Optimizing Effectiveness and Defense of Drone Surveillance Missions via Honey Drones

ZELIN WAN, computer science, Virginia Tech, Falls Church, United States
JIN-HEE CHO, Virginia Tech, Falls Church, United States
MU ZHU, North Carolina State University at Raleigh, Raleigh, United States
AHMED ANWAR, US Army DEVCOM Army Research Laboratory, Adelphi, United States
CHARLES KAMHOUA, US Army DEVCOM Army Research Laboratory, Adelphi, United States
MUNINDAR SINGH, North Carolina State University, Raleigh, United States

This work aims to develop a surveillance mission system using unmanned aerial vehicles (UAVs) or drones when Denial-of-Service (DoS) attacks are present to disrupt normal operations for mission systems. In particular, we introduce the concept of *cyber deception* using honey drones (HDs) to protect the mission system from DoS attacks. HDs exhibit fake vulnerabilities and employ stronger signal strengths to lure DoS attacks, unlike the legitimate drones called *mission drones* (MDs) deployed for mission execution. This research formulates an optimization problem to identify an optimal set of signal strengths of HDs and MDs to best prevent the system from DoS attacks while maximizing mission performance under the resource constraints of UAVs. To solve this optimization problem, we leverage deep reinforcement learning (DRL) to achieve these multiple objectives of the mission system concerning system security and performance. Particularly, for efficient and effective parallel processing in DRL, we utilize a DRL algorithm called the *Asynchronous Advantage Actor-Critic* (A3C) algorithm to model attack-defense interactions. We employ a physical engine-based simulation testbed to consider realistic scenarios and demonstrate valid findings from the realistic testbed. The extensive experiments proved that our HD-based approach could achieve up to a 32% increase in mission completion, a 20% reduction in energy consumption, and a 62% decrease in attack success rates compared to existing defense strategies.

CCS Concepts: • **Security and privacy** → **Denial-of-service attacks**; **Mobile and wireless security**; • **Theory of computation** → *Sequential decision making*.

Additional Key Words and Phrases: Defensive deception, deep reinforcement learning, unmanned aerial vehicle, mission effectiveness

1 Introduction

Unmanned Aerial Vehicle (UAV)-based mission systems have been extensively studied in ensuring wireless communications [1, 2], quality data collection [3], optimal strategy selection for Quality-of-Service (QoS) [4], energy-efficient task completion [5], post-disaster mapping, managing and searching [6–8], accurate location

Authors' Contact Information: Zelin Wan, computer science, Virginia Tech, Falls Church, Virginia, United States; e-mail: zelin@vt.edu; Jin-Hee Cho, Virginia Tech, Falls Church, Virginia, United States; e-mail: jicho@vt.edu; Mu Zhu, North Carolina State University at Raleigh, Raleigh, North Carolina, United States; e-mail: mzhu5@ncsu.edu; Ahmed Anwar, US Army DEVCOM Army Research Laboratory, Adelphi, Maryland, United States; e-mail: a.h.anwar@knights.ucf.edu; Charles Kamhoua, US Army DEVCOM Army Research Laboratory, Adelphi, Maryland, United States; e-mail: charles.a.kamhoua.civ@mail.mil; Munindar Singh, North Carolina State University, Raleigh, North Carolina, United States; e-mail: mpsingh@ncsu.edu.

ACM 1557-6051/2024/10-ART
https://doi.org/10.1145/3701233

detection [9], and management and orchestration of 5G/B5G UAV systems [10]. However, defenses to mitigate Denial-of-Service (DoS) attacks [11–13], a serious threat, have been significantly less explored. Addressing this gap, we propose a novel cyber deception defense using the so-called *honey drone* (HD) for the UAV-based surveillance mission system. The HDs are deployed to deceive the DoS attacker and gather attack intelligence. This approach is first introduced to handle DoS attacks in UAV-based mission systems. Cyber deception (or defensive deception, DD) is used to mislead the attacker towards fake assets, such as honeypots, to protect the system. We consider an HD to serve as a mobile honeypot aimed at attracting DoS attacks while simultaneously collecting data on these attacks to update the security configuration of the system. Our proposed technique enhances security and performance in UAV-based surveillance missions by leveraging key drone features: mobility allows honey drones to dynamically position themselves for optimal coverage and threat mitigation; dynamic signal strength adjustment enables HDs to create convincing decoys that attract DoS attacks away from mission-critical drones; autonomous operation supports real-time adaptive defense strategies without constant human intervention; and distributed network formation through ad hoc networks like FANETs increases network resilience. While the approach is optimized for drones, it could also benefit other Wi-Fi-based wireless devices with similar mobility and flexibility characteristics.

Camouflage-based Defensive Deception (DD) techniques have been widely used to mislead attackers and protect critical assets, such as in enterprise networks through decoy systems like honeypots. However, their application in wireless networks, particularly for UAV-based mission systems, remains underexplored. To address this gap, our research introduces a novel approach by employing honey drones (HDs) to protect mission drones (MDs) from DoS attacks. Unlike traditional static honeypots, our method uses mobile HDs with dynamically adjustable signal strengths, optimized through deep reinforcement learning (DRL), to actively deceive attackers. HDs use stronger signal strengths than MDs to lure DoS attacks, while MDs maintain sufficient signal strengths to ensure successful mission execution. This work aims to identify optimal signal strength levels for both HDs and MDs to achieve effective security and performance. We leverage DRL to dynamically adjust the signal strength levels to enhance mission performance and system security against adversarial threats. Our approach addresses challenges unique to UAV networks, including mobility, energy constraints, and real-time decision-making, and demonstrates its effectiveness against intelligent adversaries. While previous efforts have optimized UAV-based mission systems in areas like improving data accuracy in cadastral mapping [3], avoiding communication congestion in drone fleets [4], enhancing post-disaster mapping [6], correcting coordinates in GPS-denied environments [9], and managing 5G/B5G communication systems [10], they primarily focus on specific applications without leveraging DRL for comprehensive system optimization. Building on our preliminary work [14], which introduced the concept of using HDs within UAV systems to create deceptive defenses against DoS attacks using game theory or DRL, this paper further enhances the framework by incorporating a physics engine-based simulation that accounts for actual drone behavior, including speed, physical interactions, and stabilization. This enables more accurate simulation outcomes and adopts the Asynchronous Advantage Actor-Critic (A3C) algorithm to improve DRL training efficiency and decision-making quality under uncertain conditions. Additionally, we broaden the evaluation criteria to five metrics and conduct sensitivity analyses to examine the impact of varying mission lengths and drone availability, providing deeper insights into the strategic effectiveness of our approach under diverse scenarios.

