observe evolving environmental states and learn effective policies for timely threat detection and mitigation.

6.4.1 State representation

The UAV agent receives a multi-dimensional state vector at each timestep to enable robust policy learning. Our RL-based IDS system uses a comprehensive state representation that captures key indicators of environmental reliability, system integrity, and classifier confidence. The state s_t is composed of six high-level categories, each aggregating critical sensor and model-derived indicators.

Formally, the state space is structured as in Eq. 6.1:

$$\begin{aligned} s_t = [&\text{GPS Integrity}, \text{ Jamming Indicators}, \text{ Spoofing Indicators}, \\ &\text{Motion Dynamics}, \text{ Battery Level}, \text{ Attack Probability}] \end{aligned} \tag{6.1}$$

Each component is described as follows:

GPS Integrity Metrics

This feature captures the UAV's reliance on authentic satellite signals and reflects the likelihood of GPS disruption. High integrity scores suggest reliable positioning, whereas low values may indicate spoofing or jamming effects. The GPS integrity value, denoted as $g_t \in [0,1]$, represents a normalized indicator of the quality of satellite-based positioning data. A value of $g_t = 1$ corresponds to a perfect GPS signal with maximum integrity, while $g_t = 0$ indicates complete signal failure. Specifically, GPS integrity is computed using the weighted sum of key features such as horizontal dilution of precision $hdop$, estimated horizontal position error eph_x, estimated vertical position error epv, and the number of satellites currently in use. The formulation is expressed as in Eq. 6.2:

$$g_t = w_1 \cdot (\text{hdop}) + w_2 \cdot (\text{eph})_x + w_3 \cdot (\text{epv}) + w_4 \cdot (\text{satellites\_used}) \tag{6.2}$$

where (w_i) are empirically tuned weights. This composite measure ensures that spatial geometry, signal dispersion, and satellite availability are all considered in evaluating GPS reliability within the reinforcement learning framework.

Jamming Indicators

Jamming metrics describe anomalies in signal-to-noise ratio or disruptions in communication channels. These indicators are extracted from both real sensor logs and synthetic datasets generated by the attacker module.

Spoofing Indicators

Spoofing indicators identify mimicry of GPS signals. These indicators are derived from both real sensor logs and synthetic datasets generated by the attacker module. Spoofing is often subtle; thus, these features are probabilistic and essential for temporal consistency evaluation.

Motion Dynamics

Motion dynamics reflect the physical state of the UAV, independent of external signals. These values are read directly from onboard sensors and include **velocity** and **altitude**. Include UAV velocities, acceleration deltas, heading, and altitude to detect physical anomalies.

Battery Level

The variable is maintained internally as **battery_level** and updated at each time step. This approach fosters energy-efficient decision-making and penalizes prolonged high-cost behavior.

Attack Probability Attack probabilities are outputs from the XGBoost classifier, indicating the estimated likelihood of an ongoing attack.

Altogether, these six categories form a compact yet expressive 7-feature vector (depending on implementation) that accurately represents UAV situational awareness. These inputs are fed into the Deep Q-Network to estimate action values $Q(s,a)$ for policy improvement.

6.4.2 Action space

The agent chooses from a discrete action set in Eq. 6.3:

$$\mathscr{A} = \{0,1,2,3,4\} \tag{6.3}$$

The actions correspond to:

- 0 — No Action : Baseline operation when no threat detected
- 1 — Alert Ground Control Station (GCS): Notifies operators with minimal system impact
- 2 — Switch to IMU navigation: GPS-denied navigation (high energy cost)
- 3 — Throttle Communications: Reduces RF exposure to jamming
- 4 — Force Return-to-Home (RTH): Initiates emergency landing or route back to safety.

These choices represent plausible UAV responses to spoofing or jamming attacks in real-time operation.

6.4.3 Multi-objective reward function design

In this study, the reinforcement learning-based Intrusion Detection System (IDS) for UAVs relies on a multi-objective reward function that balances attack mitigation, mission success, energy preservation, and decision rationality.

Figure 6.5 illustrates the architecture of the multi-objective reward function used in the UAV reinforcement learning framework. The design integrates five key components that contribute to the overall reward signal. Starting with **Detection Accuracy**, this branch evaluates whether the UAV correctly responds to threats, feeding into the *Base Reward* component. Simultaneously, each UAV action incurs an **Action Cost**, which is split into both direct penalties (e.g., invoking IMU mode or RTH) and energy-related implications, such as battery consumption.

The **Mission Progress** component tracks advancement toward mission completion, rewarding the UAV for maintaining forward momentum. The **GPS Integrity** module monitors signal reliability and encourages the system to trust or distrust GPS readings accordingly. Finally, the **Battery Level** module promotes long-term energy sustainability through small incremental adjustments based on current power levels.

All five reward components—Base Reward, Action Cost, Mission Progress, GPS Integrity, and Battery Level—are aggregated into a final **Multi-Objective Reward** signal, which serves as feedback to guide the UAV's policy updates. This modular and interpretable structure ensures the agent balances responsiveness, energy efficiency, and mission success in a unified reward-driven manner.

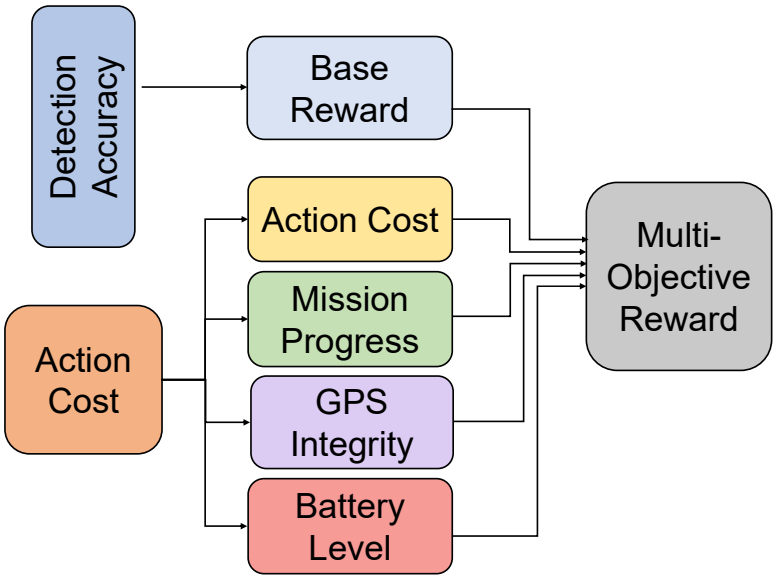

Figure 6.5 : A Figure of Multi-Objective Reward Function Design

Each component is described as follows:

Base Reward for Detection Accuracy

The base reward function is structured around four key detection outcomes, each contributing to the UAV's decision-making through distinct rewards or penalties. A **True Positive (TP)**, where the agent correctly identifies and mitigates an ongoing attack, is rewarded with $+10$, reinforcing the importance of accurate threat response. In contrast, a **False Negative (FN)**—failing to take action during an actual attack—results in a severe penalty of -50, underscoring the critical need to avoid missed detections.

For **False Positives (FP)**, where unnecessary mitigating actions are taken during benign conditions, the penalty is dynamic and scales with the frequency of false alarms.

$$R_{\text{FP}} = \begin{cases} -2 - 0.5 \times \texttt{fp\_count}, & \text{(standard)} \\ -5 - 0.5 \times \texttt{fp\_count}, & \text{(if GPS integrity is high)} \end{cases} \tag{6.4}$$

The formulation in Eq. 6.4 penalizes excessive reactions, particularly under reliable GPS conditions, and helps reduce alert fatigue.

Finally, a **True Negative (TN)**, where the agent appropriately refrains from acting when no threat is present, is assigned a modest reward of $+2$, encouraging operational efficiency without unnecessary interventions.

Overall, this reward design strategically balances urgency in threat response with stability and efficiency in normal operation, guiding the UAV toward optimal situational awareness and decision-making.

Action and Battery Cost

The action and battery cost function incorporates two key cost components for each action taken by the UAV. First, each action $a_t \in \mathscr{A} = \{\text{NoAction}, \text{Alert}, \text{IMU}, \text{Throttle}, \text{RTH}\}$ from the available action set carries a fixed penalty term $C_{\text{action}}(a_t) \in \{0, -1, -3, -2, -5\}$. These values reflect the relative resource intensity and operational impact of each mitigation strategy, ranging from no cost for passive behavior (NoAction) to the highest cost for the most resource-consuming maneuver (Return-to-Home, RTH).

In addition to this fixed action penalty, a dynamic battery cost component $C_{\text{battery}}(a_t)$ is incorporated to model energy consumption over time in Eq 6.5. It is defined as:

$$C_{\text{battery}}(a_t) = c_a \cdot \left(\frac{3000}{\text{max\_steps}} \right) \tag{6.5}$$

where c_a is a scaling coefficient and max_steps denotes the maximum allowed steps in the mission. This formulation ensures that the energy cost is normalized across different mission durations, maintaining consistent penalization across varying episode lengths.

Together, these dual penalty terms guide the UAV to balance effective threat mitigation with energy efficiency. The cost structure discourages excessive or unnecessary interventions while rewarding minimal yet effective action selection. By progressively increasing the penalties from passive actions to aggressive maneuvers, the framework ensures judicious resource usage and mission sustainability.

Mission Progress and GPS-Aware Rewards

To encourage safe navigation and continuous advancement toward mission objectives, the reward function incorporates both mission progress and GPS-aware reward terms. The mission progress reward R_{progress} increases gradually with each time step in Eq 6.6 and is defined as:

$$R_{\text{progress}} = 10 \cdot \frac{t}{T} \tag{6.6}$$

where t represents the current step, and T is the total number of allowed steps in the mission. This term promotes steady forward movement, rewarding the UAV as it nears task completion.

In addition to time-based progress, the reward function integrates GPS integrity to promote trust in reliable sensor data and penalize risky behavior when GPS is unreliable. The GPS-aware reward R_{gps} is defined in Eq 6.7.

$$R_{\text{gps}} = \begin{cases} +0.02, & \text{if GPS integrity } > 0.7 \text{ and } a_t \neq \text{IMU} \\ -0.03, & \text{if GPS integrity } < 0.3 \text{ and } a_t \neq \text{IMU} \\ 0, & \text{otherwise} \end{cases} \tag{6.7}$$

This formulation incentivizes the UAV to maintain reliance on high-integrity GPS signals when available and encourages switching to IMU mode when GPS signals are weak or untrustworthy. By combining these terms, the reward design fosters both proactive mission progress and intelligent, context-aware navigation decisions.

Battery Level Reward

The battery reward system is designed to actively promote energy conservation while accounting for natural power depletion. When the UAV's battery level exceeds 50% ($b_t > 0.5$), it receives a small positive reward of $+0.01$ to encourage maintaining a healthy energy reserve. Conversely, if the battery drops below 20% ($b_t < 0.2$), a penalty of -0.01 is applied to urge more conservative behavior. For battery levels in the intermediate range ($0.2 \leq b_t \leq 0.5$), no additional reward or penalty is assigned. All scenarios include a consistent fixed drain penalty of -0.01 per time step to reflect the inevitable energy consumption during UAV operation in Eq 6.8.

Formally, the battery-related reward is expressed in Eq 6.8. This dual-mechanism ensures that the UAV balances mission objectives with prudent energy management, preventing critical power depletion while acknowledging the operational realities of continuous flight. The modest reward magnitudes are carefully chosen to avoid over-prioritization of battery conservation at the expense of threat mitigation or mission progress.

$$R_{\text{battery}} = \begin{cases} +0.01, & \text{if } b_t > 0.5 \\ -0.01, & \text{if } b_t < 0.2 \quad \text{(plus fixed drain penalty)} \\ 0, & \text{otherwise} \end{cases} \tag{6.8}$$

Table 6.4 : Reward design summary used in the UAV environment.

Condition	Description	Reward / Penalty
True Positive (TP)	Correct action on attack	$+10$
False Negative (FN)	No action on attack	-50
True Negative (TN)	No action on benign input	$+2$
False Positive (FP)	Action on benign input	$-2-0.5\times \texttt{fp\_count}$
High GPS Integrity	FP penalty if GPS is good	$-5-0.5\times \texttt{fp\_count}$
Mission Completion	UAV reached max steps	$+300$
Battery Drain	Battery depleted	-50
Progress Bonus	Proportional to step progress	$+10\cdot \frac{\texttt{step}}{\texttt{max\_steps}}$
Battery Efficiency	Battery > 0.5 or < 0.2	$+0.01$, -0.01
IMU Decision Bonus	GPS-aware bias for IMU decision	$+0.02$, -0.03, 0
Action Cost	Cost per decision	$\in \{0,-1,-3,-2,-5\}$

Table 6.4 outlines the reward design principles integrated into the UAV environment to train the reinforcement learning-based intrusion detection system (IDS). The reward structure balances multiple objectives such as attack mitigation, energy efficiency, and mission success. Positive rewards are assigned for correct intrusion responses, such as taking appropriate actions during attacks (True Positive: $+10$) and avoiding unnecessary responses when the environment is benign (True Negative: $+2$). Severe penalties are imposed for critical failures, including missing an attack (False Negative: -50) or depleting the UAV's battery before mission completion (-50). To discourage false alarms, the agent is penalized when unnecessary actions are taken during safe conditions (False Positive), with a penalty that increases based on GPS quality and the number of consecutive false positives. Additional components such as mission progress bonuses, battery efficiency incentives, and GPS-aware decision shaping for IMU switching contribute to refining agent behavior. Each action taken incurs a predefined cost, encouraging minimal intervention and energy-aware operation. Collectively, this reward structure enables robust, cautious, and goal-directed policy learning.

Final Multi-Objective Reward Composition

The reward function R_t at each step t is designed to encourage correct mitigation of jamming/spoofing attacks while discouraging unnecessary or costly actions.

Combining all components, the total reward at step t is in Eq 6.9.

$$R_t = R_{\text{base}} + C_{\text{action}}(a_t) + R_{\text{progress}} + R_{\text{gps}} + R_{\text{battery}} \tag{6.9}$$

In the terminal step ($t = T$), if the UAV completes the mission, a bonus of $+300$ is applied. Additionally, if the UAV fails due to a low battery, a penalty of -50 overrides other rewards.This reward shaping ensures that the UAV agent learns to (i) correctly detect and mitigate attacks, (ii) minimize false positives, (iii) preserve battery life, and (iv) complete missions efficiently and securely.

6.4.3.1 Energy-aware state transition dynamics

The state transition function $\mathscr{T} : \mathscr{S} \times \mathscr{A} \to \mathscr{S}$ governs the evolution of the UAV's state $s_t \in \mathscr{S}$ upon execution of an action $a_t \in \mathscr{A}$. In our implementation, the transition dynamics integrate deterministic updates (from the action policy), stochastic environmental fluctuations, and adversarial effects (via synthetic attack sampling) in Figure 6.6. Our model integrates *physical dynamics*, *attack perturbations*, and *mitigation effects* through equation shown in Equation (6.10):

$$s_{t+1} = f_{\text{phys}}(s_t, a_t) + f_{\text{attack}}(s_t, \xi_t) + \varepsilon_t \tag{6.10}$$

In this formulation, s_t is the multi-dimensional state vector at time t, encoding features such as battery level, GPS integrity, jamming and spoofing probabilities, velocity, and altitude. The action a_t is a discrete control signal issued by the agent. The function f_{phys} models the deterministic evolution of UAV dynamics in response to control actions. The function f_{attack} introduces adversarial disruptions, where xi_t is a latent variable reflecting stochastic attack generation. Finally, $\varepsilon_t \sim \mathscr{N}(0, \Sigma)$ captures zero-mean Gaussian noise injected into the system to simulate sensor uncertainty, with Σ as the covariance matrix.

The physical update function f_{phys} includes changes to battery level b_t , velocity v_t, and altitude α_t. These dynamics are governed by Equations (6.19)–(6.13):

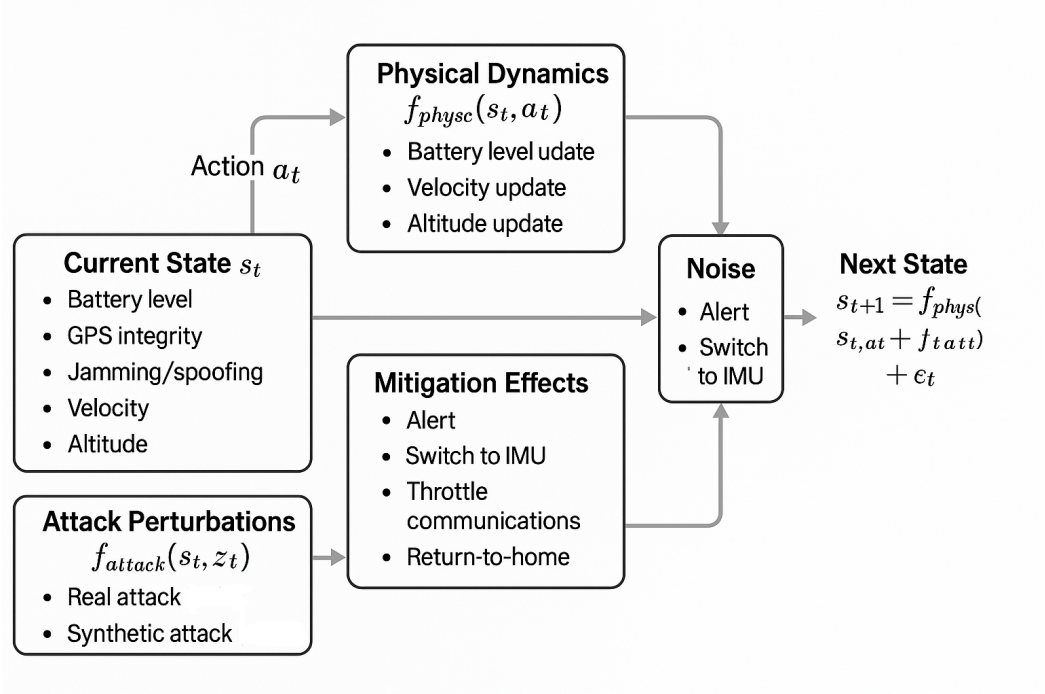

Figure 6.6 : Energy Aware State Transition Dynamics.

$$b_{t+1} = \max(b_t - \Delta_b(a_t),\ 0) \qquad \text{(Battery)} \tag{6.11}$$

$$v_{t+1} = \text{clip}(v_t + \mathcal{N}(0, \sigma_v^2),\ [0,\ v_{\max}]) \qquad \text{(Velocity)} \tag{6.12}$$

$$\alpha_{t+1} = \alpha_t + \mathcal{N}(0, \sigma_\alpha^2) \qquad \text{(Altitude)} \tag{6.13}$$

In these expressions, $\Delta_b(a_t)$ represents the energy cost associated with action a_t, σ_v^2 and σ_α^2 denote the variance of process noise for velocity and altitude, respectively, and the `clip` function enforces predefined operational bounds.

The attack perturbation function f_{attack} adds synthetic or real-world adversarial influences to the system state. As shown in Equation (6.14), real attacks occur with a fixed probability of 10%, modeled via a Bernoulli distribution, while synthetic attacks are sampled from a generative model G_ϕ , where ϕ denotes the model parameters and $z_t \sim \mathscr{Z}$ is a latent vector:

$$f_{\text{attack}}(s_t) = \begin{cases} \text{Bernoulli}(p = 0.1), & \text{(Real Attack)} \\ G_\phi(z_t),\ z_t \sim \mathscr{Z}, & \text{(Synthetic Attack)} \end{cases} \tag{6.14}$$

Synthetic attacks are generated using a collection of generative artificial intelligence models, including VAE, Gaussian Copula, CTGAN, and DDPM. These methods produce telemetry-aligned attack traces with structural resemblance to real threats, contributing to the realism of training scenarios.