Specifically, we have the following **key research contributions**:

- **Honey drones for a UAV-based surveillance mission system**: This work presents the novel "Honey Drone" concept, a dynamic defense strategy within UAV surveillance systems utilizing mobile honeypots to safeguard against sophisticated DoS attacks. Unlike existing stationary honeypot systems based on Raspberry Pi for UAVs [15], HDs proactively lure and mislead attackers, providing a more effective and strategic layer of protection for MDs during aerial surveillance tasks.

- **DRL-based intelligent attack strategies**: In contrast to existing DRL applications for UAVs that focus on system performance and energy efficiency [16–19], our approach provides attack-defense interactions where both parties use DRL to make their best strategies. To our knowledge, most existing cybersecurity solutions use DRL only to develop intelligent defense strategies, not attack strategies. Despite some studies conducted on using DRL for UAV security [20–22], no research has utilized DRL to model intelligent attack behaviors for UAV-based mission systems.
- **Effective and efficient DRL for parallel processing using A3C**: The adoption of A3C [23] in this work facilitates efficient and simultaneous exploration of different states, significantly enhancing decision quality in environments under partial observability.
- **Evaluation using realistic physics engine-based testbed**: A physics engine-based simulated environment and parameters from the real drone are considered in our experimental testbed. This increases the validity of our experimental findings.
- **Validation of the DRL-based dynamic signal strength identification with HDs**: We demonstrate the superiority of our HD-based approach via extensive experiments with in-depth performance comparative analyses for evaluating mission performance, energy consumption, and attack success rates. Via our extensive experiments, our HD-based approach can achieve up to a 32% increase in mission completion, a 20% reduction in energy consumption, and a 62% decrease in attack success rates compared to existing defense strategies.

2 Related Work

2.1 DRL-based Approaches for UAVs

Existing UAV security research often uses stationary honeypots like Raspberry Pi devices, which lack mobility and are less effective in dynamic environments. Our approach introduces mobile drones as honeypots that dynamically adjust signal strength and position in real-time. By integrating deep reinforcement learning (DRL), our system continuously optimizes these parameters, offering a more robust and adaptive defense for UAVs in unpredictable settings.

Cetin et al. [16] used DRL for autonomous drone navigation in suburban environments, addressing obstacles, such as trees, cables, parked cars, and other drones. Utilizing a combination of front camera depth images and scalar data, including distance to the virtual barrier and angle to goal, the drones learned to navigate and avoid obstacles with high success rates. The research highlighted extensive training and testing scenarios, demonstrating the effectiveness of the DRL algorithm in managing complex UAV navigational challenges. Xie et al. [17] introduced a DRL-based three-dimensional UAV path planning in dynamic environments. It optimized UAV path planning in scenarios with limited local environmental information, addressing partial observability challenges and large state spaces. The proposed method combined current reward and state-action values for efficient action selection and used adaptive experience replay to improve learning efficiency and stability. Wang et al. [24] developed a learning-based game approach for collaborative honeypot defense in UAV networks, focusing on the optimal and fair incentive design for sharing trapped attack data among UAVs to enhance network security. Their framework utilizes game theory and contract-theoretic approaches under varying levels of information asymmetry, demonstrating through simulations that their scheme motivates UAV cooperation in defense effectiveness significantly better than conventional schemes.

Zhou et al. [19] used DRL to optimize the multi-target tracking (MTT) in large-scale UAV swarms. This work introduced a decentralized swarm-oriented Markov Decision Process and a cartogram feature representation to handle the challenges of scalability and variable information dimensions in UAV swarms. This method improved UAVs' cooperative tracking capabilities and scalability. In addition, some studies use DRL to enhance UAV security. While DRL has been used for UAV path planning and navigation, it has not addressed cyber threats like DoS

attacks. Our work fills this gap by focusing on DRL-based defenses against such attacks. Jing et al. [20] proposed a UAV-assisted mobile edge computing (MEC) system by leveraging DRL to handle eavesdropping attacks. The DRL algorithm optimized the UAV's flight strategy to maximize the average secrecy rate of communicating with users. Specifically, it addressed the challenge of secure communications by using DRL to adaptively change the UAV's position and counter eavesdropping threats. Zhang et al. [21, 22] proposed a multi-agent deep deterministic policy gradient to optimize the trajectory of UAVs for secure communications. UAVs were used as aerial base stations and jammers to counteract eavesdropping threats, improving the secure capacity in a 3D space by strategically adjusting UAV positions and power settings in response to ground eavesdroppers.

Limitations of the existing DRL-based UAVs: While DRL has been successfully employed in various UAV-related tasks such as path planning, obstacle avoidance, and multi-target tracking [16–19], its application in optimizing defense mechanisms against cyberattacks in UAVs remains underexplored. Existing DRL-based security solutions [20–22] have primarily focused on static or enterprise network environments, which differ significantly from the dynamic, resource-constrained settings typical of UAV operations. In the context of UAVs, these settings demand more lightweight and adaptable algorithms that can respond effectively to real-time threats. This requirement has not been fully addressed in the current literature.

2.2 DRL-based Modeling of Cyberattack

Hou et al. [25] studied optimizing DoS attack scheduling using DRL. Their study mainly focused on determining which sensors to attack at each time step in linear time-invariant processes of networked systems, such as smart grid environments. They integrated Q-learning with deep neural networks (DNNs) to maximize the estimation error covariance, offering a method for managing DoS attacks in energy-constrained and computation-limited sensor networks. Yang et al. [26] proposed a query-efficient black-box attack method, called *PatchAttack*, that uses reinforcement learning to optimize the position and texture of adversarial patches superimposed on images to induce misclassification in deep convolutional neural networks. The *PatchAttack* was designed for image recognition tasks, which usually perform in high computing power entities rather than UAVs. It effectively circumvented adversarial defenses using a texture dictionary learned from a Visual Geometry Group (VGG) backbone, representing a group of layers for the feature extraction layer, focusing on texture-based rather than shape-based modifications to the image. Rouzbahani et al. [27] considered an attacker using DRL to target the Internet of Energy-based smart grids, exploiting the dynamic nature of these networks. Unlike UAV systems, they mainly simulated a dummy power system for online learning and initiating various possible False Data Injection Attacks (FDIAs). This approach enabled the attacker to adaptively interact with the smart grid to design complex and dynamic attack strategies.

Although these DRL-based modeling approaches provide insights into optimizing attack strategies, they do not consider the unique challenges of UAV environments, particularly the need for real-time, adaptive defenses against DoS attacks. Our work leverages the principles of DRL in a novel way to protect UAVs from such threats, further extending the application of DRL in cybersecurity.

Limitations of the existing DRL-based attack modeling: The works above [25–27] do not apply to UAV-based network environments because they primarily address issues related to stationary network systems and do not consider the dynamic and mobile nature of UAVs. Hou et al. [25] discussed DoS attacks in a more traditional networked environment, not considering the aerial mobility and specific communication types used by UAVs. Yang et al. [26] mainly investigated computer vision and image processing vulnerabilities, which may be partly relevant to some UAV operations. However, their approaches did not address the broader scope of UAV network security, such as protecting against DoS attacks or optimizing signal strength levels. Rouzbahani et al. [27] considered security issues in static smart grid environments; however, their approaches are not directly applicable to UAV systems with high dynamics and different vulnerabilities.

Building on the application of DRL in UAV operations and the modeling of cyber threats, we now focus on the specific challenge of defending against DoS attacks in UAV networks. While traditional defenses have been proposed, they lack the dynamic adaptability that DRL can offer. Our approach bridges this gap by integrating DRL into defense mechanisms, offering a more resilient solution to evolving cyber threats.

2.3 Defenses Against DoS Attacks in UAVs

Daubert et al. [15] introduced the first honeypot technique specifically designed for UAVs to enhance their security. The honey drone in this work used UAV-specific protocols to lure attackers and ran on devices like Raspberry Pis. It aims to detect and analyze attacks that target drones and redirect attackers away from real drones. Chen et al. [28] discussed a defense mechanism to deal with DoS attacks in real-time UAV systems using containers, called *ContainerDrone* (CD), to prevent excessive resource usage for protecting critical resources, such as CPU, memory, and communication channel, from DoS attacks. Hence, if a certain level of resource utilization is reached, CD stops providing services to avoid drone crashing. We considered the CD one of our work's comparing schemes. Sedjelmaci et al. [29] proposed a hierarchical detection and response system to enhance security against lethal cyber-attacks in UAV networks. This work used a two-layer design that combines a rule-based intrusion detection system (IDS) and an assessment process for each drone to reduce the IDS's overall false positive and negative rates. The IDS was designed to detect GPS spoofing, false data injection, and gray/black hole attacks. Muniraj and Farhood [30] used an IDS to detect sensor attacks on small unmanned aircraft systems (UAS). This IDS was designed to monitor the outputs of the sensors on the UAS and detect any anomalies on the sensors. This IDS is deployed onboard the UAV and is integrated with the navigation system and sensor suite. Ouiazzane et al. [31] proposed a model to detect DoS attacks in UAV networks using various machine learning-based classifiers, such as Random Forest, Naïve Bayes, and Support Vector Machine (SVM) to detect known and unknown DoS attacks. This multi-agent IDS architecture aimed to avoid single points of failure and simplified troubleshooting. The system's knowledge base was continuously updated as new DoS attacks were detected, allowing it to improve its detection capabilities over time. Wang et al. [24] proposed a learning-based game approach for collaborative honeypot defense in UAV networks, focusing on designing optimal, fair, and feasible incentive mechanisms to encourage UAVs to share trapped attack data. Their extensive simulations demonstrated the effectiveness of their scheme in motivating UAV cooperation for improved defensive performance compared to conventional schemes.

Table 1. Comparison of DoS Attack Detection and Prevention Techniques

Technique	Strengths	Weaknesses	Comparison with Honey Drones
Intrusion Detection System (IDS) [29, 30, 32, 33]	- Identifies known attack patterns. - Customizable for different attacks.	- High false positive rates. - Limited against novel attacks.	- Honey drones reduce false positives by luring attackers, improving detection accuracy.
Contained Drone (CD) [28]	- Manages resources under attack. - Maintains system stability.	- May cause mission suspension. - Does not deceive attackers.	- Honey drones actively deceive attackers, preventing mission suspension.
Stationary Honeypot Systems [15, 31, 34]	- Attracts attackers away from critical assets.	- Static deployment. - Easily detected by attackers.	- Honey drones are mobile and harder to detect, increasing the attacker's resource expenditure.
Mobile Honeypot Systems [24]	- Attracts attackers away from critical assets. - Allows for more flexible deployment.	- May still be detected if patterns are predictable.	- Honey drones use DRL to adjust their behavior dynamically, making them more unpredictable.
Game-Theoretic Defense Strategies [19, 24, 27]	- Optimizes defense strategies. - Resource allocation.	- High computational complexity. - Slow adaptation to real-time changes.	- Honey drones adapt quickly with DRL, providing robust defense with lower computational overhead.
DRL-based Attack Modeling [21, 22, 25, 26]	- Optimizes attack strategies. - Adapts to dynamic environments.	- Requires significant resources. - Less effective against adaptive defenses.	- Honey drones use DRL to adjust to attack patterns, providing resilient defense against DRL-based attacks.
DRL-based Defense for UAVs [16–18, 20]	- Enhances UAV security and efficiency. - Handles complex tasks.	- Focused on specific issues. - Lacks comprehensive defense.	- Honey drones enhance DRL-based defense with deception strategies, improving overall security.
Honey Drone (Our Work)	- Adapts to attacker strategies. - Improves mission performance.	- Initial setup required.	- Demonstrated superiority in complex, dynamic environments.