Action-Specific Modifications

In addition to passive dynamics, the agent directly modifies elements of the state through discrete mitigation actions. The action-dependent transformation $\mathscr{T}(s_t, a_t)$ deterministically alters the state by injecting rule-based effects, as expressed in Equation (6.15):

$$\mathscr{T}(s_t, a_t) = s_t \oplus \begin{cases} [\text{GPS\_integrity} = 0.1], & a_t = 1 \text{ (Alert)} \\ [\text{GPS\_integrity} = 0.9,\ \Delta_b = 0.03], & a_t = 2 \text{ (Switch to IMU)} \\ [j_t = 0.9], & a_t = 3 \text{ (Throttle Communications)} \\ [\alpha_t = 0.1], & a_t = 4 \text{ (Return-to-Home)} \end{cases} \tag{6.15}$$

Here, $\oplus$ denotes a partial overwrite of specific state dimensions. For example, action 2 (Switch to IMU) increases GPS integrity confidence to 0.9 and incurs an additional battery penalty of 0.03. Similarly, action 3 (Throttle Communications) simulates heightened jamming sensitivity, while action 4 (Return-to-Home) reduces altitude abruptly to initiate emergency descent.

Physical Constraints

To maintain operational feasibility and prevent unrealistic system states, each updated state vector is bounded using element-wise clipping. This is formalized in Equation (6.16):

$$s_{t+1} = \text{clip}(\mathscr{T}(s_t, a_t),\ [s_{\min},\ s_{\max}]) \tag{6.16}$$

In this expression, $s_{\min}$ and $s_{\max}$ are predefined vectors encoding minimum and maximum permissible values for each state dimension (e.g., battery level $\in [0,1]$, velocity $\in [0, v_{\max}]$, GPS integrity $\in [0,1]$. The `clip` function ensures that no physical constraint is violated, thereby preserving the fidelity of the simulation environment and enforcing hardware-specific safety margins.

6.4.4 Experimental settings

The Deep Q-Network (DQN) agent used in this study was implemented to learn adaptive intrusion response policies within a custom simulation environment for UAV systems. The environment was structured as a Markov Decision Process (MDP), enabling the agent to iteratively improve its policy through interaction with both real and synthetic attack data.

Table 6.5 : DQN training hyperparameters.

Hyperparameter	**Value**
Optimizer	Adam
Learning Rate	0.001
Discount Factor (γ)	0.99
Batch Size	64
Replay Buffer Size	10,000
Target Network Update Frequency	Every 100 episodes
Exploration Strategy	ε-greedy
Initial ε	1.0
Minimum ε	0.01
ε Decay Rate	0.995
Training Episodes	500
Maximum Steps per Episode	400
Neural Network Layers	FC(128) + LSTM(128) + Output(5)
Activation Function	ReLU
Loss Function	Mean Squared Error (MSE)
State Representation Size	7 (Telemetry + IDS Outputs)
Action Space Size	5 (NoAction, Alert, IMU, Throttle, RTH)

DQN model employed for UAV intrusion detection is trained using a carefully selected set of hyperparameters to ensure stable learning, efficient convergence, and optimal policy development. DQN used in our system consists of a neural network designed to estimate optimal action-values for a given state under UAV attack scenarios. Its architecture and training hyperparameters are summarized in Table 6.5.

The training process utilizes the **Adam optimizer** with a **learning rate of 0.001**, enabling adaptive gradient updates. A **discount factor** of $\gamma = 0.99$ is applied to emphasize long-term rewards over immediate gains. Each training batch consists of **64 samples**, while a **replay buffer** of size 10,000 maintains past experiences to improve learning through uncorrelated sampling. The **target Q-network** is updated every 100 episodes to stabilize temporal difference targets.

An ε**-greedy** strategy is used to guide exploration during training, beginning with an **initial** ε **of 1.0** and decaying to a **minimum** ε **of 0.01** at a rate of **0.995**. The model is trained for **500 episodes**, with each episode limited to a maximum of **400 steps**. The neural network architecture includes a **fully connected (FC) layer with 128 nodes**, followed by a **Long Short-Term Memory (LSTM) layer with 128 nodes**, and a final **output layer** corresponding to five possible mitigation actions. The **ReLU activation function** is used in the hidden layers, and the **Mean Squared Error (MSE)** loss function is used for Q-value approximation.

The **state representation** comprises **7 input features**, including telemetry data and IDS classifier outputs. The **action space** includes five discrete actions: *NoAction*, *Alert*, *IMU*, *Throttle*, and *Return-to-Home (RTH)*. This setup supports a robust and adaptive policy learning framework suitable for the dynamic and resource-constrained environment of UAV-based security systems.

After the DQN models were trained using real attack data, their robustness was evaluated across five different test datasets: Real, VAE, Gaussian Copula, DDPM, and CTGAN. These test datasets were designed to represent diverse and complex adversarial patterns, enabling an assessment of the agent's generalization capabilities. Importantly, the IDS variant used during training for each DQN model corresponds to the naming convention of the test sets (e.g., Real+VAE indicates a DQN trained with IDS output from a VAE-augmented dataset, tested on VAE-based synthetic attacks).

To ensure a realistic simulation of real-world UAV intrusion scenarios, the dataset was partitioned as follows: 70% of the real and synthetic attack data were used for IDS training, 10% were allocated for DQN training, and the remaining 20% were reserved for DQN testing. Crucially, DQN training was performed solely on real attack data to preserve the authenticity of training interactions and prevent bias toward synthetic patterns. This design choice ensures that the reinforcement learning agent learns defense policies in a setting that closely resembles real UAV missions, while its adaptability and resilience are evaluated under challenging and diverse synthetic attack conditions. This comprehensive evaluation highlights the effectiveness of the hybrid GenAI + DRL framework under both familiar and unseen adversarial scenarios.

6.5 Evaluation Criteria

To comprehensively evaluate the performance of our DQN-based UAV intrusion detection system, we adopt a diverse set of metrics organized into four categories: detection accuracy, resource efficiency, reinforcement learning stability, and comparative benchmarking. Each metric is formally defined below.

6.5.1 Detection performance

The primary objective of the IDS is to detect malicious activity accurately and promptly. We assess classification quality using standard binary detection metrics:

Attack Detection Rate (ADR): Also known as true positive rate(TPR), this metric is defined as the ratio of true positives (TP) to the sum of true positives and false negatives (FN). It quantifies the system's ability to correctly detect attacks:

$$\text{ADR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6.17}$$

False Alarm Rate (FAR): Also known as false positive rate(FPR), this metric represents the likelihood of false positives, i.e., benign samples misclassified as attacks:

$$\text{FAR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{6.18}$$

where FP denotes false positives and TN denotes true negatives.

6.5.2 Resource efficiency

As UAVs are constrained by limited power and computation, we also evaluate system efficiency:

Average Battery Consumption: This metric estimates the average energy usage across an episode, computed as:

$$\text{Battery}_{\text{avg}} = \frac{1}{T}\sum_{t=1}^{T}\Delta_b(a_t) \tag{6.19}$$

where $\Delta_b(a_t)$ is the energy cost associated with action a_t and T is the total number of time steps.

Cumulative Action Cost: The total cost incurred from the sequence of agent actions is given by:

$$\text{TotalCost} = \sum_{t=1}^{T}\text{cost}(a_t) \tag{6.20}$$

where $\text{cost}(a_t)$ is defined by the action-to-cost mapping.

6.5.3 RL-specific metrics

To understand the training behavior of the DQN agent, we report the following reinforcement learning diagnostics:

Reward Convergence: The agent's policy is considered stable if the cumulative episode reward R_{episode} converges over training iterations:

$$R^{(k)}_{\text{episode}} = \sum_{t=1}^{T} r_t^{(k)} \tag{6.21}$$

where $r_t^{(k)}$ is the reward at time t during episode k.

Return to Home Activity:

The Return to Home Activity is a metric that evaluates the system's efficacy in initiating RTH actions in response to detected attacks. It quantifies the proportion of attack episodes in which the UAV safely executed an RTH action to prevent potential threat actors from compromising the system. The definition in Equation 6.22 has been provided for the metric.

$$\text{RTH\_Activity} = \frac{\text{RTH Under Attacks}}{\text{RTH Invocations}} \tag{6.22}$$

RTH Under Attacks refers to the number of times the RTH mechanism was invoked in response to confirmed attack scenarios (i.e., true positives). RTH Invocations denotes the total number of RTH actions triggered, regardless of whether an actual attack was present. A higher RTH Activity indicates that the agent not only detects threats but can also return to the safety starting point with a high degree, thus increasing mission resilience.

Mission Completion Rate:

The mission completion metric evaluates whether an agent successfully fulfills two critical conditions simultaneously: (1) reaching the maximum allowed number of steps in an episode $step\_count >= env.max\_steps$, and (2) maintaining a positive battery level $state[0] > 0$. If both conditions are satisfied, the mission is deemed completed $mission\_completed = True$, and a value of 1 is appended to the ($mission\_completions$) list; otherwise, a 0 is recorded. This binary metric enables tracking of the agent's capability to endure the full mission duration without depleting its energy resources. Over multiple episodes, the success rate is computed by averaging the values in ($mission\_completions$), offering a quantitative measure of the agent's reliability and energy efficiency. This is particularly relevant in energy-constrained environments for UAV operations, where sustained performance and power-aware behavior are essential for mission success.

6.5.4 Comparative evaluation

To validate generalization and robustness, we compare our model across multiple test conditions:

Synthetic vs. Real Attack Performance: We compare detection scores (ADR and FAR) separately for real and synthetic attack datasets, generated using VAE, Gaussian Copula, CTGAN, and DDPM.

Generalization Across Attack Variants: Performance is evaluated against unseen spoofing and jamming patterns to test cross-attack generalization.

Baseline Comparison: Our DQN-based adaptive IDS is compared against a static supervised IDS (XGBoost classifier) to demonstrate improvements in adaptability and detection timeliness under dynamic threat conditions.

These metrics comprehensively capture the IDS's effectiveness, stability, and resource awareness in handling both real and synthetic adversarial conditions.

7. RESULTS

This chapter presents the outcomes of our proposed affective framework. The subsequent sections provide a systematic and detailed account of the results obtained from each module implemented within the framework. In 7.1 section, statistical results of synthetic attacks generated with GenAI models are analyzed. IDS performance results are given for both spoofing and jamming attacks with the generated synthetic attack data.

7.1 GenAI Statistical Results

7.1.1 GPS jamming attack generation results

Table 7.1 contains a detailed comparison of synthetic data generation methods CTGAN, VAE-GAN, Gaussian Copula, and DDPM for features in jamming scenarios. Metrics used for the evaluation include KL Divergence, Wasserstein Distance, and KS Statistic.

VAE struggles to match the real data distribution, showing significantly higher KL Divergence for variables like *vel_d_m_s* and *c_variance_rad*, suggesting its synthetic data is less aligned with the real data. VAE also shows higher Wasserstein Distances across most variables, with notable gaps for *vel_d_m_s* and *c_variance_rad*, confirm its weaker alignment with the real data. VAE consistently performs poorly, with higher KS Statistic scores across all variables, showing significant distributional differences between its synthetic data and the real data.

Table 7.1 : Similarity of important features between real and synthetic data for GPS jamming attack.

Metric	Important Features	CTGAN	VAE-GAN	Gaussian Copula	DDPM
KL Divergence	*eph_x*	**1.234**	2.345	1.567	1.789
	epv	0.987	1.234	0.876	**0.765**
	satellites_used	**0.567**	2.345	1.234	0.987
	vel_d_m_s	2.345	3.456	2.123	**1.890**
	c_variance_rad	1.456	2.678	1.345	**0.987**
	epv_x	0.890	1.234	0.765	**0.654**
	hdop	0.543	0.678	0.432	**0.321**
	noise_per_ms	**1.234**	2.345	1.678	**1.234**
	vdop	0.987	1.456	0.876	**0.654**
	s_variance_m_s	0.678	1.234	0.543	**0.432**
Wasserstein Distance	*eph_x*	0.234	0.345	0.123	**0.098**
	epv	0.123	0.234	0.098	**0.076**
	satellites_used	0.456	0.678	0.432	**0.321**
	vel_d_m_s	0.567	0.789	0.543	**0.432**
	c_variance_rad	0.234	0.456	0.321	**0.210**
	epv_x	0.098	0.123	0.076	**0.054**
	hdop	0.321	0.432	0.210	**0.098**
	noise_per_ms	0.654	0.789	**0.543**	0.432
	vdop	0.210	0.321	0.098	**0.054**
	s_variance_m_s	0.098	0.123	0.076	**0.054**
KS Statistic	*eph_x*	0.543	0.654	0.432	**0.321**
	epv	0.321	0.543	0.234	**0.123**
	satellites_used	0.789	0.890	0.678	**0.543**
	vel_d_m_s	0.432	0.543	0.321	**0.234**
	c_variance_rad	0.543	0.678	0.432	**0.321**
	epv_x	0.321	0.432	0.234	**0.123**
	hdop	0.654	0.789	0.543	**0.432**
	noise_per_ms	0.543	0.654	0.432	**0.321**
	vdop	0.432	0.543	0.321	**0.234**
	s_variance_m_s	0.321	0.432	0.234	**0.123**

Gaussian Copula performs well for variable *eph_x* achieving competitive KL Divergence values close to CTGAN. It provides competitive Wasserstein Distance values, particularly excelling for variables *noise_er_ms*. Gaussian Copula achieves reasonable KS Statistic values for important features. The results of Gaussian Copula suggest its ability to accurately replicate certain variable distributions. Furthermore, Gaussian Copula is the model that produces the closest data to reality after DDPM.

CTGAN consistently achieves the lowest KL Divergence for most variables, such as *eph_x*, *satellites_used*, and *noise_per_ms* indicating its strength in closely approximating the distribution of real data.

DDPM demonstrates strong performance with the lowest KL Divergence, Wasserstein Distance and KS Statistic for all important features. This highlights DDPM's capability to minimize the distributional difference between real and synthetic data.

The evaluation of synthetic data generation methods for jamming scenarios highlights DDPM as the most reliable and consistent performer across all metrics, particularly excelling in KL Divergence and Wasserstein Distance. Its ability to closely replicate real data distributions for critical variables. Gaussian Copula provides competitive results for selected variables, such as *c_variance_rad* and *vel_d_m_s*, which indicates its suitability for these distributions. However, VAE consistently under-performs, with higher KL Divergence and Wasserstein Distance values across most variables, underscoring its limitations in accurately capturing real data characteristics. These findings emphasize the importance of selecting the appropriate synthetic data generation method based on the specific application, with DDPM emerging as the most robust choice, while CTGAN and Gaussian Copula offer valuable alternatives for variable-specific requirements.

CTGAN emerges as the most reliable model, consistently producing synthetic data with mean and standard deviation values closest to the real data across most variables. For instance, variables like *eph_x*, *satellites_used*, and *hdop* show minimal deviations, reflecting CTGAN's robust ability to replicate the central tendency and variability of real data distributions. This makes CTGAN particularly suitable for tasks that demand precise statistical replication.

DDPM performs competitively, especially for variables such as *vel_d_m_s* and *s_variance_m_s*, where it closely approximates the real data in terms of both mean and standard deviation. However, for more complex variables like *c_variance_rad* and *noise_per_ms*, DDPM demonstrates slightly larger deviations, suggesting limitations in modeling distributions with higher variability.

Gaussian Copula provides moderate accuracy and performs well for variables such as *epv* and *vdop*, achieving mean and standard deviation values relatively close to real data. However, its performance is inconsistent, as evident in variables like *epv_x* and *satellites_used*, where it shows significant deviations, indicating challenges in handling complex or multimodal distributions.

In VAE, generating plausible data for some variables, such as *noise_per_ms*, exhibits significant deviations in both mean and standard deviation for variables like *c_variance_rad* and *hdop*. This underperformance highlights limitations in its architecture, making it less effective for accurately replicating the statistical properties of real data in jamming scenarios. Table 7.2 summarizes the mean and standard deviation of the specified variables for real data and synthetic data generated by CTGAN, DDPM, Gaussian Copula, and VAE, which have been calculated.

In general, the Gaussian Copula demonstrates the most consistent and reliable performance, excelling at reproducing real data distributions across most variables. DDPM shows promise for certain features but lacks the overall consistency of Gaussian Copula. CTGAN delivers competitive results, especially for features like *eph_x* and *satellites_used*, but slightly underperforms on others. VAE exhibits strong performance for some variables but requires further optimization to achieve the consistency observed in the Gaussian Copula.

The distributions of three selected important features are shown as density plots on the graphs for each model, VAE, Gaussian Copula, CTGAN, DDPM, in the jamming attack. The density plot demonstrates the distribution of the *s_variance_m_s* feature across the datasets comprising the real data, CTGAN, Gaussian Copula, DDPM, and VAE-GAN models in the context of a GPS jamming attack in Figure 7.1. The real data, depicted by the blue curve, represents the ground truth and exhibits a uni-modal distribution with a peak occurring at approximately 0.8. Among the generative models, the Gaussian Copula, green curve, most closely resembles the real data, exhibiting a peak at nearly the same location and a similar spread.

VAE, purple curve, shows a broader distribution, deviating from the compactness of the real data while slightly overestimating the density at higher values. CTGAN, orange curve, demonstrates a reasonable alignment with the real data, though it shifts the peak slightly to the right and broadens the spread. DDPM, red curve, captures the general shape but underestimates the density at the peak while overestimating it in the tails. This analysis suggests that the Gaussian Copula is the most effective model for replicating this feature, while VAE and DDPM require further refinement to improve their fidelity. CTGAN offers a moderate balance between fidelity and diversity.

Table 7.2 : Synthetic data comparison for GPS jamming attack.

Feature	**Real Data**	**CTGAN**	**DDPM**	**Gaussian Copula**	**VAE-GAN**
eph_x	1.23 ± 0.34	1.25 ± 0.36	1.21 ± 0.33	1.22 ± 0.35	1.27 ± 0.37
epv	0.98 ± 0.12	0.97 ± 0.11	0.96 ± 0.10	0.99 ± 0.13	0.95 ± 0.14
satellites_used	5.12 ± 1.34	5.15 ± 1.35	5.10 ± 1.33	5.08 ± 1.32	5.13 ± 1.36
vel_d_m_s	2.45 ± 0.54	2.50 ± 0.55	2.42 ± 0.52	2.46 ± 0.53	2.48 ± 0.56
c_variance_rad	0.23 ± 0.06	0.24 ± 0.07	0.22 ± 0.05	0.25 ± 0.08	0.21 ± 0.07
epv_x	0.34 ± 0.09	0.33 ± 0.08	0.35 ± 0.10	0.32 ± 0.09	0.36 ± 0.10
hdop	1.12 ± 0.31	1.13 ± 0.32	1.10 ± 0.30	1.15 ± 0.33	1.14 ± 0.34
noise_per_ms	0.45 ± 0.12	0.46 ± 0.13	0.44 ± 0.11	0.47 ± 0.14	0.43 ± 0.12
vdop	0.67 ± 0.18	0.66 ± 0.17	0.68 ± 0.19	0.69 ± 0.20	0.65 ± 0.18
s_variance_m_s	0.78 ± 0.22	0.80 ± 0.23	0.76 ± 0.21	0.79 ± 0.24	0.77 ± 0.22

The density plot demonstrates the distribution of *epv_x* feature across the datasets in GPS jamming attacks in Figure 7.2. The real data, blue curve, exhibits a sharp, concentrated peak at approximately 4.375. Among the generative models, DDPM, red curve, aligns most closely with the real data, accurately replicating the peak density and shape.

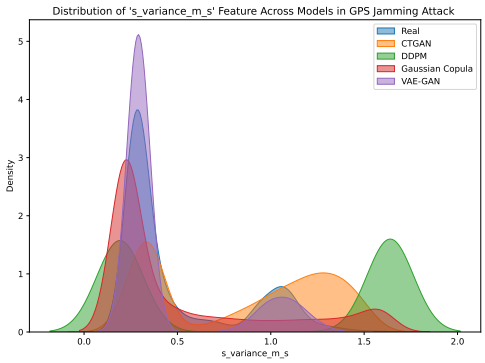

Figure 7.1 : Distributions of 's_variance_m_s' feature across models in GPS jamming attacks.