Gudla et al. [34] discussed the vulnerabilities of UAVs and explored various defense techniques to secure them. This work provided several defense techniques, including wireless network encryption, IDS, and moving target defense (MTD) to protect the Parrot AR drone from DoS attacks. The defense techniques were implemented using a Raspberry Pi and Kismet wireless IDS to detect and alert users about any malicious activity on the network. This work concluded that the defense techniques harden the wireless network and protect drones against cyberattacks. Sedjelmaci et al. [35] proposed a cybersecurity system for the UAV network to protect against cyber-attacks, such as injecting wrong data into sensors to compromise data integrity and false GPS coordinates to cause a UAV crash. The IDS agent was deployed on each UAV in the network to monitor the behavior of its neighbors. Each UAV activated the monitoring process and communicated with the base station to report suspicious behavior. The IDS agent assigns a threat level (TL) for each drone's behavior and forwards the TL to the base station for further analysis.

Limitations of the existing defenses against DoS attacks in UAVs: While Daubert et al. [15] pioneered Raspberry Pi-based stationary honeypots for drone security, their static nature and unvarying signal strengths made them detectable to attackers, revealing a critical vulnerability despite being a significant advancement in drone security. Conversely, Wang et al. [24] presents a collaborative defense approach but falls short by not providing a comprehensive description of UAV honeypot designs and components and lacking clarity on deployment strategies. This omission potentially diminishes the defense's effectiveness. Moreover, although it proposes a learning-based game for UAVs to share data on trapped attacks, it fails to seamlessly integrate honeypot deployment within its learning algorithm, possibly limiting the optimization of defense tactics against advanced threats. Prior research [28–31, 34] has fortified UAV security but has not explored cyber deception techniques for proactive protection, a gap this study aims to fill.

To comprehensively understand how our proposed honey drone system compares with these existing approaches, we present a comparative analysis in Table 1. As highlighted, our honey drone-based approach addresses many limitations found in previous methods, offering dynamic adaptation to attacker strategies, enhanced mission

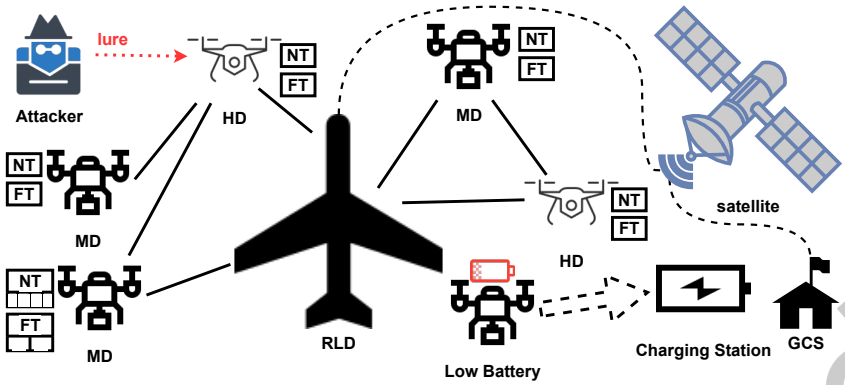


Fig. 1. This example demonstrates a drone fleet utilizing multi-hop communications, consisting of Mission Drones (MDs), Honey Drones (HDs), a Regional Leader Drone (RLD), and a Ground Control Station (GCS). Each drone maintains a Neighbor Table (NT) that lists its immediate communication links. Drones exit the mission to recharge at a charging station when their battery level drops below the threshold T_e. The Fleet Table (FT) tracks the fleet's composition, categorizing drones as actively engaged in the mission, recharging, or compromised.

performance, and reduced false positives. This positions our solution as a robust defense mechanism against DoS attacks in UAV-based mission systems.

3 System Model

3.1 Network Model

We consider a drone fleet executing a surveillance mission. The mission team comprises Mission Drones (MDs) and Honey Drones (HDs) that defend against DoS attacks. A Regional Leader Drone (RLD) is connected to a Ground Control Station (GCS) through a satellite network and also forms a WiFi-based flying ad hoc network (FANET) with MDs and HDs. The FANET [36] supports multihop drone communication, enabling the drones to relay data between each other, thus extending the communication range and ensuring robust connectivity even if some drones are compromised. The GCS, assumed to be trusted and invulnerable to DoS attacks, has robust computational power and firewalls. Fig. 1 describes an example network model. Each drone maintains a neighbor table (NT) that tracks the location information and time-to-live (TTL) [36] of its neighbors, ensuring up-to-date knowledge of the network topology. Additionally, each drone maintains a fleet table (FT) that provides the status of each drone (i.e., active or not) and the membership of the mission team (i.e., member or not). Note that we do not consider any privacy issues beyond our work's scope.

Drones utilize a beacon frame system from IEEE 802.11 WLAN (Wireless Local Area Network) to periodically announce their presence within the network without disclosing sensitive information. To establish a connection and obtain specific details such as ID and location, a newly joined drone must initiate a secure handshake protocol, which is protected by robust encryption (i.e., AES-256). This process is designed to prevent unauthorized access and ensure that such sensitive information is only exchanged after mutual authentication. Once a secure connection is established, drones in the fleet update their NTs, and the collected surveillance data is then transmitted over the network using encrypted User Datagram Protocol (UDP) to preserve both privacy and efficiency. These measures ensure continuous network connectivity while mitigating the risk of privacy breaches.

3.2 Node Model

Our network includes a GCS and a CS. The GCS is the central hub responsible for mission planning and coordination and is equipped with high computational power and secure communication capabilities. The charging station ensures that drones can autonomously recharge when their batteries are depleted, maintaining operational continuity. Three types of drones operate within the fleet: one RLD, multiple MDs, and multiple HDs. The RLD is a high-altitude, long-endurance drone that oversees mission execution and fleet management. MDs are equipped with surveillance cameras for data collection and monitoring. HDs are specialized drones designed to lure and mislead attackers away from MDs, using dynamic signal strength adjustments to mimic vulnerable targets. Drones depart fully charged and return to the CS when the battery is depleted, ensuring they are always ready for the mission.

Table 2. Notation and Symbols

Symbol	Meaning
T_M	Mission duration
$T_M^{\max}$	Maximum mission duration
$\mathcal{M}_t$	Mission effectiveness
sg_{HD}	Signal strength of honey drones (HDs)
sg_{MD}	Signal strength of mission drones (MDs)
sg_{max}	Maximum signal strength level
N_{MD}	Number of MDs
N_{HD}	Number of HDs
$\mathcal{R}_{MC}$	Ratio of completed mission tasks
$\mathcal{EC}$	Energy consumption
$\mathcal{N}_{AS}$	Number of attack successes
G^A	Accumulated reward for attacker
G^D	Accumulated reward for defender
T_C	Time duration of battery being charged in a drone
τ_l	Lower bound of the number of MDs that an HD can protect simultaneously
τ_u	Upper bound of the number of MDs that an HD can protect simultaneously
E_P	Energy consumption rate by a drone
E_C	Energy consumption rate by a camera
ζ	Maximum number of targeted drones by the attacker in a single round
AS_i	Attack strategy
DS_j	Defense strategy
$S_{\text{target},i}$	Set of target drones for the DoS attack
γ^A	Attacker's decay factor in DRL
γ^D	Defender's decay factor in DRL
$\mathcal{R}_t^A$	Attacker's immediate reward
$\mathcal{R}_t^D$	Defender's immediate reward
$\mathcal{S}_t^A$	Attacker's state set
$\mathcal{S}_t^D$	Defender's state set
$\mathcal{A}^A$	Attacker's action set
$\mathcal{A}^D$	Defender's action set
$d(A, \kappa)$	Distance between attacker A and drone κ

3.2.1 Charging Station (CS). Once a drone returns to the CS due to energy depletion, it undergoes full charging before being available again for the mission team. The recharge process takes T_C time, requiring the drone to be unavailable for the duration of charging. This can potentially affect the overall mission performance, particularly if the task has time constraints and there are no backup drones.

3.2.2 Ground Control Station (GCS). Given a mission with a maximum completion time of $T_M^{\max}$, the GCS monitors the mission progress reported by the RLD via a satellite network [37], while the RLD optimizes drone operations under the constraints of satellite communications.

3.2.3 Regional Leader Drone (RLD). The RLD is a high-altitude, long-endurance drone that manages fleet trajectory and signal strength during the mission duration, T_M ($\leq T_M^{\max}$). The RLD can adapt to drone offline/online status, reconfiguring drone trajectories and protecting MDs from DoS attacks. If compromised, a backup RLD will be launched from the GCS. The RLD uses a DRL agent to control drone signal strengths for maximizing mission performance ($\mathcal{M}_t$).

3.2.4 Mission Drone (MD). MDs, equipped with cameras for surveillance [38], are responsible for detecting suspicious objects in the target region. They transmit the collected data to the RLD via the FANET [36]. Each MD follows an assigned trajectory and updates their online/offline status to the RLD. If an MD goes offline, the trajectories of the remaining drones are recalculated to ensure mission continuity. Initially, each MD is loaded with a specific trajectory for the mission based on the known parameters of the target region and mission.

3.2.5 Honey Drone (HD). Drawing inspiration from the smartphone honeypot concept [39] and known UAV vulnerabilities [40], we introduce a drone-based honeypot system, named 'Honey Drone' (HD), to shield MDs from DoS attacks. The HD is structured with two logically distinct parts: a honeypot virtual machine (VM) and an infrastructure VM. The honeypot VM is crafted to emulate a susceptible drone system, making it a lucrative target for potential DoS attacks. To increase the authenticity and intricacy of the decoy, this VM presents a dynamically changing set of open ports, mirroring the behavior of an active drone rather than a static multitude of open ports. Complementarily, the infrastructure VM, operating on a lightweight Linux environment, monitors the memory and log of the honeypot VM.

The infrastructure VM plays a crucial role when a DoS attack is detected by abnormal computing consumption (e.g., a spike in CPU or memory usage). In that case, the infrastructure VM's back channel informs the RLD about the port used by the attacker. Consequently, the RLD reconfigures the open port of the MD to evade any further attacks. Should the honeypot VM become compromised or malfunction, the honeypot VM can be restored. This honeypot structure can be run on a mobile device [39], as it is lightweight and will not overload the drone. For effective utilization of HDs, we deploy HDs based on Algorithm 1. Each HD is responsible for a set of MDs with a size between τ_l and τ_u for protection. Depending on the number of MDs assigned to each HD and the HD's location, some HDs may not be assigned with any MDs and may move around to find available MDs. If the case happens (i.e., line 13), the HD finds a location when $\tau_l \leq |N(l'_H)| \leq \tau_u$. We search only MDs' positions to reduce computational complexity instead of all cells. This will reduce the complexity of Algorithm 1 from $O(|L_H||N_{cell}|)$ to $O(|L_H||L_M|)$, where the number of cell $|N_{cell}|$ is significantly higher than $|L_M|$. Algorithm 1 runs until all MDs are assigned to the available HDs. Algorithm 2 illustrates the complete learning process in which the defender utilizes the A3C method to choose a strategy and deploy HDs to protect MDs. To further clarify, we provide the following list summarizing the key symbols and notations used in Algorithm 1:

- L_M: The set of active MDs not currently in the GCS.
- L_H: The set of active HDs.
- P_r^H: The signal radius/range of an HD when the RLD selects $sg_{HD} = DS_j$.
- D(x, y): The distance between two drones x and y.