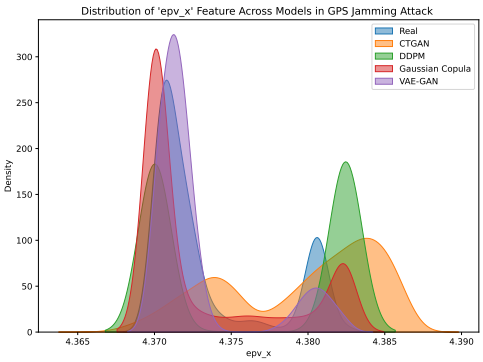

Figure 7.2 : Distributions of 'epv_x' feature across models in GPS jamming attacks.

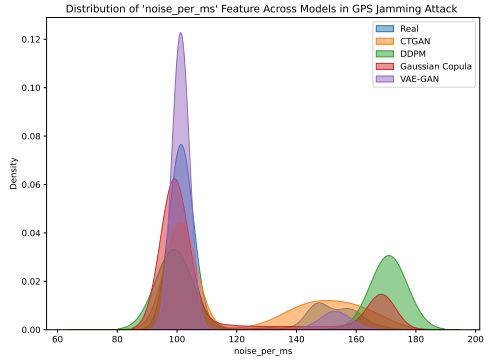

Figure 7.3 : Distributions of 'noise_per_ms' feature across models in GPS jamming attacks.

Gaussian Copula, green curve, also performs well but slightly broadens the distribution, reducing its fidelity. CTGAN, orange curve, demonstrates a significant shift in the peak and a broader spread, resulting in a less accurate replication. VAE, purple curve, further deviates, exhibiting inconsistencies in both the peak density and the general shape of the distribution. These results indicate that DDPM is the most effective model for replicating this feature, with Gaussian Copula as a close second. Both CTGAN and VAE demonstrate notable deviations, highlighting areas for further optimization.

The density plot for *noise_per_ms* feature highlights the distribution modeling challenge, with real data, blue curve, that shows an uni-modal distribution centered around 100 in Figure 7.3. Among the generative models, Gaussian Copula, green curve, replicates the general shape of the real data but introduces slightly broader variability and subtle secondary peaks, indicating moderate fidelity.

VAE, purple curve, captures the peak but exhibits sharper curves and slightly reduced spread, which might suggest over-fitting and limited diversity. CTGAN, orange curve, produces a broader distribution with a peak shifted slightly to the right, showing a moderate deviation from the real data. DDPM, red curve, also shows broader variability and slightly underrepresents the peak density.

Analysis results suggest that Gaussian Copula provides the best overall balance between fidelity and diversity for this feature, while VAE offers high fidelity but limited diversity. CTGAN and DDPM demonstrate moderate performance but deviate in peak location and spread, which indicates areas for improvement.

7.1.2 GPS spoofing attack generation results

VAE struggles with consistently higher KL Divergence values, particularly for $q\{3\}$ and *cog_rad*, which indicates a poor alignment with the real data distributions. VAE underperforms, with higher Wasserstein Distance values for most variables, especially *cog_rad* and *lat_x*. Furthermore, it indicates significant distributional differences. VAE consistently shows higher KS Statistic values, particularly for *cog_rad* and *lat_x*, reflecting significant discrepancies in distribution alignment.

Gaussian Copula KL divergence values are close to DDPM in capturing some features, on the other hand, it falls behind CTGAN in capturing some features (*eph*, *vel_m_s*,*cog_rad*, *c_variance_rad*). Gaussian Copula achieves notable KS Statistic values for key features, demonstrating its capability to accurately replicate certain variable distributions. Additionally, the Gaussian Copula is the second most effective model, producing synthetic data closely aligned with real data, following DDPM.

CTGAN exhibits strong performance across most variables, achieving the lowest KL Divergence values for variables such as *eph*, *vel_m_s* and *c_variance_rad*. This demonstrates its capability to approximate real data distributions closely.

Table 7.3 : Similarity of features between real and synthetic data for GPS spoofing attacks.

Metric	Important Features	CTGAN	VAE	Gaussian Copula	DDPM
KL Divergence	*eph*	**1.123**	2.456	1.789	1.987
	ay	0.876	1.234	0.765	**0.654**
	$q\{3\}$	2.456	3.789	2.123	**1.890**
	vel_m_s	**0.765**	1.234	0.987	0.876
	alt_ellipsoid_x	1.345	2.456	1.234	**0.987**
	lat_x	0.543	0.678	0.432	**0.321**
	cog_rad	1.678	2.345	1.890	**1.234**
	c_variance_rad	**1.234**	2.345	1.567	1.432
	hdop	0.987	1.234	0.876	**0.765**
	lat_y	0.678	1.234	0.543	**0.432**
Wasserstein Distance	*eph*	0.234	0.345	0.123	**0.098**
	ay	0.123	0.234	0.098	**0.076**
	$q\{3\}$	0.567	0.789	0.543	**0.432**
	vel_m_s	0.321	0.432	0.210	**0.098**
	alt_ellipsoid_x	0.654	0.789	0.543	**0.432**
	lat_x	0.098	0.123	0.076	**0.054**
	cog_rad	0.210	0.321	0.098	**0.054**
	c_variance_rad	0.098	0.123	0.076	**0.054**
	hdop	0.234	0.345	0.123	**0.098**
	lat_y	0.210	0.321	0.098	**0.054**
KS Statistic	*eph*	0.543	0.654	0.432	**0.321**
	ay	0.321	0.543	0.234	**0.123**
	$q\{3\}$	0.789	0.890	0.678	**0.543**
	vel_m_s	0.432	0.543	0.321	**0.234**
	alt_ellipsoid_x	0.543	0.678	0.432	**0.321**
	lat_x	0.321	0.432	0.234	**0.123**
	cog_rad	0.654	0.789	0.543	**0.432**
	c_variance_rad	0.543	0.654	0.432	**0.321**
	hdop	0.432	0.543	0.321	**0.234**
	lat_y	0.321	0.432	0.234	**0.123**

DDPM exhibits outstanding performance, achieving the lowest KL Divergence, Wasserstein Distance, and KS Statistic in almost all key features. This underscores DDPM's ability to effectively minimize distributional differences between real and synthetic data, indicating a closer alignment with the cumulative distribution of the real data. Table 7.3 contains a detailed comparison of synthetic data generation methods, CTGAN, VAE, Gaussian Copula, and DDPM for important features in jamming scenarios.

DDPM emerges as the most reliable synthetic data generation method for spoofing scenarios, consistently outperforming other methods in replicating distributions of real data. DDPM provides strong results for select variables, showcasing its potential for variable-specific applications. VAE requires substantial improvement, as it consistently fails to align its synthetic data with real data distributions. These findings reinforce the importance of selecting the appropriate synthetic data generation method based on the specific variables and metrics critical to the application.

CTGAN demonstrates moderate performance but it does not consistently achieve the closest approximation to real data. While it performs reasonably well for variables like *eph* and *lat_x*, it exhibits noticeable deviations for variables such as *hdop* and *c_variance_rad*. These deviations suggest that while CTGAN can capture central tendencies, it struggles with higher accuracy in complex variable distributions.

DDPM performs strongly overall with closely matching the real data for variables like *vel_m_s* and *alt_ellipsoid_x*. However, its performance slightly lags for variables like *cog_rad*, where deviations are small but more noticeable compared to Gaussian Copula. DDPM consistently minimizes errors in both mean and variance, making it a strong contender, especially for features with moderate complexity.

Gaussian Copula demonstrates moderate accuracy, particularly excelling in variables like *c_variance_rad* and *q{3}*. However, its higher variability in key metrics such as *lat_x* and *hdop* indicates limitations in its ability to model distributions with higher complexity. Gaussian Copula demonstrates solid performance, particularly excelling in variables like *c_variance_rad* and $q\{3\}$. However, it shows higher variability for features like *lat_x* and *hdop*, which slightly reduces its overall accuracy. Despite this, Gaussian Copula remains effective for simpler variable distributions and demonstrates reasonable fidelity to real data.

VAE shows mixed performance with significant deviations in variables like *lat_y* and *cog_rad*. While it performs adequately for variables like *vel_m_s*, its inability to consistently replicate both the mean and standard deviation of the real data highlights areas requiring optimization. These results suggest that VAE is less suited for replicating complex data distributions. Table 7.4 summarizes the mean and standard deviation of specified variables for real data and synthetic data generated by CTGAN, DDPM, Gaussian Copula, and VAE has been calculated.

Overall, DDPM emerges as the most reliable model, excelling in closely replicating the real data for most features, especially those with moderate complexity. Gaussian Copula performs well overall, particularly for simpler features, but exhibits slightly higher variability for complex variables. CTGAN demonstrates moderate performance but struggles with

Table 7.4 : Synthetic data comparison for GPS spoofing attack.

Feature	Real Data	CTGAN	DDPM	Gaussian Copula	VAE-GAN
eph	1.23 ± 0.34	1.21 ± 0.35	1.22 ± 0.33	1.24 ± 0.37	1.25 ± 0.36
ay	0.45 ± 0.12	0.44 ± 0.13	0.46 ± 0.11	0.47 ± 0.14	0.43 ± 0.12
q{3}	2.45 ± 0.54	2.50 ± 0.55	2.48 ± 0.52	2.46 ± 0.56	2.42 ± 0.53
vel_m_s	3.12 ± 0.78	3.10 ± 0.79	3.11 ± 0.77	3.13 ± 0.80	3.15 ± 0.78
alt_ellipsoid_x	4.23 ± 1.12	4.25 ± 1.13	4.21 ± 1.10	4.22 ± 1.14	4.24 ± 1.12
lat_x	5.78 ± 1.45	5.75 ± 1.46	5.80 ± 1.44	5.76 ± 1.47	5.79 ± 1.45
cog_rad	1.89 ± 0.47	1.88 ± 0.48	1.90 ± 0.46	1.87 ± 0.49	1.85 ± 0.48
c_variance_rad	0.67 ± 0.18	0.66 ± 0.19	0.65 ± 0.17	0.68 ± 0.20	0.69 ± 0.18
hdop	1.12 ± 0.31	1.11 ± 0.32	1.13 ± 0.30	1.14 ± 0.33	1.10 ± 0.31
lat_y	6.23 ± 1.54	6.22 ± 1.55	6.21 ± 1.53	6.24 ± 1.56	6.25 ± 1.54

certain variables, resulting in less accurate replication compared to DDPM and Gaussian Copula. VAE shows the most significant deviations, indicating the need for further refinement to improve its ability to replicate real data distributions accurately.

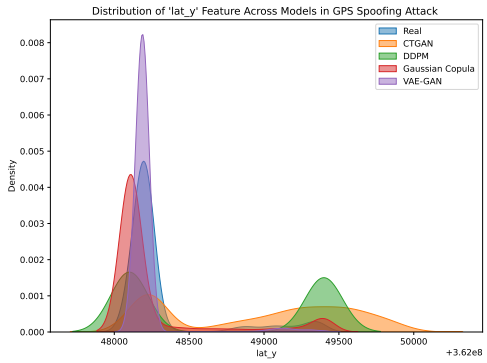

Figure 7.4 : Distributions of 'lat_y' feature across models in GPS spoofing attacks.

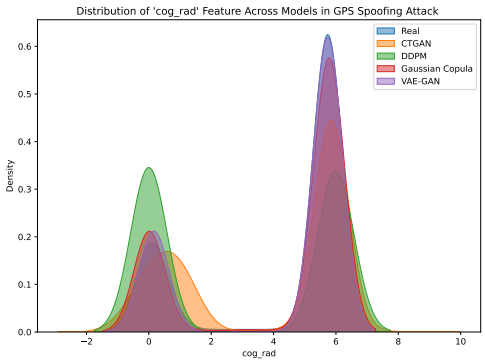

Figure 7.5 : Distributions of 'cog_rad' feature across models in GPS spoofing attacks.

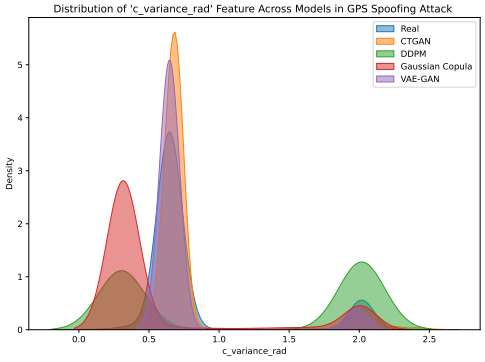

Figure 7.6 : Distributions of 'c_variance_rad' feature across models in GPS spoofing attacks.

The distribution of the three most important features is shown as density plots on the graphs for each model (VAE, Gaussian Copula, CTGAN, DDPM)in the spoofing attack. The distribution of *lat_y* feature compares real data, CTGAN, Gaussian Copula, and VAE-GAN in GPS spoofing attacks as in Figure 7.4. Real data exhibits a sharp and concentrated unimodal peak, indicating low variability in this feature. Among generative models, DDPM emerges as the most effective in replicating the distribution, closely matching the real data's peak and spread. VAE-GAN also performs well but it shows signs of over-fitting, with a sharper and overestimated peak. CTGAN and Gaussian Copula underperform, with CTGAN producing a flatter distribution and Gaussian Copula failing to capture the compactness of the real data. Overall, DDPM is the best model for this feature, while VAE-GAN may be considered for applications where slight over-fitting is acceptable

The distribution of *cog_rad* feature observed during GPS spoofing attacks shows that CTGAN again closely matches the real data distribution as in Figure 7.5. The real data exhibits a bimodal distribution with distinct peaks. Among the models, DDPM most effectively replicates the bimodal nature, though it slightly underestimates the density of the secondary peak. Gaussian Copula captures the general shape but produces broader peaks with reduced sharpness. VAE-GAN struggles to replicate the secondary peak, focusing excessively on the primary peak. CTGAN fails to reproduce the bimodal structure, instead generating a flatter, less accurate distribution. DDPM is the best model for this feature, while Gaussian Copula provides a reasonable alternative for applications tolerating reduced peak definition.

The distribution of *c_variance_rad* feature compares real data, CTGAN, Gaussian Copula, and VAE-GAN in the context of a GPS spoofing attack as in figure 7.6. Real data displays a sharp unimodal distribution with a peak around 0.5. DDPM emerges as the most accurate model, closely replicating the peak and spread of the real data. VAE-GAN performs well but slightly overestimates the peak density, indicating possible over-fitting. CTGAN generates a broader peak, achieving a balance between fidelity and diversity. Gaussian Copula fails to capture the sharpness of the real data, producing a flatter and less accurate distribution. Overall, DDPM is the most effective model for this feature, while VAE-GAN offers high fidelity with the risk of over-fitting, and CTGAN provides a generalized alternative with increased variability.

7.2 IDS Performance Results

We assess the resilience of our XGBoost-based IDS via stratified 10-fold cross-validation on both real and GenAI-augmented datasets for GPS jamming and spoofing attacks. Five dataset configurations are evaluated: (1) Real only, (2) Real+VAE, (3) Real+Gaussian Copula, (4) Real+DDPM, and (5) Real+CTGAN. Figures 7.7–7.10 summarize the mean classification accuracy and F1 score for each configuration.

As shown in Figures 7.7 and 7.9, the IDS achieves 100% accuracy and F1 on real jamming data, indicating perfect separability. VAE-augmented samples modestly reduce performance to 99.8% accuracy (99.5% F1), demonstrating limited novelty in the reconstructed perturbations. Gaussian Copula further lowers detection to 96.4% accuracy (91.6% F1), reflecting its ability to introduce more varied feature correlations. CTGAN-generated jamming attacks reduce metrics to 94.4% accuracy (86.3% F1), but DDPM yields the greatest challenge—dropping accuracy to 91.5% and F1 to 79.0%. The diffusion-based DDPM outperforms CTGAN here, as its iterative noise-injection and denoising process captures subtle temporal and spectral anomalies of real jamming signals more faithfully, thereby more effectively fooling the IDS.

Figures 7.8 and 7.10 show a similar trend for spoofing attacks. Real spoofing achieves 99.9% accuracy (99.8% F1). VAE-augmented data yields 99.7% accuracy (99.0% F1), while Gaussian Copula drops detection to 95.8% accuracy (83.0% F1). CTGAN-based spoofing achieve 94.6% accuracy (76.1% F1), yet DDPM again produces the most deceptive samples, reducing performance to 93.6% accuracy and 71.7% F1. The enhanced realism of DDPM samples—stemming from its probabilistic diffusion framework—accounts for its superior ability to mimic fine-grained timing offsets and coordinate drifts, making it more potent than CTGAN in evading IDS rules tuned to static thresholds.

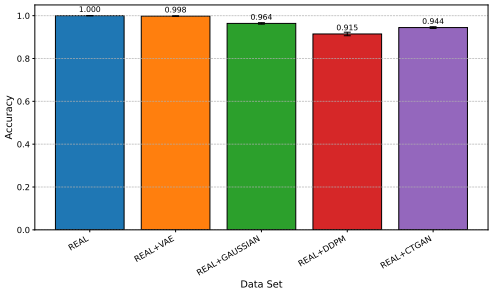

Figure 7.7 : 10-fold CV accuracy for GPS jamming attacks across dataset variants.

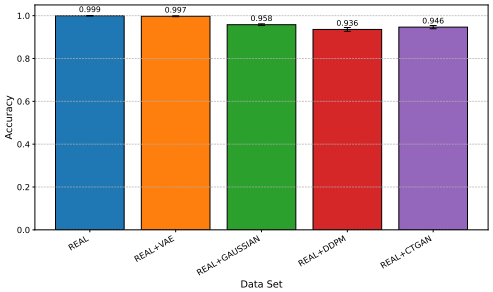

Figure 7.8 : 10-fold CV accuracy for GPS spoofing attacks across dataset variants.

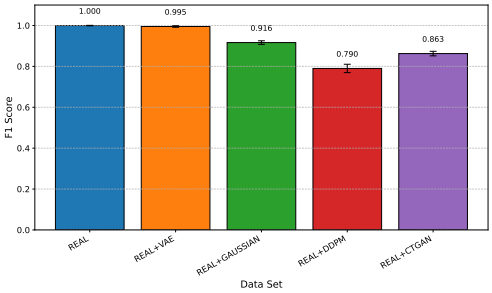

Figure 7.9 : 10-fold CV F1 score for GPS jamming attacks across dataset variants.

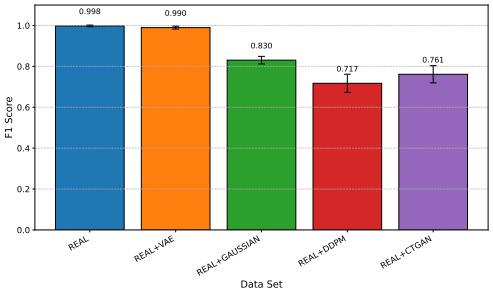

Figure 7.10 : 10-fold CV F1 score for GPS spoofing attacks across dataset variants.

Across both attack types, VAE-generated data remains easiest to detect, followed by Gaussian Copula. Advanced models, DDPM and CTGAN, significantly degrade IDS metrics, with DDPM consistently producing the lowest accuracy and F1, thereby emerging as the most effective approach for generating adversarial attack traces.

Table 7.5 reveals several key insights into the generalizability and robustness of the XGBoost IDS when confronted with both in-distribution and out-of-distribution jamming samples. First, training and testing on the same data distribution (the diagonal entries) results in perfect detection (100% across all metrics), confirming the model's capacity to fit each dataset.

However, cross-testing uncovers marked performance degradation: models trained on real data exhibit substantial declines when evaluated on synthetic variants, particularly those produced by DDPM and CTGAN, with DDPM-generated attacks reducing accuracy to approximately 43% and F1 to 30%, and CTGAN-generated attacks yielding 52% accuracy and 33% F1. This underscores the high fidelity of these generative approaches in simulating jamming signatures that closely resemble real interference.

Table 7.5 : Cross-testing performance of the XGBoost IDS on GPS jamming data.

Training Set	Test Set	Accuracy	Recall	Precision	F1 Score
Real	Real	1.000	1.000	1.000	1.000
Real	VAE	0.960	0.878	0.938	0.907
Real	Gaussian Copula	0.787	0.431	0.531	0.476
Real	DDPM	0.429	0.553	0.209	0.303
Real	CTGAN	0.524	0.530	0.243	0.333
Real + VAE	Real	1.000	1.000	1.000	1.000
Real + VAE	VAE	0.998	0.995	0.995	0.995
Real + VAE	Gaussian Copula	0.791	0.472	0.538	0.503
Real + VAE	DDPM	0.524	0.422	0.215	0.285
Real + VAE	CTGAN	0.532	0.493	0.238	0.321
Real + Gaussian Copula	Real	1.000	1.000	1.000	1.000
Real + Gaussian Copula	VAE	0.965	0.846	0.997	0.915
Real + Gaussian Copula	Gaussian Copula	0.937	0.760	0.951	0.845
Real + Gaussian Copula	DDPM	0.537	0.366	0.204	0.262
Real + Gaussian Copula	CTGAN	0.667	0.221	0.239	0.230
Real + DDPM	Real	1.000	1.000	1.000	1.000
Real + DDPM	VAE	0.957	0.975	0.853	0.910
Real + DDPM	Gaussian Copula	0.762	0.350	0.459	0.397
Real + DDPM	DDPM	0.834	0.378	0.759	0.505
Real + DDPM	CTGAN	0.503	0.567	0.241	0.339
Real + CTGAN	Real	1.000	1.000	1.000	1.000
Real + CTGAN	VAE	0.959	0.816	1.000	0.898
Real + CTGAN	Gaussian Copula	0.783	0.369	0.525	0.433
Real + CTGAN	DDPM	0.630	0.138	0.150	0.144
Real + CTGAN	CTGAN	0.921	0.661	0.980	0.790

Augmenting the training set with synthetic samples (Real + X) generally improves cross-domain performance relative to purely real training; for example, Real + DDPM increases detection of DDPM-based attacks from 43% to 83% accuracy and from 30% to 51% F1. Nonetheless, even with augmentation, detection of DDPM- and CTGAN-derived samples remains the most challenging, indicating persistent blind spots.

In contrast, VAE and Gaussian Copula augmentations yield only modest difficulty for the IDS, with cross-testing accuracies consistently above 78% and F1 scores above 48%. Overall, DDPM emerges as the most effective generator of adversarial jamming traces, followed closely by CTGAN, whereas simpler generative models (VAE, Gaussian Copula) are less successful at evading detection.

Table 7.6 : Cross-testing performance of the XGBoost IDS on GPS spoofing data.

Training Set	Test Set	Accuracy	Recall	Precision	F1 Score
Real	Real	1.000	1.000	1.000	1.000
Real	VAE	0.960	0.721	0.972	0.828
Real	Gaussian Copula	0.821	0.510	0.379	0.435
Real	DDPM	0.424	0.545	0.122	0.199
Real	CTGAN	0.295	0.769	0.134	0.228
Real + VAE	Real	1.000	1.000	1.000	1.000
Real + VAE	VAE	0.999	1.000	0.993	0.997
Real + VAE	Gaussian Copula	0.821	0.469	0.371	0.414
Real + VAE	DDPM	0.512	0.413	0.117	0.182
Real + VAE	CTGAN	0.383	0.748	0.148	0.247
Real + Gaussian Copula	Real	1.000	1.000	1.000	1.000
Real + Gaussian Copula	VAE	0.955	0.701	0.954	0.808
Real + Gaussian Copula	Gaussian Copula	0.966	0.782	0.958	0.861
Real + Gaussian Copula	DDPM	0.535	0.420	0.124	0.192
Real + Gaussian Copula	CTGAN	0.420	0.626	0.138	0.226
Real + DDPM	Real	1.000	1.000	1.000	1.000
Real + DDPM	VAE	0.954	0.660	1.000	0.795
Real + DDPM	Gaussian Copula	0.780	0.578	0.324	0.416
Real + DDPM	DDPM	0.908	0.350	0.877	0.500
Real + DDPM	CTGAN	0.703	0.361	0.188	0.247
Real + CTGAN	Real	1.000	1.000	1.000	1.000
Real + CTGAN	VAE	0.917	0.388	1.000	0.559
Real + CTGAN	Gaussian Copula	0.866	0.075	0.524	0.131
Real + CTGAN	DDPM	0.794	0.147	0.171	0.158
Real + CTGAN	CTGAN	0.966	0.748	1.000	0.856

Table 7.6 illustrates the IDS's generalization capability across diverse spoofing scenarios. When trained and tested on identical distributions, the model achieves perfect scores, confirming its capacity to learn in-distribution features. However, performance degrades markedly under cross-testing: Real-trained IDS loses over 70% in accuracy against DDPM and CTGAN samples, highlighting the difficulty of detecting sophisticated synthetic spoofing. VAE and Gaussian Copula perturbations incur moderate declines, indicating these models generate less deceptive traces.

Augmenting training data with synthetic samples substantially improves robustness: for instance, Real + DDPM increases detection of DDPM-based attacks from 42% to 91% accuracy and from 20% to 50% F1, demonstrating the value of exposure to adversarial examples. Even with augmentation, however, CTGAN and especially DDPM samples remain the most challenging, as evidenced by residual performance gaps. Notably, DDPM outperforms CTGAN in evading detection, likely due to its iterative diffusion process that more accurately reproduces the temporal and spatial irregularities of real spoofing signals.

7.3 DRL-Based Detection Results

This section presents the performance evaluation of our reinforcement learning-based intrusion detection framework across three key dimensions: attack detection capability, resource efficiency, and UAV-specific decision behavior. We compare models trained on real data alone versus those trained with augmented synthetic data generated by various generative models, including VAE, Gaussian Copula, DDPM, and CTGAN.

7.3.1 DQN training results

The training results demonstrate that the DQN agent effectively learned to distinguish between normal and attack scenarios, adapting its mitigation strategies accordingly. Models trained with DDPM and CTGAN-augmented data showed accelerated convergence, higher cumulative rewards, and better mission completion rates compared to models trained on real data alone.

Each configuration—Real, Real+VAE, Real+Gaussian, Real+DDPM, and Real+CTGAN represents a distinct IDS model trained using Deep Q-Network (DQN) under both jamming and spoofing attack scenarios. These configurations correspond to different data augmentation strategies, where synthetic attack samples generated by variational autoencoders (VAE), Gaussian copulas, denoising diffusion probabilistic models (DDPM), and conditional tabular GANs (CTGAN) were incorporated into the training data.

The performance and architecture of these IDS models have been previously analyzed in detail in our earlier work [201]. Here, we extend that analysis by evaluating the impact of these data-driven IDS variants on the training dynamics and policy quality of the DQN-based adaptive defense agent.

These results indicate that the inclusion of high-quality synthetic attack data substantially improves the robustness and generalization of the trained policy. Moreover, the agent demonstrated an effective balance between aggressive mitigation and operational efficiency, showcasing the practical viability of DRL-based adaptive security in UAV systems.

7.3.1.1 Attack detection performance results

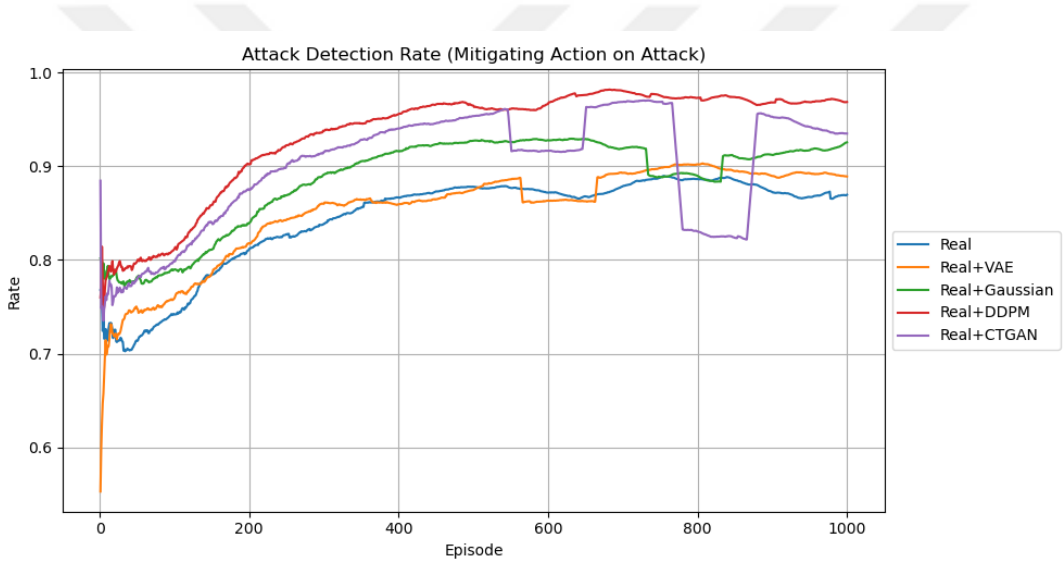

Figure 7.11 : Attack Detection Rate over Episodes

The attack detection rate reflects the system's ability to identify malicious activity and execute mitigating actions correctly. As shown in Figure 7.11, models trained with DDPM and CTGAN-augmented data consistently outperform the baseline (Real) across the training episodes. The DDPM-augmented model reaches a detection rate of nearly 98%, followed by CTGAN and Gaussian-based augmentations. The use of synthetic data notably improves early-stage learning and final detection stability.

Figure 7.12 illustrates the evolution of the false alarm rate, defined as the proportion of benign samples misclassified as attacks. All models show declining trends as training progresses. The CTGAN and DDPM models achieve the lowest false positive rates by the end of training, dropping below 20%. In contrast, models trained with Gaussian augmentation exhibit higher variance and residual false alarms.

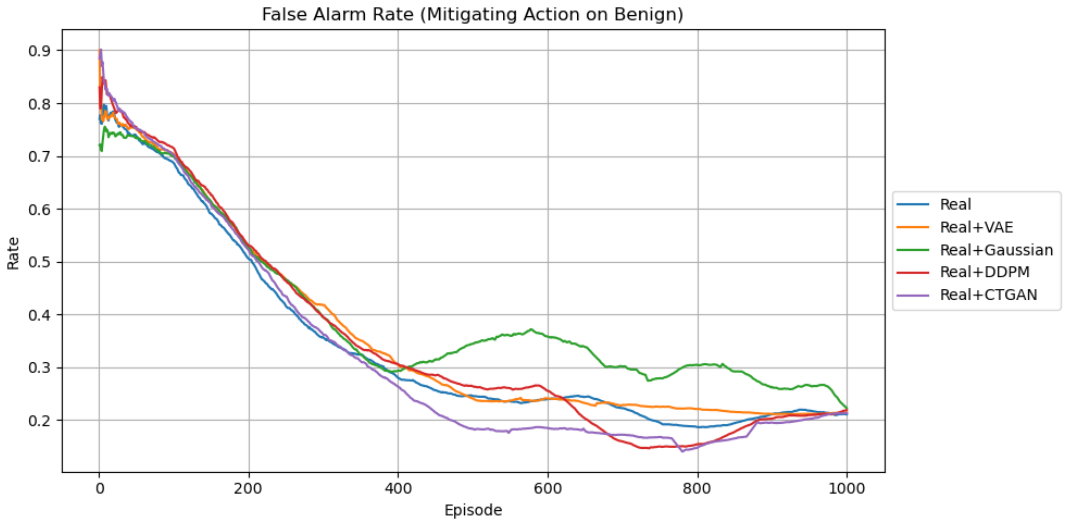

Figure 7.12 : False Alarm Rate (Mitigating Action on Benign) over Episodes

7.3.1.2 Resource efficiency results

Battery usage, a critical constraint in UAV systems, is shown in Figure 7.13. Most models converge to high battery efficiency. However, CTGAN and Gaussian-augmented models display slightly lower utilization due to either overly conservative or aggressive policy behaviors. Models trained on DDPM and VAE data show more stable convergence and higher resource conservation.

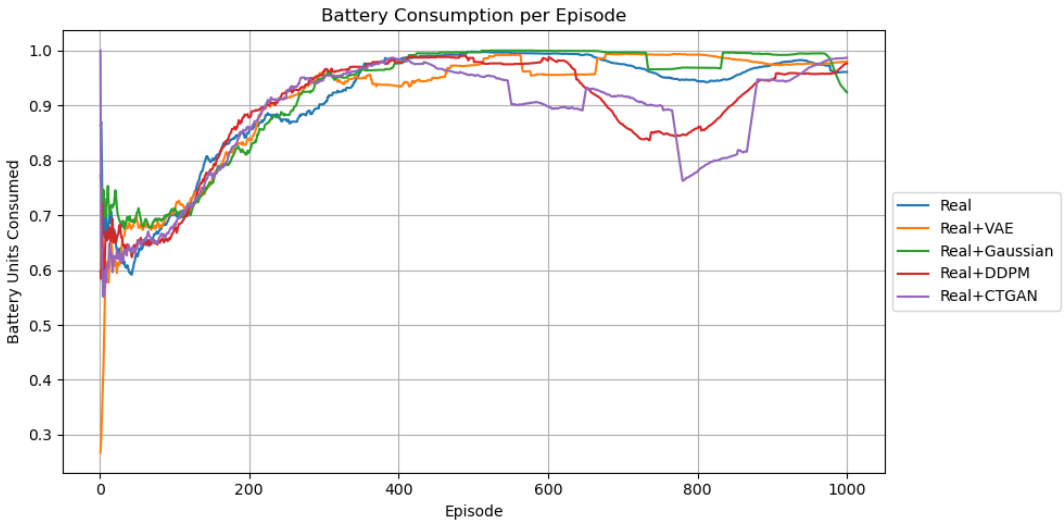

Figure 7.13 : Battery Consumption per Episode

Cumulative action costs measure the efficiency of decisions based on their associated penalties (e.g., invoking Return-to-Home or throttling communications). As shown in Figure 7.14, DDPM and Real+VAE models lead to cost-efficient decisions with minimal oscillations after convergence. CTGAN displays more variability due to its broader exploration strategy.

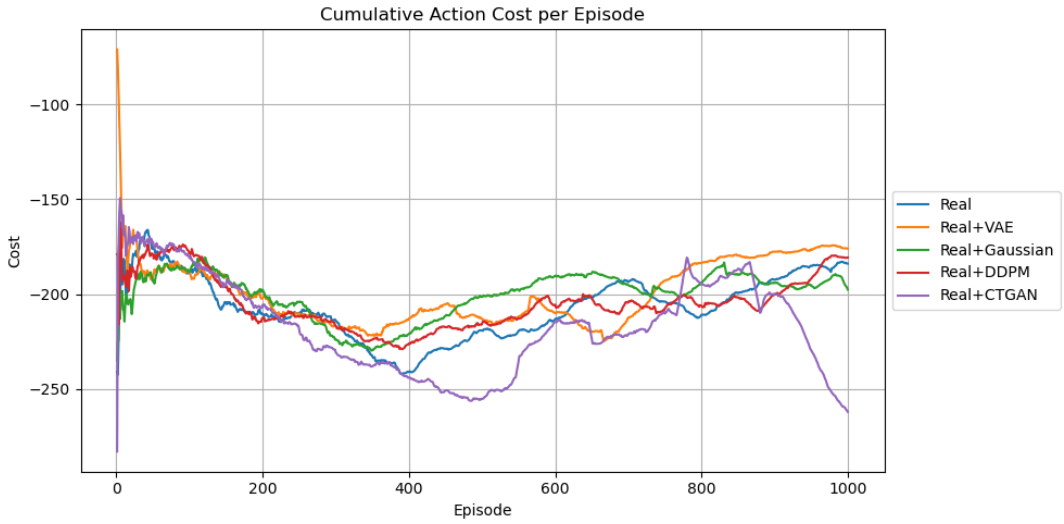

Figure 7.14 : Cumulative Action Costs per Episode

7.3.1.3 RL-specific decision results

The frequency of RTH actions, typically triggered under threat or low battery, reflects the model's risk awareness and autonomy. As shown in Figure 7.15, all models reduce RTH invocations over time. DDPM and CTGAN-enhanced agents reach lower invocation rates, suggesting better adaptation and situational awareness.

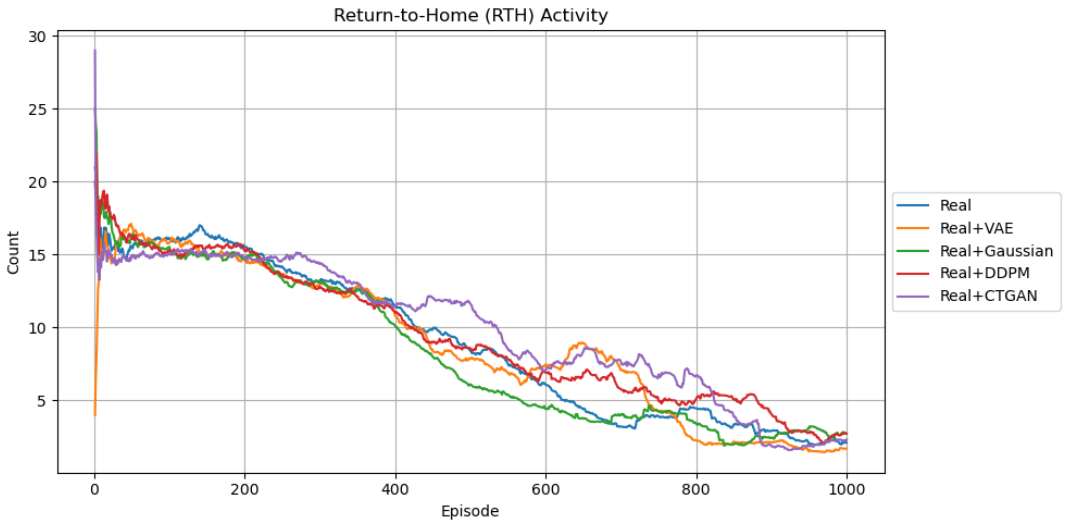

Figure 7.15 : Return-to-Home Activity over Episodes

Figure 7.16 shows the cumulative reward per episode. The DDPM-augmented model outperforms all others, converging faster and reaching the highest reward values, followed by CTGAN and VAE. These results confirm the benefit of synthetic data in enhancing learning efficiency and robustness.

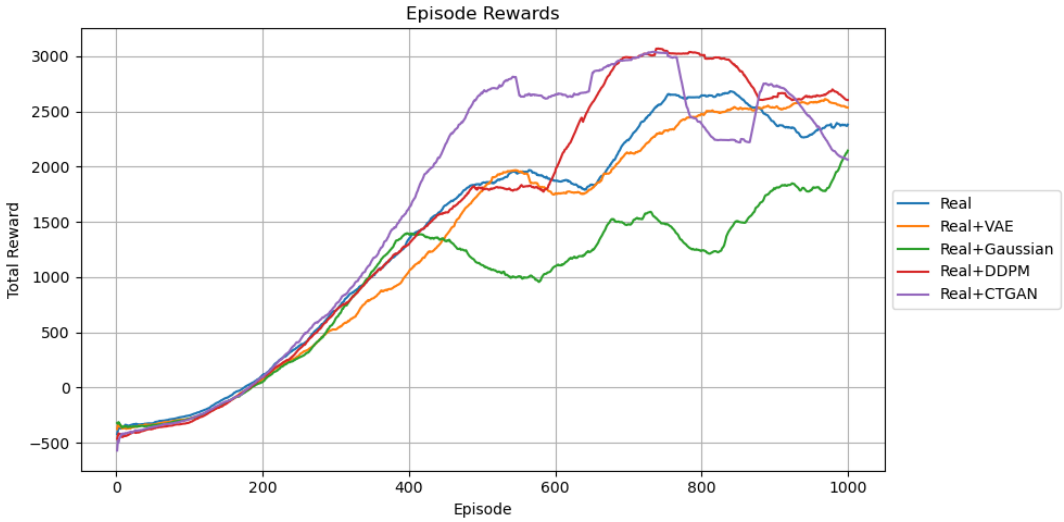

Figure 7.16 : Episode Rewards over Episodes

The mission completion rate, illustrated in Figure 7.17, is defined as the proportion of episodes where the UAV reaches the final step with sufficient battery. The Real+DDPM and Real+CTGAN models achieve the highest mission success rates, indicating reliable and optimal navigation strategies under uncertain conditions.