Algorithm 1 Honey Drone Deployment

$L_M \leftarrow$ A set of active MD that not in GCS
$L_H \leftarrow$ A set of active HD
$P_r^H \leftarrow d(sg_{HD}), sg_{HD} \leftarrow DS_j$ ▷ The signal radius/range of an HD when RLD selects $sg_{HD} = DS_j$ for the HD
$\mathrm{D}(x, y) \leftarrow$ The distance between two drones x and y
$S = \emptyset$ ▷ A set of HDs with assigned MDs
$[\tau_l, \tau_u] \leftarrow$ The lower and upper bounds of the number of MDs that HDs can protect simultaneously
for $l_H \in L_H$ **do**
  if $|L_M| = 0$ **then**
    $S \leftarrow S \cap \{l_H\}$
  else
    $N(l_H) = \{\mathrm{D}(l_H, l_M) < P_r^H : l_M \in L_M\}$ ▷ A set of MDs in the protect range of HD l_H
    if $|N(l_H)| < \tau_l$ **then**
      Find l'_H with new position such that $\tau_l \leq |N(l'_H)| \leq \tau_u$
      where $N(l'_H) = \{\mathrm{D}(l'_H, l_M) < P_r^H : l_M \in L_M\}$ ▷ A set of MDs detected/protected by HD l'_H
      $L_M \leftarrow L_M \setminus N(l'_H)$ ▷ Remove protected MDs from set L_M
      $S \leftarrow S \cap \{l'_H\}$ ▷ Add deployed HD to set S
    else if $\tau_l \leq |N(l_H)| \leq \tau_u$ **then**
      $L_M \leftarrow L_M \setminus N(l_H)$
      $S \leftarrow S \cap \{l_H\}$
    else
      $N'(l_H) = \{n_1, \ldots, n_\tau : n_{\tau_u} \in N(l_H)\}$ ▷ Generate the set of MD with size lower than upper bound τ_u and assign MDs to HD l_H
      $L_M \leftarrow L_M \setminus N(l'_H)$
      $S \leftarrow S \cap \{l'_H\}$
    end if
  end if
end for
Output: HD location set $S = \{l_{H1}, l_{H2}, \ldots, l_{H|L_H|}\}$

- S: The set of HDs assigned to protect specific MDs.
- $[\tau_l, \tau_u]$: The lower and upper bounds on the number of MDs that HDs can protect simultaneously.

Furthermore, HDs can also serve as communication relays to stabilize network connections within the fleet. By maintaining stronger signal strengths, they help ensure robust communication pathways, indirectly supporting the MDs by providing reliable data relay points. This dual functionality of HDs, both as decoys and relay nodes, significantly enhances the resilience and performance of the UAV network.

3.3 Energy Model

We assume that both MDs and HDs are Crazyflie 2.X quadrotor drones [41], which can stream video [42] and have autonomous recharging capabilities [43]. We chose Bitcraze's Crazyflie nano-quadrotor drones for our proposed approach due to their compact size, agility, and suitability for both indoor and outdoor missions. The Crazyflie offers extensive developer support, including open-source firmware and modular expansion options like GPS and AI decks, enabling customization for advanced surveillance tasks. Detailed energy consumption data allows for accurate modeling, and the active community provides valuable insights and resources. These attributes make the Crazyflie an ideal choice for implementing and testing our honey drone-based defense strategy.

We detail the simulation setup in Section 6. An MD and an HD consume energy as follows:

$$E_{MD} = E_P + E_C + E_R \times \frac{sg_{MD}}{sg_{max}}, \quad E_{HD} = E_P + E_R \times \frac{sg_{HD}}{sg_{max}}, \tag{1}$$

Algorithm 2 Honey drone deployment and strategy selection for signal strength adjustment (pseudocode for each thread)

// Assume global shared parameter vectors θ and θ_v and global shared counter $T = 0$
// Assume thread-specific parameter vectors θ' and θ'_v
Initialize thread step counter $t \leftarrow 1$
$L_M \leftarrow$ A set of active and not in GCS MD locations
$L_H \leftarrow$ A set of active HD locations
$P_r^H \leftarrow d(sg_{HD}), sg_{HD} \leftarrow DS_j$ ▷ The signal radius/range of an HD when RLD selects $sg_{HD} = DS_j$ for the HD
$\mathrm{D}(x, y) \leftarrow$ The distance between two drones x and y
$S = \emptyset$ ▷ A set of HDs with assigned MDs
$[\tau_l, \tau_u] \leftarrow$ The lower and upper bounds of the number of MDs that HDs can protect simultaneously
repeat
  Reset gradients: $d\theta \leftarrow 0$ and $d\theta_v \leftarrow 0$.
  Synchronize thread-specific parameters $\theta' = \theta$ and $\theta'_v = \theta_v$
  $t_{\text{start}} \leftarrow t$
  Get state s_t
  repeat
    Perform strategy a_t according to policy $\pi(a_t|s_t;\theta')$
    for $l_H \in L_H$ **do**
      if $|L_M| = 0$ **then**
        $S \leftarrow S \cap \{l_H\}$
      else
        $N(l_H) = \{\mathrm{D}(l_H, l_M) < P_r^H : l_M \in L_M\}$ ▷ A set of MDs in the protect range of HD l_H
        if $|N(l_H)| < \tau_l$ **then**
          Find l'_H with new position such that $\tau_l \le |N(l'_H)| \le \tau_u$
          where $N(l'_H) = \{\mathrm{D}(l'_H, l_M) < P_r^H : l_M \in L_M\}$ ▷ A set of MDs protected by HD l'_H
          $L_M \leftarrow L_M \setminus N(l'_H)$ ▷ Remove protected MDs from set L_M
          $S \leftarrow S \cap \{l'_H\}$ ▷ Add deployed HD to set S
        else if $\tau_l \le |N(l_H)| \le \tau_u$ **then**
          $L_M \leftarrow L_M \setminus N(l_H)$
          $S \leftarrow S \cap \{l_H\}$
        else
          $N'(l_H) = \{n_1, \ldots, n_\tau : n_{\tau_u} \in N(l_H)\}$ ▷ Generate the set of MD with size lower than upper bound τ_u and assign MDs to HD l_H
          $L_M \leftarrow L_M \setminus N(l'_H)$
          $S \leftarrow S \cap \{l'_H\}$
        end if
      end if
    end for
    Deploy HDs according to location set $S = \{l_{H1}, l_{H2}, \ldots, l_{H|L_H|}\}$
    Receive reward r_t and new state s_{t+1}
    $t \leftarrow t + 1$
    $T \leftarrow T + 1$
  until terminal s_t or $t - t_{\text{start}} == t_{\max}$
  $R = \begin{cases} 0 & \text{for terminal } s_t \\ V(s_t, \theta'_v) & \text{for non-terminal } s_t \text{ // Bootstrap from last state} \end{cases}$
  for $i \in \{t-1, \ldots, t_{\text{start}}\}$ **do**
    $R \leftarrow r_i + \gamma R$
    Accumulate gradients wrt $\theta' : d\theta \leftarrow d\theta + \nabla_{\theta'} \log \pi(a_i|s_i;\theta')(R - V(s_i;\theta'_v))$
    Accumulate gradients wrt $\theta'_v : d\theta_v \leftarrow d\theta_v + \frac{\partial (R - V(s_i;\theta'_v))^2}{\partial \theta'_v}$
  end for
  Perform asynchronous update of θ using $d\theta$ and of θ_v using $d\theta_v$.
until $T > T_{\max}$