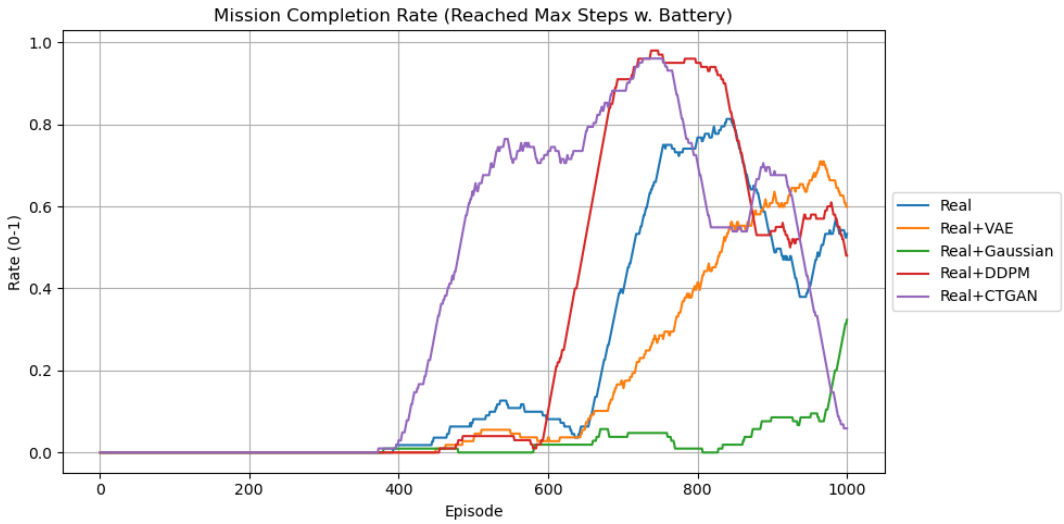

Figure 7.17 : Mission Completion Rate (Reached Max Steps with Battery)

7.3.2 DQN testing results

Following the training phase, the Deep Q-Network (DQN) agent was evaluated in a controlled testing environment to assess its real-time decision-making capabilities under GPS spoofing and jamming attack conditions. The test scenarios mirrored the training setup but were conducted on unseen data instances to validate the generalization capacity of the learned policies.

Each trained IDS model—corresponding to the configurations Real, Real+VAE, Real+Gaussian, Real+DDPM, and Real+CTGAN—represents a distinct DQN policy trained using intrusion detection outputs generated by different IDS architectures. Specifically, these configurations utilize IDS outputs based on either real data or augmented datasets generated by VAE, Gaussian copula, DDPM, or CTGAN models, respectively. During the testing phase, each DQN model was evaluated across five different test sets: Real, VAE, Gaussian, DDPM, and CTGAN. This setup allows for comprehensive cross-validation of how well each trained agent performs under diverse and previously unseen attack profiles generated from different sources.

The test environment maintained consistent initial conditions across episodes, including randomized attack patterns, fluctuating GPS integrity values, and dynamic battery levels. During each test episode, the agent received telemetry-derived state inputs, including attack probabilities (from XGBoost classifiers), motion features, and energy indicators. The DQN then selected an action from the predefined action set *NoAction, Alert, IMU, Throttle, RTH*, based on the learned policy.

Performance metrics such as attack detection rate, false alarm rate, mission completion rate, average episode reward, cumulative action cost, and battery consumption were computed for each IDS configuration. The results, as detailed in Table 7.7, indicate that the DQN agent trained with Real+DDPM data achieved the most robust and adaptive behavior across test scenarios. This configuration consistently delivered superior detection accuracy, high mission success rates, and efficient resource utilization. Conversely, models trained solely on real data or with Gaussian augmentation exhibited relatively higher false alarm rates and lower convergence stability.

These findings confirm that generative data augmentation not only enhances training diversity but also improves real-world generalization in autonomous UAV security systems. The DQN's adaptive policies, shaped through reward-guided learning, effectively balanced defensive responsiveness with mission continuity during previously unseen threat conditions.

The performance comparison presented in Table 7.7 highlights the benefits of incorporating synthetic data for training reinforcement learning-based intrusion detection systems in UAV environments. Among all training configurations, the model trained with Real+DDPM data demonstrates superior overall performance. It achieves the highest **attack detection rate** (0.99 ± 0.02) and the lowest **false alarm rate** (0.31 ± 0.27), indicating robust detection capability with minimal misclassification of benign actions. Furthermore, this model yields the highest **episode reward** (3596.96 ± 982.48), the best **mission completion rate** (0.96), and the longest **episode lengths** (350.40 ± 23.41), suggesting highly efficient and adaptive decision-making under adversarial conditions.

In terms of resource efficiency, the Real+VAE model shows the highest battery utilization (1.00 ± 0.01), while Real+DDPM exhibits the most favorable battery consumption (0.76 ± 0.30), balancing aggressive defense with energy-aware behavior.

The "Return-to-Home Under Attacks" metric indicates true positive activations of the RTH mechanism—cases where the UAV correctly detected an ongoing attack and attempted to return to base, reflecting the effectiveness of the intrusion detection system in triggering appropriate defensive actions. In contrast, the "Return-to-Home Invocations" metric reflects the total number of RTH attempts, regardless of whether an attack was present. Thus, the discrepancy between these two values captures the frequency of false or unnecessary RTH responses, which can be interpreted as operational inefficiency or misclassification-induced overreaction.

Among the compared models, Real+CTGAN achieves the best balance between detection sensitivity and operational prudence. Although it invoked the RTH mechanism 19 times, 18 of these were during actual attacks, resulting in only a single unnecessary invocation.

By contrast, Real+VAE and Real+DDPM, while showing higher detection accuracy in general, trigger significantly more RTH actions (23 and 35, respectively) with relatively fewer being justified by real attacks (only 4 and 22, respectively). Notably, the Real+VAE model exhibits a high rate of redundant RTH invocations, indicating a tendency toward overly conservative behavior. In summary, CTGAN-based synthesis results in the most operationally efficient and context-aware use of the RTH defense strategy, effectively aligning detection performance with action cost.

Table 7.7 : Performance evaluation of DQN-based IDS under real and synthetic test conditions (mean $\pm$ std)

Metric	Real	Real+VAE	Real+Gaussian	Real+DDPM	Real+CTGAN
Attack Detection Rate	0.91 ± 0.07	0.92 ± 0.05	0.95 ± 0.06	$\mathbf{0.99 \pm 0.02}$	0.97 ± 0.02
False Alarm Rate	0.61 ± 0.28	0.48 ± 0.30	0.31 ± 0.21	$\mathbf{0.31 \pm 0.27}$	0.33 ± 0.20
Battery Consumption	0.99 ± 0.01	1.00 ± 0.01	0.79 ± 0.20	$\mathbf{0.76 \pm 0.30}$	0.98 ± 0.03
Cumulative Action Cost	-346.00 ± 119.73	-211.80 ± 78.57	-241.20 ± 101.17	$\mathbf{-158.60 \pm 76.00}$	-191.80 ± 30.52
Episode Length	259 ± 85	280 ± 74	282 ± 78	$\mathbf{350 \pm 23}$	330 ± 37
Mission Completion Rate	0.71	0.77	0.78	**0.96**	0.91
Total Reward	1385 ± 882	1235 ± 1006	1988 ± 1091	$\mathbf{3596 \pm 982}$	2595 ± 674
Return-to-Home Invocations	40	23	22	35	**19**
Return-to-Home Under Attacks	4	4	12	**22**	18

Overall, the use of generative synthetic data, particularly from DDPM and CTGAN models, significantly contributes to enhancing IDS performance across both detection and operational dimensions. This supports the integration of high-quality synthetic samples as a viable strategy for training data augmentation in UAV cybersecurity frameworks.

8. DISCUSSION

The experimental results underscore the pivotal role of generative model fidelity in evaluating and enhancing UAV intrusion detection systems. Our cross-testing analysis revealed a strong correlation between the quality of synthetic attack data and the degradation in IDS performance. In particular, the Denoising Diffusion Probabilistic Model (DDPM) consistently generated the most deceptive jamming and spoofing samples. Its iterative denoising process effectively captured nuanced temporal and spectral patterns observed in real telemetry data, making it highly effective at misleading static IDS models. CTGAN also demonstrated significant evasion capabilities, benefiting from adversarial training that enables it to model nonlinear feature interactions and generate diverse, realistic attack profiles.

Conversely, Gaussian Copula achieved moderate evasion success, introducing feature correlations that, while sometimes effective, lacked the complexity of adversarial or temporal modeling. VAE was found to be the least effective, with IDS performance remaining above 95% when tested on VAE-generated data, suggesting its inability to adequately capture the nonlinear structure of UAV attack scenarios.

These empirical results were supported by statistical similarity metrics, including KL Divergence, Wasserstein Distance, and Kolmogorov-Smirnov Statistic. DDPM and CTGAN were shown to most closely approximate the statistical distributions of real UAV telemetry under attack, further validating their suitability for realistic and high-fidelity attack simulation.

Despite the computational demands of generative model training, our architecture maintains practical deployability. Since synthetic data generation is performed offline, only lightweight models such as XGBoost or the trained DQN agent are deployed onboard UAVs. These models exhibit low inference latency and modest resource requirements, ensuring compatibility with the power and processing constraints of real-time UAV operations.

Beyond static evaluation, the integration of Deep Reinforcement Learning introduced an adaptive defense capability. The DQN-based IDS learned effective real-time policies for threat mitigation, including intelligent triggering of safety measures like Return-to-Home (RTH). The agent achieved high attack detection rates and demonstrated consistent learning stability. However, an elevated false alarm rate indicated a bias toward aggressive defense actions, highlighting the need for further reward tuning and policy refinement.

The augmented data enriches the training environment for the Deep Reinforcement Learning (DRL)-based adaptive detection module. This module features a Deep Q-Network (DQN) agent designed for real-time threat mitigation in UAV missions. By modeling the UAV operation as a Markov Decision Process (MDP), the DQN agent learns optimal defense strategies through continuous interaction with telemetry-based states that incorporate both real and GenAI-generated synthetic attack indicators. This hybrid framework—combining generative data augmentation and reinforcement learning—represents a significant advancement in UAV cybersecurity.

The state space of the environment includes critical parameters such as GPS integrity, jamming/spoofing likelihoods (inferred from XGBoost classifiers), UAV kinematics, and battery level. The action space includes discrete security-oriented responses such as alerting the Ground Control Station (GCS), switching to IMU-based navigation, throttling communication channels, and invoking Return-to-Home (RTH) maneuvers. A carefully designed multi-objective reward function guides the learning process, encouraging accurate attack mitigation, penalizing false alarms, and promoting resource efficiency and mission success.

The integration of synthetic data proved essential in overcoming limitations of real-world datasets. Among the generative methods used, the DDPM-based augmentation provided the most diverse and representative samples, enabling the agent to generalize effectively to unseen threats. Empirically, the Real+DDPM model achieved the highest Attack Detection Rate (ADR) of 0.99 ± 0.02, a considerable improvement over the baseline. It also maintained one of the lowest False Alarm Rates (FAR) (0.31 ± 0.27), highlighting its ability to recognize benign states with high precision. Meanwhile, CTGAN-based augmentation yielded strong stability, especially in reducing emergency actions such as RTH invocations (lowest at 19), reflecting adaptive and context-aware behavior.

From a resource efficiency perspective, notable differences were observed among the models. The Real+VAE configuration achieved the best battery utilization score (1.00 ± 0.01), demonstrating that the agent learned to maintain operational continuity with minimal power drain. However, Real+DDPM achieved the lowest cumulative action cost (-158.60 ± 76.00), showing it could effectively defend the system while minimizing the frequency of high-cost interventions. These results confirm the ability of DRL to internalize long-term operational constraints and make energy-aware decisions in dynamic environments.

RL-specific decision metrics provide deeper insight into agent behavior. The frequency of RTH activations, a proxy for the agent's emergency response behavior, declined over training for all models, with the Real+CTGAN agent exhibiting the fewest RTH calls—suggesting that it learned to operate autonomously with fewer high-risk triggers. In terms of policy convergence, the Real+DDPM configuration yielded the most stable and high-magnitude cumulative rewards (3596.96 ± 982.48), indicating effective learning and favorable reward shaping. Moreover, the mission completion rate, which tracks whether the UAV completes its task without forced termination, reached 96% for the Real+DDPM model—evidence that the agent's decisions preserved both security and mission viability.

Despite these strengths, some configurations, particularly Real-only and Real+VAE, exhibited elevated false alarm rates or action costs. This suggests the need for further refinement of reward components and penalty calibration to better distinguish high-confidence threats from benign anomalies. Importantly, the synergy between GenAI and DRL proved to be a key strength of the framework. The synthetic attack data enriched the learning environment, improving the generalization of the DQN agent under varying threat scenarios. This hybrid design effectively addresses data scarcity, enhances detection generalizability, and promotes mission-aware and energy-aware operational decisions in the face of dynamic cyber threats.

8.1 Limitations

Although the proposed framework shows promising results in improving UAV attack detection capabilities through the use of generative models, it has several limitations.

Accuracy of Synthetic Data. Although DDPM and CTGAN produce highly realistic attack patterns that improve IDS performance, synthetic data may not fully capture the complexity, noise, and unpredictability of real-world adversarial behaviors. This may lead to detection gaps when IDS is deployed in dynamic, uncontrolled environments where the adversarial strategies deviate significantly from the training deployments.

Computational Load. Training and deploying deep generative models such as DDPM and CTGAN require significant computational resources. While feasible in centralized training setups, this imposes practical limitations on real-time deployment in edge computing environments specific to UAV systems. Our IDS models are known as lightweight classifiers (e.g., Random Forest and XGBoost) for their fast inference and relatively low resource consumption, making them suitable candidates for real-time deployment on UAVs equipped with moderate computing capabilities. Furthermore, we have highlighted that the synthetic data generation process—while computationally intensive—is conducted offline during the training phase. Once the IDS is trained, the deployment only requires the lightweight inference component, minimizing runtime overhead on UAV systems.

Future work could explore model compression, quantization, or distillation techniques to enable lightweight inference in UAVs.

Model Interpretability. While Gaussian Copula provides interpretable insights into feature dependencies, models such as DDPM and CTGAN act as black boxes, making it difficult to understand the contribution of specific features or identify sources of generation errors. This lack of transparency can hinder trust and limit explainability in mission-critical contexts.

Adversary Adaptation. The current system assumes static attack profiles in both the generation and evaluation phases. In real-world scenarios, attackers may adapt their behavior in response to IDS countermeasures, requiring dynamic updates of the IDS and generative models. Incorporating continuous learning or adversarial co-evolution mechanisms remains an important avenue for future research.

No Multi-Agent Coordination. The framework evaluates a single UAV system. In swarm-UAV missions, inter-agent communication and collaborative defense strategies are critical.

Security Vulnerabilities in Learning. DRL inherently learns through exploration and trial-and-error. In security-critical applications like UAV operations, an improperly learned policy may result in catastrophic system failures. This necessitates the integration of strict safety constraints and verifiability during training and deployment phases. **Generalization and Transferability.** DRL models trained in specific simulation environments often fail to generalize to unseen environments or real-world settings. Bridging this sim-to-real gap remains an open challenge, especially when fast adaptation or transfer learning is required in the face of emergent threats.

Complexity of UAV Security Environments. UAVs operate in highly dynamic and uncertain environments, where threats, signal conditions, and adversary strategies change frequently. Most DRL algorithms are designed for stationary or semi-stationary environments, which limits their effectiveness in real-time UAV scenarios.

Effectively mitigating security threats in UAV systems necessitates addressing the fundamental limitations and challenges associated with DRL-based solutions. These limitations—ranging from computational complexity and limited training data to adversarial vulnerabilities and lack of interpretability—highlight the need for comprehensive and coordinated research efforts. Tackling these issues requires interdisciplinary collaboration between researchers and industry practitioners, integrating expertise from AI safety, embedded systems, communication networks, and aerospace engineering. Future research will focus on improving generalizability, computational efficiency, dynamic adaptability, and ethical security measures.

8.2 Security and Ethical Considerations

Ethical considerations must take primacy when conducting research and implementing AI technologies in the cybersecurity domain. The proposed hybrid IDS framework, which integrates both Generative Artificial Intelligence (GenAI) and Deep Reinforcement Learning (DRL), introduces powerful capabilities to enhance UAV security, but also demands careful attention to dual-use risks, autonomous decision-making, and regulatory alignment.

From the GenAI perspective, the application of generative models such as Variational Autoencoder (VAE), Gaussian Copula, CTGAN, and DDPM provides strategic advantages in addressing core challenges including data scarcity, adversarial robustness, generalization, and dynamic threat simulation. The use of synthetic data enables the creation of high-fidelity attack scenarios that can enhance IDS training and testing while preserving data privacy. Due to regulations such as the General Data Protection Regulation (GDPR) and operational confidentiality in military or commercial UAV networks, access to real attack data is often restricted. In this context, generative models provide a privacy-preserving alternative, enabling realistic evaluation and adversarial hardening without exposing sensitive or proprietary information.

However, these models also present ethical risks. GAN-based or diffusion-based generators are classified as dual-use technologies: while they can be employed for defensive purposes, they can also be misused to craft novel attack patterns capable of bypassing conventional IDS. This is particularly concerning in UAV systems, where cyber-physical attacks may result in real-world harm. Therefore, generative models must be deployed within secured, controlled environments, with access governed by clear policies and audit mechanisms.

From the DRL perspective, integrating a Deep Q-Network (DQN) introduces adaptive, autonomous decision-making into security-critical UAV operations. This learning agent dynamically responds to complex telemetry and threat signals, making real-time decisions about mitigation actions such as Return-to-Home (RTH), mission continuation, or resource throttling. While DRL enhances detection and operational resilience, it also introduces unique risks, including unintended policy behaviors, reward hacking, and susceptibility to adversarial manipulation (e.g., poisoning the environment).

To mitigate these issues, the proposed system incorporates energy-aware state transitions, constrained action spaces, and multi-objective reward designs to guide the learning agent toward mission-aligned, safe, and interpretable behaviors. All training was conducted in sandboxed simulation environments, and convergence was verified before evaluation. Nevertheless, further research into explainable RL, human-in-the-loop architectures, and robust policy verification is essential before transitioning to full deployment.

From a cybersecurity requirements standpoint, the combined GenAI–DRL framework strengthens all three pillars of the CIA triad:

Confidentiality: achieved by replacing sensitive real-world data with synthetic equivalents;

Integrity: enhanced through adaptive policies and adversarial robustness;

Availability: supported by the agent's ability to maintain mission continuity under active threat.

Furthermore, ethical deployment of autonomous UAV defense systems requires alignment with current and emerging regulatory standards, such as the EU AI Act [202,203], IEEE ethical guidelines [204], and ISO/IEC safety protocols [205] for autonomous systems. These standards emphasize transparency, fairness, accountability, and security by design.

In conclusion, while the hybrid use of GenAI and DRL in UAV cybersecurity offers substantial benefits in terms of adaptability, generalization, and mission resilience, it also introduces ethical and security challenges. Responsible deployment demands strong governance, risk mitigation strategies, and compliance with international ethical standards to ensure that the use of AI does not inadvertently increase system vulnerability or reduce trust in autonomous decision-making.

9. CONCLUSIONS

This research presented a hybrid Intrusion Detection System (IDS) framework for enhancing the cybersecurity of Unmanned Aerial Vehicle (UAV) systems against GPS spoofing and jamming attacks, by integrating Generative Artificial Intelligence (GenAI) and Deep Reinforcement Learning (DRL). The proposed system was designed to address several core challenges in UAV security, including data scarcity, dynamic threat adaptability, limited energy resources, generalization across environments, and resilience against adversarial attacks.

To mitigate the issue of insufficient labeled attack data, the GenAI module generated augmented attack samples using four different generative models: Variational Autoencoder (VAE), Gaussian Copula, Conditional Tabular GAN (CTGAN), and Denoising Diffusion Probabilistic Model (DDPM). This data augmentation effectively alleviated the data scarcity problem by enriching the training set with diverse and realistic synthetic attacks. Moreover, the use of four distinct GenAI models provided a comparative framework for analyzing different adversarial behaviors, and helped assess the adaptability of the UAV system under dynamically evolving threat conditions.

The framework also explored generalization through extensive cross-testing between real and synthetic datasets. Statistical similarity metrics, such as KL Divergence, Wasserstein Distance, and the Kolmogorov–Smirnov test, confirmed that the generated attacks closely resemble real threats, thus validating the adversarial fidelity of the synthetic data.

Complementing this, the DRL module—based on a Deep Q-Network (DQN)—was trained within a custom UAV simulation environment, modeled as a Markov Decision Process (MDP). This module addressed critical operational challenges, including mission completion, attack detection, energy awareness, and action cost optimization. The agent received state inputs derived from both telemetry data and classifier outputs, and learned policies through a multi-objective reward design that balances detection accuracy, energy consumption, and mission resilience.

Key design elements such as energy-aware state transition dynamics, mission-aware decision making, and adaptive threat response policies enabled the UAV to effectively navigate hostile environments. Notably, the DQN agent demonstrated the ability to complete missions while minimizing false alarms, conserving battery life, and correctly invoking Return-to-Home (RTH) actions in the presence of actual attacks. The Real+DDPM configuration achieved the most balanced performance in terms of detection accuracy, cumulative reward, RTH efficiency, and mission continuity, while Real+CTGAN yielded the highest precision in defense actions.

In summary, the synergy between GenAI-based data augmentation and DRL-based decision making provides a scalable, adaptive, and generalizable solution to UAV security. The proposed framework equips autonomous aerial systems with the ability to defend against both real and synthetic threats, while intelligently managing mission objectives and energy constraints in dynamic adversarial environments.

Future work will focus on deploying the system to real-world UAV platforms, integrating online adaptation mechanisms, and refining the reward shaping strategy for evolving multi-modal threats. These advancements aim to establish a foundation for next-generation UAV defense architectures built upon self-learning agents and adversarially enriched training regimes.

Several future directions are envisioned to further strengthen and expand the capabilities of the proposed framework. First, integrating a closed-loop architecture will be essential, where the detection outcomes from the IDS dynamically influence the retraining of generative models. This feedback-driven design will enable the system to adapt in real-time to changing attack tactics and data patterns.

Second, more efforts will be directed toward optimizing the framework for resource-constrained environments. Techniques such as model compression, quantization, and knowledge distillation will be explored to ensure that both the IDS and DQN modules can be efficiently deployed on embedded UAV platforms without compromising performance.

Third, expanding the framework to support multi-agent reinforcement learning could significantly enhance its real-world applicability. Coordinated learning among multiple UAVs would allow for distributed threat detection and cooperative defense, increasing the system's resilience against large-scale or coordinated cyber-physical attacks.

Fourth, ethical considerations and compliance with data privacy regulations such as GDPR will continue to guide the use of synthetic data. The proposed framework reduces dependence on sensitive real-world traffic, making it a privacy-preserving solution for IDS training and evaluation.

Fifth, future studies will prioritize extensive field testing on live UAVs under realistic environmental and operational conditions. This will help validate the robustness, adaptability, and reliability of the framework in real-world applications.

Finally, the framework will be extended to simulate and defend against zero-day and rare attacks by incorporating adversarial training strategies and generative interpolation techniques. These enhancements will prepare IDS models for high-impact but low-frequency threats that often go undetected in conventional systems.

In conclusion, this thesis demonstrates that combining GenAI with DRL can result in a powerful, adaptive, and intelligent IDS framework for UAV systems. The proposed approach not only improves attack detection under limited data conditions but also lays a foundation for next-generation autonomous defense systems that are resilient, efficient, and ethically aligned with real-world deployment needs.

REFERENCES

[1] **Mekdad, Y.**, **Aris, A.**, **Babun, L.**, **Fergougui, A.E.**, **Conti, M.**, **Lazzeretti, R. and Uluagac, A.S.** (2023). A survey on security and privacy issues of UAVs, *Computer Networks*, *224*, 109626.

[2] **Patel, T.**, **Salot, N. and Parikh, V.** (2022). *A systematic literature review on Security of Unmanned Aerial Vehicle Systems*, `https://arxiv.org/abs/2212.05028`, `2212.05028`.

[3] **Hadi, H.J.**, **Cao, Y.**, **Nisa, K.U.**, **Jamil, A.M. and Ni, Q.** (2023). A comprehensive survey on security, privacy issues and emerging defence technologies for UAVs, *Journal of Network and Computer Applications*, *213*, 103607, `https://www.sciencedirect.com/science/article/pii/S1084804523000267`.

[4] **Kumar, N. and Chaudhary, A.** (2024). Surveying cybersecurity vulnerabilities and countermeasures for enhancing UAV security, *Computer Networks*, *252*, 110695.

[5] **Rodday, N.M.**, **Schmidt, R.d.O. and Pras, A.** (2016). Exploring security vulnerabilities of unmanned aerial vehicles, *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, pp.993–994.

[6] **Krishna, C.G.L. and Murphy, R.R.** (2017). A review on cybersecurity vulnerabilities for unmanned aerial vehicles, *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, pp.194–199.

[7] **Hassija, V.**, **Chamola, V.**, **Agrawal, A.**, **Goyal, A.**, **Luong, N.C.**, **Niyato, D.**, **Yu, F.R. and Guizani, M.** (2021). Fast, reliable, and secure drone communication: a comprehensive survey, *IEEE Communications Surveys amp; Tutorials*, *23*(4), 2802–2832, `http://dx.doi.org/10.1109/COMST.2021.3097916`.

[8] **Sharma, J. and Mehra, P.S.** (2023). Secure communication in IOT-based UAV networks: A systematic survey, *Internet Things*, *23*, 100883, `https://api.semanticscholar.org/CorpusID:260101024`.

[9] **Hassija, V.**, **Saxena, V. and Chamola, V.** (2021). A mobile data offloading framework based on a combination of blockchain and virtual voting, *Software: Practice and Experience*, *51*(12), 2428–2445.

[10] **Hafeez, S.**, **Khan, A.R.**, **Al-Quraan, M.M.**, **Mohjazi, L.**, **Zoha, A.**, **Imran, M.A. and Sun, Y.** (2023). Blockchain-assisted UAV communication systems: a comprehensive survey, *IEEE Open Journal of Vehicular Technology*, *4*, 558–580.

[11] **García-Magariño, I.**, **Lacuesta, R.**, **Rajarajan, M. and Lloret, J.** (2019). Security in networks of unmanned aerial vehicles for surveillance with an agent-based approach inspired by the principles of blockchain, *Ad Hoc Networks*, *86*, 72–82, https://www.sciencedirect.com/science/article/pii/S1570870518301689.

[12] **Babiker Mohamed, M.**, **Matthew Alofe, O.**, **Ajmal Azad, M.**, **Singh Lallie, H.**, **Fatema, K. and Sharif, T.** (2022). A comprehensive survey on secure software-defined network for the Internet of Things, *Transactions on Emerging Telecommunications Technologies*, *33*(1).

[13] **Gupta, A. and Gupta, S.K.** (2023). A study on secured unmanned aerial vehicle-based fog computing networks, *SAE International Journal of Connected and Automated Vehicles*, *7*(12-07-02-0011).

[14] **Mpitziopoulos, A.**, **Gavalas, D.**, **Konstantopoulos, C. and Pantziou, G.** (2009). A survey on jamming attacks and countermeasures in WSNs, *IEEE Communications Surveys Tutorials*, *11*(4), 42–56.

[15] **Su, J.**, **He, J.**, **Cheng, P. and Chen, J.** (2016). A stealthy gps spoofing strategy for manipulating the trajectory of an unmanned aerial vehicle, *IFAC-PapersOnLine*, *49*(22), 291–296.

[16] **Kerns, A.J.**, **Shepard, D.P.**, **Bhatti, J.A. and Humphreys, T.E.** (2014). Unmanned Aircraft Capture and Control Via GPS Spoofing, *J. Field Robot.*, *31*(4), 617–636, https://doi.org/10.1002/rob.21513.

[17] **Whelan, J.**, **Sangarapillai, T.**, **Minawi, O.**, **Almehmadi, A. and El-Khatib, K.** (2020). Novelty-based Intrusion Detection of Sensor Attacks on Unmanned Aerial Vehicles, *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, Q2SWinet '20, Association for Computing Machinery, New York, NY, USA, p.23–28, https://doi.org/10.1145/3416013.3426446.

[18] **Sutton, R.S.**, **Barto, A.G.** *et al.* (1998). *Introduction to reinforcement learning*, volume135, MIT press Cambridge, https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf.

[19] **Nandy, A. and Biswas, M.** (2017). *Reinforcement Learning: With Open AI, TensorFlow and Keras Using Python*, Apress, https://link.springer.com/book/10.1007/978-1-4842-3285-9.

[20] **Bai, Y.**, **Zhao, H.**, **Zhang, X.**, **Chang, Z.**, **Jäntti, R. and Yang, K.** (2023). Toward autonomous multi-UAV wireless network: a survey of reinforcement learning-based approaches, *Commun. Surveys Tuts.*, *25*(4), 3038–3067, https://doi.org/10.1109/COMST.2023.3323344.

[21] **Ahmed, F.**, **Mohanta, J.**, **Keshari, A. and Yadav, P.S.** (2022). Recent advances in unmanned aerial vehicles: a review, *Arabian Journal for Science and Engineering*, *47*(7), 7963–7984.

[22] **Tlili, F.**, **Fourati, L.C.**, **Ayed, S. and Ouni, B.** (2022). Investigation on vulnerabilities, threats and attacks prohibiting UAVs charging and depleting UAVs batteries: Assessments countermeasures, *Ad Hoc Networks*, *129*, 102805, https://www.sciencedirect.com/science/article/pii/S1570870522000208.

[23] **Altawy, R. and Youssef, A.M.** (2016). Security, privacy, and safety aspects of civilian drones: a survey, *ACM Trans. Cyber-Phys. Syst.*, *1*(2), https://doi.org/10.1145/3001836.

[24] **Mekdad, Y.**, **Aris, A.**, **Babun, L.**, **Fergougui, A.E.**, **Conti, M.**, **Lazzeretti, R. and Uluagac, A.S.** (2021). *A survey on security and privacy issues of UAVs*, https://arxiv.org/abs/2109.14442, 2109.14442.

[25] **He, D.**, **Chan, S. and Guizani, M.** (2017). Communication security of unmanned aerial vehicles, *IEEE Wireless Communications*, *24*(4), 134–139.

[26] **Hartenstein, H. and Laberteaux, L.** (2008). A tutorial survey on vehicular ad hoc networks, *IEEE Communications Magazine*, *46*(6), 164–171.

[27] **Challita, U.**, **Saad, W. and Bettstetter, C.** (2019). Interference management for cellular-connected UAVs: a deep reinforcement learning approach, *IEEE Transactions on Wireless Communications*, *18*(4), 2125–2140, http://dx.doi.org/10.1109/TWC.2019.2900035.

[28] **Zeng, Y.**, **Zhang, R. and Lim, T.J.** (2016). Wireless communications with unmanned aerial vehicles: opportunities and challenges, *IEEE Communications Magazine*, *54*(5), 36–42.

[29] **Tsao, K.Y.**, **Girdler, T. and Vassilakis, V.G.** (2022). A survey of cyber security threats and solutions for UAV communications and flying ad-hoc networks, *Ad Hoc Networks*, *133*, 102894, https://www.sciencedirect.com/science/article/pii/S1570870522000853.

[30] **Jawhar, I.**, **Mohamed, N.**, **Al-Jaroodi, J.**, **Agrawal, D.P. and Zhang, S.** (2017). Communication and networking of UAV-based systems: Classification and associated architectures, *Journal of Network and Computer Applications*, *84*, 93–108, https://www.sciencedirect.com/science/article/pii/S1084804517300814.

[31] **Gupta, L.**, **Jain, R. and Vaszkun, G.** (2016). Survey of Important Issues in UAV Communication Networks, *IEEE Communications Surveys Tutorials*, *18*(2), 1123–1152.

[32] **Krichen, L.**, **Fourati, M. and Fourati, L.C.** (2018). Communication Architecture for Unmanned Aerial Vehicle System, ***N. Montavont and G.Z. Papadopoulos****, editors, Ad-hoc, Mobile, and Wireless Networks*, Springer International Publishing, Cham, pp.213–225.

[33] **FIPS, P.** (2006). 200, Minimum Security Requirements for Federal Information and Information Systems, *Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD March*.

[34] **Javaid, A.Y.**, **Sun, W.**, **Devabhaktuni, V.K. and Alam, M.** (2012). Cyber security threat analysis and modeling of an unmanned aerial vehicle system, *2012 IEEE Conference on Technologies for Homeland Security (HST)*, pp.585–590.

[35] **Behzadan, V.** (2017). *Cyber-Physical Attacks on UAS Networks- Challenges and Open Research Problems*, `https://arxiv.org/abs/1702.01251`, `1702.01251`.

[36] **Peake, T.M.** (2005). Eavesdropping in communication networks, Cambridge University Press, p.13–37.

[37] **He, D.**, **Chan, S. and Guizani, M.** (2017). drone-assisted public safety networks: the security aspect, *IEEE Communications Magazine*, *55*(8), 218–223.

[38] **Mo, Y. and Sinopoli, B.** (2009). Secure control against replay attacks, *2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp.911–918.

[39] **Carrio, A.**, **Sampedro, C.**, **Rodriguez-Ramos, A. and Cervera, P.C.** (2017). A Review of Deep Learning Methods and Applications for Unmanned Aerial Vehicles, *J. Sensors*, *2017*, 3296874:1–3296874:13, `https://api.semanticscholar.org/CorpusID:2958701`.

[40] **Luong, N.C.**, **Hoang, D.T.**, **Gong, S.**, **Niyato, D.**, **Wang, P.**, **Liang, Y.C. and Kim, D.I.** (2019). Applications of deep reinforcement learning in communications and networking: a survey, *IEEE Communications Surveys Tutorials*, *21*(4), 3133–3174.

[41] **Maxa, J.A.**, **Mahmoud, M.S.B. and Larrieu, N.** (2017). Survey on UAANET Routing Protocols and Network Security Challenges, *Ad Hoc Sens. Wirel. Networks*, *37*, 231–320, `https://api.semanticscholar.org/CorpusID:11587162`.

[42] **Chriki, A.**, **Touati, H.**, **Snoussi, H. and Kamoun, F.** (2019). FANET: Communication, mobility models and security issues, *Computer Networks*, *163*, 106877, https://www.sciencedirect.com/science/article/pii/S1389128618309034.

[43] **Chen, C.L.**, **Deng, Y.Y.**, **Weng, W.**, **Chen, C.H.**, **Chiu, Y.J. and Wu, C.M.** (2020). A Traceable and Privacy-Preserving Authentication for UAV Communication Control System, *Electronics*, *9*(1), https://www.mdpi.com/2079-9292/9/1/62.

[44] **Cervesato, I.** (2001). The Dolev-Yao intruder is the most powerful attacker, *16th Annual Symposium on Logic in Computer Science—LICS*, volume 1, Citeseer.

[45] **Jagannath, J.**, **Polosky, N.**, **Jagannath, A.**, **Restuccia, F. and Melodia, T.** (2019). Machine learning for wireless communications in the Internet of Things: A comprehensive survey, *Ad Hoc Networks*, *93*, 101913, https://www.sciencedirect.com/science/article/pii/S1570870519300812.

[46] **Yi, P.**, **Dai, Z.**, **Zhang, S.**, **Zhong, Y.** *et al.* (2005). A new routing attack in mobile ad hoc networks, *International Journal of Information Technology*, *11*(2), 83–94.

[47] **Kavitha, T. and Sridharan, D.** (2010). Security vulnerabilities in wireless sensor networks: A survey, *Journal of information Assurance and Security*, *5*(1), 31–44.

[48] **Rovira-Sugranes, A.**, **Razi, A.**, **Afghah, F. and Chakareski, J.** (2022). A review of AI-enabled routing protocols for UAV networks: Trends, challenges, and future outlook, *Ad Hoc Networks*, *130*, 102790, https://www.sciencedirect.com/science/article/pii/S1570870522000087.

[49] **Nawaz, H.**, **Ali, H.M. and Laghari, A.A.** (2020). UAV communication networks issues: a review, *Archives of Computational Methods in Engineering*, 1–21.

[50] **Westhoff, D.**, **Lamparter, B.**, **Paar, C. and Weimerskirch, A.** (2005). On digital signatures in ad hoc networks, *European transactions on telecommunications*, *16*(5), 411–425.

[51] **Saifullah**, **Ren, Z.**, **Hussain, K. and Faheem, M.** (2024). K-means online-learning routing protocol (K-MORP) for unmanned aerial vehicles (UAV) adhoc networks, *Ad Hoc Networks*, *154*, 103354, https://www.sciencedirect.com/science/article/pii/S1570870523002743.

[52] **Zear, A. and Ranga, V.** (2022). UAVs assisted Network Partition Detection and Connectivity Restoration in Wireless Sensor and Actor Networks, *Ad Hoc Networks*, *130*, 102823, https://www.sciencedirect.com/science/article/pii/S1570870522000336.

[53] **Rovira-Sugranes, A.**, **Razi, A.**, **Afghah, F. and Chakareski, J.** (2022). A review of AI-enabled routing protocols for UAV networks: Trends, challenges, and future outlook, *Ad Hoc Networks*, *130*, 102790, https://www.sciencedirect.com/science/article/pii/S1570870522000087.

[54] **Xue, J.**, **Wu, Q. and Zhang, H.** (2022). Cost optimization of UAV-MEC network calculation offloading: A multi-agent reinforcement learning method, *Ad Hoc Networks*, *136*, 102981, https://www.sciencedirect.com/science/article/pii/S1570870522001548.

[55] **Kermani, M.M.**, **Zhang, M.**, **Raghunathan, A. and Jha, N.K.** (2013). Emerging Frontiers in Embedded Security, *2013 26th International Conference on VLSI Design and 2013 12th International Conference on Embedded Systems*, pp.203–208.

[56] **Nia, A.M.**, **Mozaffari-Kermani, M.**, **Sur-Kolay, S.**, **Raghunathan, A. and Jha, N.K.** (2015). Energy-Efficient Long-term Continuous Personal Health Monitoring, *IEEE Transactions on Multi-Scale Computing Systems*, *1*(2), 85–98.

[57] **Yerlikaya, F.A. and Şerif Bahtiyar** (2022). Data poisoning attacks against machine learning algorithms, *Expert Systems with Applications*, *208*, 118101, https://www.sciencedirect.com/science/article/pii/S0957417422012933.

[58] **Mozaffari-Kermani, M.**, **Sur-Kolay, S.**, **Raghunathan, A. and Jha, N.K.** (2015). Systematic Poisoning Attacks on and Defenses for Machine Learning in Healthcare, *IEEE Journal of Biomedical and Health Informatics*, *19*(6), 1893–1905.

[59] **Mansfield, K.**, **Eveleigh, T.**, **Holzer, T.H. and Sarkani, S.** (2013). Unmanned aerial vehicle smart device ground control station cyber security threat model, *2013 IEEE International Conference on Technologies for Homeland Security (HST)*, pp.722–728.

[60] **Boccadoro, P.**, **Striccoli, D. and Grieco, L.A.** (2021). An extensive survey on the Internet of Drones, *Ad Hoc Networks*, *122*, 102600, https://www.sciencedirect.com/science/article/pii/S1570870521001335.

[61] **Su, J.**, **He, J.**, **Cheng, P. and Chen, J.** (2016). A Stealthy GPS Spoofing Strategy for Manipulating the Trajectory of an Unmanned Aerial Vehicle, *IFAC-PapersOnLine*, *49*(22), 291–296, https://www.sciencedirect.com/science/article/pii/S2405896316319991, 6th IFAC Workshop on Distributed Estimation and Control in Networked Systems NECSYS 2016.

[62] **Xiao, L.**, **Xie, C.**, **Min, M. and Zhuang, W.** (2018). User-Centric View of Unmanned Aerial Vehicle Transmission Against Smart Attacks, *IEEE Transactions on Vehicular Technology*, *67*, 3420–3430, https://api.semanticscholar.org/CorpusID:4941779.

[63] **Shafique, A.**, **Mehmood, A. and Elhadef, M.** (2021). Survey of Security Protocols and Vulnerabilities in Unmanned Aerial Vehicles, *IEEE Access*, *9*, 46927–46948.