where E_C is the consumption rate in mW of a Himax camera [38] according to [44]. E_P is the consumption rate in mW used for a drone platform consisting of Standard Operating Conditions (SoCs) and four motors and is estimated by the flight time of the drone platform. The real-time energy consumption of the radio can be represented by the equation $E_R \times \frac{sg_{MD}}{sg_{max}}$ and $E_R \times \frac{sg_{HD}}{sg_{max}}$, where E_R is the maximum energy consumption rate of the radio, sg_{max} is the maximum signal strength, and sg_{MD} and sg_{HD} are the signal strength of the MDs and HDs, respectively.

3.4 Threat Model

Our work concerns DoS attacks, which are recognized as the most common and serious threats to UAVs [25, 28, 31, 40, 45]. We simulate a DoS attack that sends multiple JSON connection requests simultaneously to a drone. DoS attacks can disconnect a drone from the fleet network, impair its navigation control, and lead to a crash [40]. This results in compromised MDs being removed from the mission, and negatively affecting the mission team's performance.

In the envisioned DoS attack scenario, the attacker starts by leveraging an RF Spectrum Analyzer to assess the signal strength of drones in the fleet. Positioned to intercept the fleet's wireless communications, the attacker uses *Wireshark* for network traffic analysis. This involves capturing and analyzing packets exchanged within the drone network, enabling the attacker to extract critical details like IP addresses, communication protocols, and insights into the network's structure.

Despite the drone fleet employing advanced security measures, such as encryption, digital signatures, and authentication, these systems are not immune to the flood of connection requests characteristic of a DoS attack. Post reconnaissance, the attacker employs *Nmap* to conduct a detailed scan of the identified target drone. *Nmap* reveals crucial information, such as open ports and services, which the attacker uses to strategize the DoS attack. The actual execution of the DoS attack involves utilizing the information gathered from *Wireshark* and *Nmap*. The attacker sends many connection requests to overwhelm the drone's network. For example, they might use the command *telnet 192.168.42.1 44444* to send thousands of parallel connection requests to a port identified as vulnerable during the *Nmap* scan. These requests aim to saturate the drone's processing capabilities, causing significant lag or complete operational failure [40]. This method demonstrates that even with robust security protocols, drones can be vulnerable to volume-based network attacks, highlighting the need for additional protective measures like honey drone deployment to lure such attacks. The selection of an attack strategy by a DoS attacker is further discussed in Section 5.1.

DoS attackers may use infrastructure, such as high-power antennas and a stable power source. We assume the DoS attackers are positioned on the ground. To avoid physical detection, the DoS attackers are not in close proximity to the UAV mission team. Each drone's presence and its connectivity with the UAV network can be detected by the signal strength level [46]. In addition, recipients receiving such strong signals can save their energy [47]. DoS attackers often target signals with higher strength. For example, attackers focus on stronger Wi-Fi signals to maximize data transfer and improve their attack success rate [48]. By exploiting stronger signals for more efficient data transmission, these attacks become more effective and impactful. Therefore, the attacker can target UAVs with stronger signals to maximize energy conservation. Although the DoS attackers are physically distant from the UAV network, the attacker is aware of the existence of HDs, which often use higher signal strength levels to lure attacks. Thus, the attacker cannot indiscriminately select UAVs with strong signals to save energy, which would risk attacking HDs. Instead, the attacker must carefully balance the goal of energy conservation with the need to avoid HDs while effectively attacking legitimate MDs.

To facilitate a clear understanding of the various symbols and notations used throughout this paper, we provide a summary in Table 2.

4 Problem Statement

UAVs are increasingly used for surveillance missions, where maintaining consistent communication and defending against cyber threats such as DoS attacks are critical challenges. In such scenarios, optimizing the signal strength of both HDs and MDs becomes essential to balance mission effectiveness and security.

This work aims to develop a UAV-based surveillance mission system that dynamically adjusts the signal strengths of HDs and MDs to maximize mission performance while ensuring robust defense against DoS attacks. Specifically, we aim to optimize the system's performance by leveraging DRL to achieve a balance between the following:

- **Mission Completion Ratio** ($\mathcal{R}_{MC}$): Ensuring a high ratio of completed mission tasks.
- **Energy Consumption** ($\mathcal{EC}$): Minimizing the overall energy consumption of the UAVs.
- **Defense Effectiveness** ($\mathcal{N}_{AS}$): Reducing the number of successful attacks on the system.