[64] **Abbaspour, A.**, **Yen, K.K.**, **Noei, S. and Sargolzaei, A.** (2016). Detection of Fault Data Injection Attack on UAV Using Adaptive Neural Network, *Procedia Computer Science*, *95*, 193–200, https://www.sciencedirect.com/science/article/pii/S1877050916324851, complex Adaptive Systems Los Angeles, CA November 2-4, 2016.

[65] **Samy, I.**, **Postlethwaite, I. and Gu, D.W.** (2011). Survey and application of sensor fault detection and isolation schemes, *Control Engineering Practice*, *19*(7), 658–674, https://www.sciencedirect.com/science/article/pii/S0967066111000414.

[66] **Talebi, H. and Patel, R.** (2006). An intelligent Fault Detection and Recovery scheme for reaction wheel actuator of satellite attitude control systems, *2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control*, pp.3282–3287.

[67] **Shen, Q.**, **Jiang, B.**, **Shi, P. and Lim, C.C.** (2014). Novel Neural Networks-Based Fault Tolerant Control Scheme With Fault Alarm, *IEEE Transactions on Cybernetics*, *44*(11), 2190–2201.

[68] **Desnitsky, V.**, **Rudavin, N. and Kotenko, I.** (2020). Modeling and Evaluation of Battery Depletion Attacks on Unmanned Aerial Vehicles in Crisis Management Systems, ***I. Kotenko, C. Badica, V. Desnitsky, D. El Baz and M. Ivanovic**, editors, Intelligent Distributed Computing XIII*, Springer International Publishing, Cham, pp.323–332.

[69] **Yi, S.**, **Naldurg, P. and Kravets, R.** (2001). Security-aware ad hoc routing for wireless networks, *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, MobiHoc '01, Association for Computing Machinery, New York, NY, USA, p.299–302, https://doi.org/10.1145/501449.501464.

[70] **Harsch, C.**, **Festag, A. and Papadimitratos, P.** (2007). Secure Position-Based Routing for VANETs, *2007 IEEE 66th Vehicular Technology Conference*, pp.26–30.

[71] **Desilva, S. and Boppana, R.** (2005). Mitigating malicious control packet floods in ad hoc networks, *IEEE Wireless Communications and Networking Conference, 2005*, volume 4, pp.2112–2117 Vol. 4.

[72] **Sun, X.**, **Ng, D.W.K.**, **Ding, Z.**, **Xu, Y. and Zhong, Z.** (2019). Physical Layer Security in UAV Systems: Challenges and Opportunities, *IEEE Wireless Communications*, *26*(5), 40–47.

[73] (2012). Contributors, **S.K. Das**, **K. Kant and N. Zhang**, editors, Handbook on Securing Cyber-Physical Critical Infrastructure, Morgan Kaufmann, Boston, pp.xi–xv, `https://www.sciencedirect.com/science/article/pii/B9780124158153000340`.

[74] **Li, Y.**, **Frasure, I.**, **Ikusan, A.A.**, **Zhang, J. and Dai, R.** (2018). Vulnerability Assessment for Unmanned Systems Autonomy Services Architecture, ***M.H. Au, S.M. Yiu, J. Li, X. Luo, C. Wang, A. Castiglione and K. Kluczniak**, editors, Network and System Security*, Springer International Publishing, Cham, pp.266–276.

[75] **Li, Y.**, **Dai, R. and Zhang, J.** (2014). Morphing communications of Cyber-Physical Systems towards moving-target defense, *2014 IEEE International Conference on Communications (ICC)*, pp.592–598.

[76] **Anastasova, M.**, **Bisheh-Niasar, M.**, **Seo, H.**, **Azarderakhsh, R. and Kermani, M.M.** (2022). Efficient and Side-Channel Resistant Design of High-Security Ed448 on ARM Cortex-M4, *2022 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp.93–96.

[77] **Bisheh-Niasar, M.**, **Azarderakhsh, R. and Mozaffari-Kermani, M.** (2021). Cryptographic Accelerators for Digital Signature Based on Ed25519, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *29*(7), 1297–1305.

[78] **Anastasova, M.**, **Khatib, R.E.**, **Laclaustra, A.**, **Azarderakhsh, R. and Kermani, M.M.** (2023). Highly Optimized Curve448 and Ed448 design in wolfSSL and Side-Channel Evaluation on Cortex-M4, *2023 IEEE Conference on Dependable and Secure Computing (DSC)*, pp.1–8.

[79] **Sönmez Sarıkaya, B.** (2020). AES Algoritmasına Yapılan Zaman Odaklı Önbellek Saldırılarının Makine Öğrenmesi ile Tespiti, *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, *13*(1), 57–68.

[80] **Sönmez, B.**, **Sarıkaya, A.A. and Bahtiyar,** (2019). Machine Learning based Side Channel Selection for Time-Driven Cache Attacks on AES, *2019 4th International Conference on Computer Science and Engineering (UBMK)*, pp.1–5.

[81] **Thing, V.L. and Wu, J.** (2016). Autonomous Vehicle Security: A Taxonomy of Attacks and Defences, *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp.164–170.

[82] **Zhu, M. and Martínez, S.** (2014). On the Performance Analysis of Resilient Networked Control Systems Under Replay Attacks, *IEEE Transactions on Automatic Control*, *59*(3), 804–808.

[83] **Chen, B.**, **Ho, D.W.C.**, **Hu, G. and Yu, L.** (2018). Secure Fusion Estimation for Bandwidth Constrained Cyber-Physical Systems Under Replay Attacks, *IEEE Transactions on Cybernetics*, *48*(6), 1862–1876.

[84] **Miao, F.**, **Pajic, M. and Pappas, G.J.** (2013). Stochastic game approach for replay attack detection, *52nd IEEE Conference on Decision and Control*, pp.1854–1859.

[85] **Mpitziopoulos, A.**, **Gavalas, D.**, **Konstantopoulos, C. and Pantziou, G.** (2009). A survey on jamming attacks and countermeasures in WSNs, *IEEE Communications Surveys Tutorials*, *11*(4), 42–56.

[86] **Chamola, V.**, **Kotesh, P.**, **Agarwal, A.**, **Naren**, **Gupta, N. and Guizani, M.** (2021). A Comprehensive Review of Unmanned Aerial Vehicle Attacks and Neutralization Techniques, *Ad Hoc Networks*, *111*, 102324, https://www.sciencedirect.com/science/article/pii/S1570870520306788.

[87] **Guan, Y. and Ge, X.** (2018). Distributed Attack Detection and Secure Estimation of Networked Cyber-Physical Systems Against False Data Injection Attacks and Jamming Attacks, *IEEE Transactions on Signal and Information Processing over Networks*, *4*(1), 48–59.

[88] **Osanaiye, O.**, **Alfa, A.S. and Hancke, G.P.** (2018). A Statistical Approach to Detect Jamming Attacks in Wireless Sensor Networks, *Sensors*, *18*(6), https://www.mdpi.com/1424-8220/18/6/1691.

[89] **Humphreys, T.E.** (2012). STATEMENT ON THE VULNERABILITY OF CIVIL UNMANNED AERIAL VEHICLES AND OTHER SYSTEMS TO CIVIL GPS, https://api.semanticscholar.org/CorpusID:35123487.

[90] **He, L.**, **Li, W.**, **Guo, C. and Niu, R.** (2014). Civilian Unmanned Aerial Vehicle Vulnerability to GPS Spoofing Attacks, *2014 Seventh International Symposium on Computational Intelligence and Design*, volume 2, pp.212–215.

[91] **Hu, H. and Wei, N.** (2009). A study of GPS jamming and anti-jamming, *2009 2nd International Conference on Power Electronics and Intelligent Transportation System (PEITS)*, volume 1, pp.388–391.

[92] **Seo, S.H.**, **Lee, B.H.**, **Im, S.H. and Jee, G.I.** (2015). Effect of spoofing on unmanned aerial vehicle using counterfeited gps signal, *Journal of Positioning, Navigation, and Timing*, *4*(2), 57–65.

[93] **Psiaki, M.L.**, **O'Hanlon, B.W.**, **Bhatti, J.A.**, **Shepard, D.P. and Humphreys, T.E.** (2013). GPS Spoofing Detection via Dual-Receiver Correlation of Military Signals, *IEEE Transactions on Aerospace and Electronic Systems*, *49*(4), 2250–2267.

[94] **Zou, Q.**, **Huang, S.**, **Lin, F. and Cong, M.** (2016). Detection of GPS spoofing based on UAV model estimation, *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pp.6097–6102.

[95] **Pu, C. and Li, Y.** (2020). Lightweight Authentication Protocol for Unmanned Aerial Vehicles Using Physical Unclonable Function and Chaotic System, *2020 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN*, pp.1–6.

[96] **Bellare, M. and Namprempre, C.** (2000). Authenticated encryption: Relations among notions and analysis of the generic composition paradigm, *Advances in Cryptology—ASIACRYPT 2000: 6th International Conference on the Theory and Application of Cryptology and Information Security Kyoto, Japan, December 3–7, 2000 Proceedings 6*, Springer, pp.531–545.

[97] **He, S.**, **Wu, Q.**, **Liu, J.**, **Hu, W.**, **Qin, B. and Li, Y.N.** (2017). Secure communications in unmanned aerial vehicle network, *Information Security Practice and Experience: 13th International Conference, ISPEC 2017, Melbourne, VIC, Australia, December 13–15, 2017, Proceedings 13*, Springer, pp.601–620.

[98] **Putranto, D.S.C.**, **Aji, A.K. and Wahyudono, B.** (2019). Design and Implementation of Secure Transmission on Internet of Drones, *2019 IEEE 6th Asian Conference on Defence Technology (ACDT)*, pp.128–135.

[99] **Wesson, K.D.**, **Humphreys, T.E. and Evans, B.L.** (2014). Can Cryptography Secure Next Generation Air Traffic Surveillance?, `https://api.semanticscholar.org/CorpusID:21207906`.

[100] **Wei-jun**, **Pan**, **Ziliang**, **Feng**, **Yang and Wang** (2012). ADS-B Data Authentication Based on ECC and X.509 Certificate, `https://api.semanticscholar.org/CorpusID:16106662`.

[101] **Teng, L.**, **Jianfeng, M.**, **Pengbin, F.**, **Yue, M.**, **Xindi, M.**, **Jiawei, Z.**, **Gao, C. and Di, L.** (2019). Lightweight Security Authentication Mechanism Towards UAV Networks, *2019 International Conference on Networking and Network Applications (NaNA)*, pp.379–384.

[102] **Srinivas, J.**, **Das, A.K.**, **Kumar, N. and Rodrigues, J.J.P.C.** (2019). TCALAS: Temporal Credential-Based Anonymous Lightweight Authentication Scheme for Internet of Drones Environment, *IEEE Transactions on Vehicular Technology*, *68*(7), 6903–6916.

[103] **Ali, Z.**, **Chaudhry, S.A.**, **Ramzan, M.S. and Al-Turjman, F.** (2020). Securing Smart City Surveillance: A Lightweight Authentication Mechanism for Unmanned Vehicles, *IEEE Access*, *8*, 43711–43724.

[104] **Koziel, B.**, **Azarderakhsh, R.**, **Mozaffari Kermani, M. and Jao, D.** (2017). Post-Quantum Cryptography on FPGA Based on Isogenies on Elliptic Curves, *IEEE Transactions on Circuits and Systems I: Regular Papers*, *64*(1), 86–99.

[105] **Bisheh-Niasar, M.**, **Azarderakhsh, R. and Mozaffari-Kermani, M.** (2021). High-Speed NTT-based Polynomial Multiplication Accelerator for Post-Quantum Cryptography, *2021 IEEE 28th Symposium on Computer Arithmetic (ARITH)*, pp.94–101.

[106] **Anastasova, M.**, **Azarderakhsh, R. and Kermani, M.M.** (2021). Fast Strategies for the Implementation of SIKE Round 3 on ARM Cortex-M4, *IEEE Transactions on Circuits and Systems I: Regular Papers*, *68*(10), 4129–4141.

[107] **Yang, Z.**, **Alfauri, H.**, **Farkiani, B.**, **Jain, R.**, **Pietro, R.D. and Erbad, A.** (2024). A Survey and Comparison of Post-Quantum and Quantum Blockchains, *IEEE Communications Surveys Tutorials*, *26*(2), 967–1002.

[108] **Choudhary, G.**, **Sharma, V.**, **You, I.**, **Yim, K.**, **Chen, I.R. and Cho, J.H.** (2018). Intrusion Detection Systems for Networked Unmanned Aerial Vehicles: A Survey, *2018 14th International Wireless Communications Mobile Computing Conference (IWCMC)*, pp.560–565.

[109] **Xu, J.**, **Deng, Z.**, **Song, Q.**, **Chi, Q.**, **Wu, T.**, **Huang, Y.**, **Liu, D. and Gao, M.** (2020). Multi-UAV counter-game model based on uncertain information, *Applied Mathematics and Computation*, *366*, 124684, https://www.sciencedirect.com/science/article/pii/S0096300319306769.

[110] **Condomines, J.P.**, **Zhang, R. and Larrieu, N.** (2019). Network intrusion detection system for UAV ad-hoc communication: From methodology design to real test validation, *Ad Hoc Networks*, *90*, 101759, https://www.sciencedirect.com/science/article/pii/S1570870518306541, recent advances on security and privacy in Intelligent Transportation Systems.

[111] **Whelan, J.**, **Sangarapillai, T.**, **Minawi, O.**, **Almehmadi, A. and El-Khatib, K.** (2020). Novelty-based Intrusion Detection of Sensor Attacks on Unmanned Aerial Vehicles, *Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, Q2SWinet '20, Association for Computing Machinery, New York, NY, USA, p.23–28, https://doi.org/10.1145/3416013.3426446.

[112] **Hoang, T.M.**, **Nguyen, N.M. and Duong, T.Q.** (2020). Detection of Eavesdropping Attack in UAV-Aided Wireless Systems: Unsupervised Learning With One-Class SVM and K-Means Clustering, *IEEE Wireless Communications Letters*, *9*(2), 139–142.

[113] **Bozkurt, A.K.**, **Sarıkaya, B.S.**, **Aköz, H.E.**, **Taşpınar, A. and Bahtiyar,** (2024). A New Method to Detect Malicious DNS over HTTPS via Feature Reduction, *2024 9th International Conference on Computer Science and Engineering (UBMK)*, pp.754–759.

[114] **Arthur, M.P.** (2019). Detecting Signal Spoofing and Jamming Attacks in UAV Networks using a Lightweight IDS, *2019 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pp.1–5.

[115] **Bejiga, M.B.**, **Zeggada, A.**, **Nouffidj, A. and Melgani, F.** (2017). A Convolutional Neural Network Approach for Assisting Avalanche Search and Rescue Operations with UAV Imagery, *Remote Sensing*, *9*(2), https://www.mdpi.com/2072-4292/9/2/100.

[116] **Yang, B.**, **Matson, E.T.**, **Smith, A.H.**, **Dietz, J.E. and Gallagher, J.C.** (2019). UAV Detection System with Multiple Acoustic Nodes Using Machine Learning Models, *2019 Third IEEE International Conference on Robotic Computing (IRC)*, pp.493–498.

[117] **Yazdinejad, A.**, **Parizi, R.M.**, **Dehghantanha, A. and Karimipour, H.** (2021). Federated learning for drone authentication, *Ad Hoc Networks*, *120*, 102574, https://www.sciencedirect.com/science/article/pii/S1570870521001165.

[118] **Pu, C. and Zhu, P.** (2021). Defending against Flooding Attacks in the Internet of Drones Environment, *2021 IEEE Global Communications Conference (GLOBECOM)*, pp.1–6.

[119] **Svaigen, A.R.**, **Boukerche, A.**, **Ruiz, L.B. and Loureiro, A.A.** (2023). Trajectory Matters: Impact of Jamming Attacks Over the Drone Path Planning on the Internet of Drones, *Ad Hoc Networks*, *146*, 103179, https://www.sciencedirect.com/science/article/pii/S1570870523000999.

[120] **Condomines, J.P.**, **Zhang, R. and Larrieu, N.** (2019). Network intrusion detection system for UAV ad-hoc communication: From methodology design to real test validation, *Ad Hoc Networks*, *90*, 101759, https://www.sciencedirect.com/science/article/pii/S1570870518306541, recent advances on security and privacy in Intelligent Transportation Systems.

[121] **Do Yoo, J.**, **Kim, H. and Kim, H.K.** (2024). GUIDE: GAN-based UAV IDS Enhancement, *Computers & Security*, *147*, 104073.

[122] **Shahriar, M.H.**, **Haque, N.I.**, **Rahman, M.A. and Alonso, M.** (2020). G-ids: Generative adversarial networks assisted intrusion detection system, *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, IEEE, pp.376–385.

[123] **Lin, Z.**, **Shi, Y. and Xue, Z.** (2022). IDSGAN: Generative Adversarial Networks for Attack Generation Against Intrusion Detection, ***J. Gama***, ***T. Li***, ***Y. Yu***, ***E. Chen***, ***Y. Zheng and F. Teng***, *editors, Advances in Knowledge Discovery and Data Mining*, Springer International Publishing, Cham, pp.79–91.

[124] **Seo, E.**, **Song, H.M. and Kim, H.K.** (2018). GIDS: GAN based Intrusion Detection System for In-Vehicle Network, *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, pp.1–6.

[125] **Sarikaya, B.S. and Bahtiyar,** (2024). Generative Adversarial Networks for Synthetic Jamming Attacks on UAVs, *2024 9th International Conference on Computer Science and Engineering (UBMK)*, pp.760–765.

[126] **Sarıkaya, B.S. and Bahtiyar,** (2025). A Novel Framework for Adversarial DoS Attack Generation on UAVs, *2025 12th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp.272–279.

[127] **Jeong, S.**, **Park, E.**, **Seo, K.U.**, **Do Yoo, J. and Kim, H.K.** (2021). MUVIDS: false MAVLink injection attack detection in communication for unmanned vehicles, *Workshop on automotive and autonomous vehicle security (AutoSec)*, volume2021, p. 25.

[128] **Agrawal, G.**, **Kaur, A. and Myneni, S.** (2024). A review of generative models in generating synthetic attack data for cybersecurity, *Electronics*, *13*(2), 322.

[129] **Wang, Y.**, **Ding, J.**, **He, X.**, **Wei, Q.**, **Yuan, S. and Zhang, J.** (2024). Intrusion detection method based on denoising diffusion probabilistic models for uav networks, *Mobile Networks and Applications*, *29*(5), 1467–1476.

[130] **Yang, Y.**, **Tang, X.**, **Liu, Z.**, **Cheng, J.**, **Fang, H. and Zhang, C.** (2025). Diff-IDS: A Network Intrusion Detection Model Based on Diffusion Model for Imbalanced Data Samples., *Computers, Materials & Continua*, *82*(3).

[131] **Ammara, D.A.**, **Ding, J. and Tutschku, K.** (2024). Synthetic Data Generation in Cybersecurity: A Comparative Analysis, *arXiv preprint arXiv:2410.16326*.

[132] **Chalé, M. and Bastian, N.D.** (2022). Generating realistic cyber data for training and evaluating machine learning classifiers for network intrusion detection systems, *Expert Systems with Applications*, *207*, 117936.

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

[134] **Chen, Y.**, **Li, Y.**, **Xu, D. and Xiao, L.** (2018). DQN-Based Power Control for IoT Transmission against Jamming, *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, pp.1–5.

[135] **Xiao, L.**, **Lu, X.**, **Xu, D.**, **Tang, Y.**, **Wang, L. and Zhuang, W.** (2018). UAV Relay in VANETs Against Smart Jamming With Reinforcement Learning, *IEEE Transactions on Vehicular Technology*, *67*(5), 4087–4097.

[136] **Kimura, H.**, **Yamamura, M. and Kobayashi, S.**, (1995). Reinforcement Learning by Stochastic Hill Climbing on Discounted Reward, **A. Prieditis and S. Russell**, editors, Machine Learning Proceedings 1995, Morgan Kaufmann, San Francisco (CA), pp.295–303, https://www.sciencedirect.com/science/article/pii/B978155860377650044X.

[137] **Lu, X.**, **Xu, D.**, **Xiao, L.**, **Wang, L. and Zhuang, W.** (2017). Anti-Jamming Communication Game for UAV-Aided VANETs, *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pp.1–6.

[138] **Pan, S.J. and Yang, Q.** (2010). A Survey on Transfer Learning, *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.

[139] **Zhang, T. and Zhu, Q.** (2017). Strategic Defense Against Deceptive Civilian GPS Spoofing of Unmanned Aerial Vehicles, ***S. Rass***, ***B. An***, ***C. Kiekintveld***, ***F. Fang and S. Schauer***, *editors, Decision and Game Theory for Security*, Springer International Publishing, Cham, pp.213–233.

[140] **Sedjelmaci, H.**, **Boudguiga, A.**, **Jemaa, I.B. and Senouci, S.M.** (2019). An efficient cyber defense framework for UAV-Edge computing network, *Ad Hoc Networks*, *94*, 101970, https://www.sciencedirect.com/science/article/pii/S1570870519302136.

[141] **Xiao, L.**, **Li, Y.**, **Liu, G.**, **Li, Q. and Zhuang, W.** (2015). Spoofing Detection with Reinforcement Learning in Wireless Networks, *2015 IEEE Global Communications Conference (GLOBECOM)*, pp.1–5.

[142] **Praveena, V.**, **Vijayaraj, A.**, **Chinnasamy, P.**, **Ali, I.**, **Alroobaea, R.**, **Alyahyan, S.Y. and Raza, M.A.** (2022). Optimal Deep Reinforcement Learning for Intrusion Detection in UAVs, *Computers, Materials & Continua*, *70*(2), 2639–2653, http://www.techscience.com/cmc/v70n2/44676.

[143] **Hu, S.**, **Yuan, X.**, **Ni, W.**, **Wang, X. and Jamalipour, A.** (2023). *RIS-Assisted Jamming Rejection and Path Planning for UAV-Borne IoT Platform: A New Deep Reinforcement Learning Framework*, https://arxiv.org/abs/2302.04994, 2302.04994.

[144] **Wang, X.**, **Cenk Gursoy, M.**, **Erpek, T. and Sagduyu, Y.E.** (2021). Jamming-Resilient Path Planning for Multiple UAVs via Deep Reinforcement Learning, *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp.1–6.

[145] **Qian, Z.**, **Deng, Z.**, **Cai, C. and Li, H.** (2023). Reinforcement Learning Based Dual-UAV Trajectory Optimization for Secure Communication, *Electronics*, *12*(9), https://www.mdpi.com/2079-9292/12/9/2008.

[146] **Yoo, S.**, **Jeong, S. and Kang, J.** (2023). Hybrid UAV-Enabled Secure Offloading via Deep Reinforcement Learning, *IEEE Wireless Communications Letters*, *12*(6), 972–976.

[147] **Tham, M.L.**, **Wong, Y.J.**, **Iqbal, A.**, **Ramli, N.B.**, **Zhu, Y. and Dagiuklas, T.** (2023). Deep Reinforcement Learning for Secrecy Energy- Efficient UAV Communication with Reconfigurable Intelligent Surface, *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, pp.1–6.

[148] **Yuan, X.**, **Hu, S.**, **Ni, W.**, **Wang, X. and Jamalipour, A.** (2023). Deep Reinforcement Learning-Driven Reconfigurable Intelligent Surface-Assisted Radio Surveillance With a Fixed-Wing UAV, *IEEE Transactions on Information Forensics and Security*, *18*, 4546–4560.

[149] **Bi, S.**, **Hu, L.**, **Liu, Q.**, **Wu, J.**, **Yang, R. and Wu, L.** (2023). Deep reinforcement learning for IRS-assisted UAV covert communications, *China Communications*, *20*(12), 131–141.

[150] **Tao, J.**, **Han, T. and Li, R.** (2021). Deep-Reinforcement-Learning-Based Intrusion Detection in Aerial Computing Networks, *IEEE Network*, *35*(4), 66–72.

[151] **Zhang, Y.**, **Mou, Z.**, **Gao, F.**, **Jiang, J.**, **Ding, R. and Han, Z.** (2020). UAV-Enabled Secure Communications by Multi-Agent Deep Reinforcement Learning, *IEEE Transactions on Vehicular Technology*, *69*(10), 11599–11611.

[152] **Louati, F.**, **Ktata, F.B. and Amous, I.** (2024). Enhancing Intrusion Detection Systems with Reinforcement Learning: A Comprehensive Survey of RL-based Approaches and Techniques, *SN Computer Science*, *5*(6), 665.

[153] **Lopez-Martin, M.**, **Carro, B. and Sanchez-Esguevillas, A.** (2020). Application of deep reinforcement learning to intrusion detection for supervised problems, *Expert Systems with Applications*, *141*, 112963, https://www.sciencedirect.com/science/article/pii/S0957417419306815.

[154] **Kolias, C.**, **Kambourakis, G.**, **Stavrou, A. and Gritzalis, S.** (2016). Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset, *IEEE Communications Surveys Tutorials*, *18*(1), 184–208.

[155] **Tavallaee, M.**, **Bagheri, E.**, **Lu, W. and Ghorbani, A.A.** (2009). A detailed analysis of the KDD CUP 99 data set, *2009 IEEE symposium on computational intelligence for security and defense applications*, Ieee, pp.1–6.

[156] **Hsu, Y.F. and Matsuoka, M.** (2020). A deep reinforcement learning approach for anomaly network intrusion detection system, *2020 IEEE 9th international conference on cloud networking (CloudNet)*, IEEE, pp.1–6.

[157] **Sethi, K.**, **Kumar, R.**, **Prajapati, N. and Bera, P.** (2020). Deep reinforcement learning based intrusion detection system for cloud infrastructure, *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, IEEE, pp.1–6.

[158] **Li, Z.**, **Lu, Y.**, **Li, X.**, **Wang, Z.**, **Qiao, W. and Liu, Y.** (2021). UAV Networks Against Multiple Maneuvering Smart Jamming With Knowledge-Based Reinforcement Learning, *IEEE Internet of Things Journal*, *8*(15), 12289–12310.

[159] **Wen, C.**, **Fang, Y. and Qiu, L.** (2022). Securing UAV Communication Based on Multi-Agent Deep Reinforcement Learning in the Presence of Smart UAV Eavesdropper, *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, pp.1164–1169.

[160] **Yang, J.**, **Liang, G.**, **Li, B.**, **Wen, G. and Gao, T.** (2021). A deep-learning-and reinforcement-learning-based system for encrypted network malicious traffic detection, *Electronics Letters*, *57*(9), 363–365.

[161] **Kim, C. and Park, J.** (2019). Designing online network intrusion detection using deep auto-encoder Q-learning, *Computers & Electrical Engineering*, *79*, 106460.

[162] **Liu, Z.**, **Yin, X. and Hu, Y.** (2020). CPSS LR-DDoS detection and defense in edge computing utilizing DCNN Q-learning, *IEEE Access*, *8*, 42120–42130.

[163] **Xia, S.**, **Qiu, M. and Jiang, H.** (2019). An adversarial reinforcement learning based system for cyber security, *2019 IEEE International Conference on Smart Cloud (SmartCloud)*, IEEE, pp.227–230.

[164] **Sethi, K.**, **Sai Rupesh, E.**, **Kumar, R.**, **Bera, P. and Venu Madhav, Y.** (2020). A context-aware robust intrusion detection system: a reinforcement learning-based approach, *International Journal of Information Security*, *19*, 657–678.

[165] **Sethi, K.**, **Madhav, Y.V.**, **Kumar, R. and Bera, P.** (2021). Attention based multi-agent intrusion detection systems using reinforcement learning, *Journal of Information Security and Applications*, *61*, 102923.

[166] **Wu, M.**, **Zhu, Z.**, **Xia, Y.**, **Yan, Z.**, **Zhu, X. and Ye, N.** (2023). A q-learning-based two-layer cooperative intrusion detection for internet of drones system, *Drones*, *7*(8), 502.

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

[168] **Ren, K.**, **Zeng, Y.**, **Zhong, Y.**, **Sheng, B. and Zhang, Y.** (2023). MAFSIDS: a reinforcement learning-based intrusion detection model for multi-agent feature selection networks, *Journal of Big Data*, *10*(1), 137.

[169] **Khayat, M.**, **Barka, E.**, **Serhani, M.A.**, **Sallabi, F.**, **Shuaib, K. and Khater, H.M.** (2025). Reinforcement Learning with Deep Features: A Dynamic Approach for Intrusion Detection in IOT Networks, *IEEE Access*.

[170] **Pasikhani, A.M.**, **Clark, J.A. and Gope, P.** (2022). Adversarial RL-Based IDS for Evolving Data Environment in 6LoWPAN, *IEEE Transactions on Information Forensics and Security*, *17*, 3831–3846.

[171] **AlKhonaini, A.**, **Sheltami, T.**, **Mahmoud, A. and Imam, M.** (2024). UAV Detection Using Reinforcement Learning, *Sensors*, *24*(6), `https://www.mdpi.com/1424-8220/24/6/1870`.

[172] **Arranz, R.**, **Carramiñana, D.**, **Miguel, G.d.**, **Besada, J.A. and Bernardos, A.M.** (2023). Application of Deep Reinforcement Learning to UAV Swarming for Ground Surveillance, *Sensors*, *23*(21), 8766, `http://dx.doi.org/10.3390/s23218766`.

[173] **Bouhamed, O.**, **Bouachir, O.**, **Aloqaily, M. and Al Ridhawi, I.** (2021). Lightweight ids for uav networks: A periodic deep reinforcement learning-based approach, *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, IEEE, pp.1032–1037.

[174] **Soliman, A.**, **Bahri, M.**, **Izham, D. and Mohamed, A.** (2022). Aireye: Uav-based intelligent drl mobile target visitation, *2022 International Wireless Communications and Mobile Computing (IWCMC)*, IEEE, pp.554–559.

[175] **Salameh, H.B.**, **Alhafnawi, M.**, **Masadeh, A. and Jararweh, Y.** (2023). Federated reinforcement learning approach for detecting uncertain deceptive target using autonomous dual UAV system, *Information Processing & Management*, *60*(2), 103149.

[176] **Strickland, C.**, **Zakar, M.**, **Saha, C.**, **Soltani Nejad, S.**, **Tasnim, N.**, **Lizotte, D.J. and Haque, A.** (2024). Drl-gan: A hybrid approach for binary and multiclass network intrusion detection, *Sensors*, *24*(9), 2746.

[177] **Masadeh, A.**, **Alhafnawi, M.**, **Salameh, H.A.B.**, **Musa, A. and Jararweh, Y.** (2022). Reinforcement learning-based security/safety uav system for intrusion detection under dynamic and uncertain target movement, *IEEE Transactions on Engineering Management*.

[178] **Whelan, J.**, **Almehmadi, A. and El-Khatib, K.** (2022). Artificial intelligence for intrusion detection systems in Unmanned Aerial Vehicles, *Computers and Electrical Engineering*, *99*, 107784, https://www.sciencedirect.com/science/article/pii/S0045790622000842.

[179] **Koubâa, A.**, **Allouch, A.**, **Alajlan, M.**, **Javed, Y.**, **Belghith, A. and Khalgui, M.** (2019). Micro Air Vehicle Link (MAVlink) in a Nutshell: A Survey, *IEEE Access*, *7*, 87658–87680.

[180] **Kwon, Y.M.**, **Yu, J.**, **Cho, B.M.**, **Eun, Y. and Park, K.J.** (2018). Empirical Analysis of MAVLink Protocol Vulnerability for Attacking Unmanned Aerial Vehicles, *IEEE Access*, *6*, 43203–43212.

[181] **Allouch, A.**, **Cheikhrouhou, O.**, **Koubâa, A.**, **Khalgui, M. and Abbes, T.** (2019). MAVSec: Securing the MAVLink Protocol for Ardupilot/PX4 Unmanned Aerial Systems, *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, 621–628, https://api.semanticscholar.org/CorpusID:141502583.

[182] **Patki, N.**, **Wedge, R. and Veeramachaneni, K.** (2016). The synthetic data vault, *2016 IEEE international conference on data science and advanced analytics (DSAA)*, IEEE, pp.399–410.

[183] **Xu, L.**, **Skoularidou, M.**, **Cuesta-Infante, A. and Veeramachaneni, K.** (2019). Modeling tabular data using conditional gan, *Advances in neural information processing systems*, *32*.

[184] **Borisov, V.**, **Leemann, T.**, **Seßler, K.**, **Haug, J.**, **Pawelczyk, M. and Kasneci, G.** (2024). Deep Neural Networks and Tabular Data: A Survey, *IEEE Transactions on Neural Networks and Learning Systems*, *35*(6), 7499–7519, http://dx.doi.org/10.1109/TNNLS.2022.3229161.

[185] **Ho, J.**, **Jain, A. and Abbeel, P.** (2020). Denoising Diffusion Probabilistic Models, *ArXiv*, *abs/2006.11239*, https://api.semanticscholar.org/CorpusID:219955663.

[186] **Sohl-Dickstein, J.N.**, **Weiss, E.A.**, **Maheswaranathan, N. and Ganguli, S.** (2015). Deep Unsupervised Learning using Nonequilibrium Thermodynamics, *ArXiv*, *abs/1503.03585*, https://api.semanticscholar.org/CorpusID:14888175.

[187] **Wan, Z.**, **Zhang, Y. and He, H.** (2017). Variational autoencoder based synthetic data generation for imbalanced learning, *2017 IEEE symposium series on computational intelligence (SSCI)*, IEEE, pp.1–7.

[188] **Figueira, A. and Vaz, B.** (2022). Survey on synthetic data generation, evaluation methods and GANs, *Mathematics*, *10*(15), 2733.

[189] **Razghandi, M.**, **Zhou, H.**, **Erol-Kantarci, M. and Turgut, D.** (2022). Variational autoencoder generative adversarial network for synthetic data generation in smart home, *ICC 2022-IEEE International Conference on Communications*, IEEE, pp.4781–4786.

[190] **Hollenbach, F.M.**, **Bojinov, I.**, **Minhas, S.**, **Metternich, N.W.**, **Ward, M.D. and Volfovsky, A.** (2021). Multiple imputation using Gaussian copulas, *Sociological Methods & Research*, *50*(3), 1259–1283.

[191] **Pathare, A.**, **Mangrulkar, R.**, **Suvarna, K.**, **Parekh, A.**, **Thakur, G. and Gawade, A.** (2023). Comparison of tabular synthetic data generation techniques using propensity and cluster log metric, *International Journal of Information Management Data Insights*, *3*(2), 100177.

[192] **Nichol, A. and Dhariwal, P.** (2021). Improved Denoising Diffusion Probabilistic Models, *ArXiv*, *abs/2102.09672*, `https://api.semanticscholar.org/CorpusID:231979499`.

[193] **Chen, T. and Guestrin, C.** (2016). Xgboost: A scalable tree boosting system, *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp.785–794.

[194] **Xu, Y.**, **Zhao, X.**, **Chen, Y. and Yang, Z.** (2019). Research on a Mixed Gas Classification Algorithm Based on Extreme Random Tree, *Applied Sciences*, *9*(9), `https://www.mdpi.com/2076-3417/9/9/1728`.

[195] **Kaelbling, L.P.**, **Littman, M.L. and Moore, A.W.** (1996). Reinforcement learning: a survey, *J. Artif. Int. Res.*, *4*(1), 237–285.

[196] **Kaelbling, L.P.**, **Littman, M.L. and Moore, A.W.**, (1995). An introduction to reinforcement learning, The Biology and Technology of Intelligent Autonomous Agents, Springer, pp.90–127.

[197] **Xu, L.**, **Skoularidou, M.**, **Cuesta-Infante, A. and Veeramachaneni, K.** (2019). Modeling Tabular data using Conditional GAN, *Advances in Neural Information Processing Systems*.

[198] **Patki, N.**, **Wedge, R. and Veeramachaneni, K.** (2016). The Synthetic Data Vault, *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp.399–410.

[199] **Xu, L.**, **Skoularidou, M.**, **Cuesta-Infante, A. and Veeramachaneni, K.** (2019). Modeling tabular data using conditional GAN, Curran Associates Inc., Red Hook, NY, USA.

[200] **Glorot, X. and Bengio, Y.** (2010). Understanding the difficulty of training deep feedforward neural networks, *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, JMLR Workshop and Conference Proceedings, pp.249–256.

[201] **Sarikaya, B.S. and Bahtiyar, Ş.** (2025). GenAI-Based Jamming and Spoofing Attacks on UAVs, *IEEE Access*.

[202] (2023). *EU AI Act: first regulation on artificial intelligence*, European Parliament Topics page, https://www.europarl.europa.eu/topics/en/article/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence.

[203] **Soler Garrido, J.**, **Tolan, S.**, **Hupont Torres, I.**, **Fernandez Llorca, D.**, **Charisi, V.**, **Gomez Gutierrez, E.**, **Junklewitz, H.**, **Hamon, R.**, **Fano Yela, D. and Panigutti, C.** (2023). AI Act: Landscaping of AI standards in support of trustworthy AI, **JRC Scientific and Technical Reports KJ-NA-31-343-EN-N (online)**, Publications Office of the European Union, Luxembourg, https://publications.jrc.ec.europa.eu/repository/handle/JRC131155.

[204] **The IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems** (2019). *Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems*, IEEE Standards Association, first edition edition, https://standards.ieee.org/content/ieee-standards/en/industry-connections/ec/autonomous-systems.html.

[205] (2023). *ISO/IEC 23894:2023 – Information technology — Artificial intelligence — Guidance on risk management*, https://www.iso.org/standard/77304.html, edition 1.

CURRICULUM VITAE

Name SURNAME: Burcu SÖNMEZ SARIKAYA

EDUCATION:

- **PhD:** 2025, Istanbul Technical University, Department of Computer Engineering
- **B.Sc.:** 2012, Fırat University, Department of Computer Engineering, Faculty of Computer Engineering,
- **M.Sc.:** 2018, Fırat University, Graduate School of Natural and Applied Sciences, Computer Engineering

PROFESSIONAL EXPERIENCE AND SCHOLARSHIP:

- 2024, Turkcell Scholarship Research Assistant, Istanbul Technical University. Turkey.
- 2020-2024, YÖK 100/2000 Doctoral Scholarship, Istanbul Technical University.
- 2018-2019, Research Assistant(ÖYP), Istanbul Technical University.
- 2014-2018, Research Assistant (ÖYP), Ağrı Ibrahim Cecen University.
- 2015-2016, Research Assistant (ÖYP), Fırat University

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Sarikaya, B. S.**, & Bahtiyar, Ş. (2025). GenAI-Based Jamming and Spoofing Attacks on UAVs *IEEE Access* (Q1).
- **Sarikaya, B. S.**, & Bahtiyar, Ş. (2025). A Novel Framework for Adversarial DoS Attack Generation on UAVs. *12th IFIP International Conference on New Technologies, Mobility and Security (NTMS)* (pp. 272 - 279). IEEE.
- **Sarikaya, B. S.**, & Bahtiyar, Ş. (2024). A survey on security of UAV and deep reinforcement learning. *Elsevier, Ad Hoc Networks*, 103642. (Q1).
- **Sarikaya, B. S.**, & Bahtiyar, Ş. (2024). Generative Adversarial Networks for Synthetic Jamming Attacks on UAVs. *In 2024 9th International Conference on Computer Science and Engineering (UBMK)* (pp. 760-765). IEEE.

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- Bozkurt, A. K., **Sarıkaya, B. S.**, Aköz, H. E., Taşpınar, A., & Bahtiyar, Ş. (2024). A New Method to Detect Malicious DNS over HTTPS via Feature Reduction. *In 2024 9th International Conference on Computer Science and Engineering (UBMK)* (pp. 754-759). IEEE.

- **Sarıkaya, B. S.**, & Bahtiyar, Ş. (2020). AES algoritmasına yapılan zaman odaklı önbellek saldırılarının makine öğrenmesi ile tespiti. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 13(1), 57-68.

- **Sönmez, B.**, Sarıkaya, A. A., & Bahtiyar, Ş. (2019). Machine Learning-Based Side Channel Selection for Time-Driven Cache Attacks on AES. *In 2019 4th International Conference on Computer Science and Engineering (UBMK)* (pp. 1-5). IEEE.