Formally, the optimization problem can be expressed as:

$$\underset{sg_{HD},\ sg_{MD}}{\arg\max}\ \mathcal{M}_t(sg_{HD}, sg_{MD}), \quad s.t. \quad sg_{MD} \leq sg_{HD},$$

where the mission effectiveness $\mathcal{M}_t(sg_{HD}, sg_{MD})$ is defined as:

$$\mathcal{M}_t(sg_{HD}, sg_{MD}) = \frac{\alpha \cdot \mathcal{R}_{MC}}{\beta \cdot \mathcal{EC} + \lambda \cdot \mathcal{N}_{AS}}.$$

Here, α, β, and λ are weighting factors that adjust the relative importance of each component.

The challenge lies in finding the optimal signal strength levels for HDs and MDs that satisfy these objectives, particularly in the presence of intelligent adversaries who may target drones with higher signal strengths to maximize disruption. By employing DRL, we dynamically adjust these signal strengths based on real-time feedback from the environment, ensuring that mission objectives are met while maintaining system security.

5 DRL-based Strategy Selection

The inherent unpredictability of attack behaviors poses significant challenges for UAV-based mission systems. It becomes particularly problematic since defense strategies cannot be preplanned and must adapt dynamically in real time. The high latency in remote control via satellite networks further underscores the need for autonomous decision-making without reliance on central instructions (see Section 3). In this regard, DRL presents an ideal solution, equipping agents with the ability to interact, learn, and make informed decisions in an evolving environment without any prerequisite knowledge. In our proposed framework, the DRL agent determines the optimal signal strength, a key determinant in maintaining network connectivity, facilitating effective mission execution, and countering DoS attacks.

We formulate the attack-defense interactions as a game where the attacker and defender strive to achieve their individual goals: the attacker aims for mission failure by compromising the mission-critical drones, while the defender aims for mission success by protecting these drones and ensuring mission completion. The time to complete a mission is denoted by T_M, constrained by a maximum limit $T_M^{\max}$. If $T_M > T_M^{\max}$, it indicates that only a part of the target region has been covered or that the mission has failed. The mission unfolds through interaction rounds between the attacker and the defense system, persisting until the mission ends successfully, partially, or fails. Each interaction round is defined as a single simultaneous action taken by the attacker and defender based on knowledge gleaned from previous rounds. The mission team's success is evaluated using the count of completed cells in the target region. The mission team, composed of the RLD, multiple MDs, and HDs, launches from the GCS and continues until the mission ends. The RLD coordinates the fleet, assigns tasks to MDs, and dynamically adjusts the signal strengths of HDs to lure potential attacks and protect MDs. The attacker, on the

other hand, monitors signal strengths and patterns to identify and target MDs, aiming to disrupt the mission by causing drone failures.

5.1 Attacker Model

5.1.1 Attacker's Action Space based on Signal Strengths. The attacker observes the signal strengths of the drones and selects the attack strategy accordingly. We define the attack strategy as $AS_i \in \{AS_1, \dots AS_{10}\}$, where each action corresponds to a range of received signal strengths $[sg_i^l, sg_i^u]$ chosen by the attacker to form the set of target drones, $S_{\text{target},i}$, for the DoS attack. If $S_{\text{target},i} = \emptyset$, then it implies no attack. The set $S_{\text{target},i}$ is defined by:

$$S_{\text{target},i} = \{\kappa \mid sg_i^l \leq sg_\kappa \leq sg_i^u\}, \tag{2}$$

where κ is the ID of a drone and sg_κ is the signal strength received by the attacker from drone κ. The resource-limited attacker can target at most ζ drones. We map the attacker's strategies to ten signal strength ranges in *dBm*:
$[sg_i^l, sg_i^u] \in \{(-100, -98.1], (-98.1, -96.1], (-96.1, -93.8], (-93.8, -91.1],$
$(-91.1, -87.9], (-87.9, -84.0], (-84.0, -79.0], (-79.0, -72.0], (-72.0, -60], (-60, 20]\}$.

The ranges are formed based on Eq. (3), which evenly groups drones based on the distance between the defender and attacker and then converts the distances to the received signal strength. The signal strength in *dBm* decreases as the distance between the transmitter and receiver increases. Based on the wireless signal attenuation model [49], which has also been validated by the Bitcraze company under real-world conditions [41], we estimate the attenuation as follows:

$$P_{dBm}(d) = P_{dBm}(d_0) - \eta \cdot 10 \cdot \log_{10}(\frac{d}{d_0}), \tag{3}$$

where $\eta = 4$ is a path loss exponent with an average value of 4 for mobile devices and $P_{dBm}(d)/P_{dBm}(d_0)$ is an observed signal strength at a distance d/d_0, respectively. According to the drone's specifications [41], the $P_{dBm}(d_0) = 20$ dBm (decibel-milliwatts) when $d_0 = 1$ m (meter). Since -60 dBm and -100 dBm are typical values for the very strong and lowest signal strength the drone can use [50], we obtain that a signal is strong when $d < 100m$, and the signal is weak when $d = 1000m$. Based on the real-world test result [41], the maximum control range of $1000m$ well reflects a real-world scenario described in Eq. (3). The Euclidean distance (d) between attacker A and drone κ is estimated by:

$$d(A, \kappa) = \sqrt{|x^A - x^\kappa|^2 + |y^A - y^\kappa|^2 + |z^A - z^\kappa|^2}, \tag{4}$$

where x^κ, y^κ, and z^κ (height) are 3D-coordinates for drone κ's location and x^A, y^A, and z^A are the coordinates of attacker A's location where $z^A = 0$ for its ground location.

5.1.2 DRL-based Attack Strategy Selection. The attacker DRL agent identifies the best strategy AS_i to maximize its accumulated reward, G^A, by:

- **State Set** ($\mathcal{S}_t^A$): A state is defined as $\mathcal{S}_t = \{T_M^t, N_{AS}^t, N_A^t, S_{\tilde{sg}}^t\}$, where T_M^t represents the elapsed time, N_{AS}^t is the number of time attack success at time t, N_A^t is the number of attacks performed at time t, and $S_{\tilde{sg}}^t$ is a set of the received signal levels of the drones at round t. After the attacker launches attacks on target drones and does not detect some drones' signals, the attacker treats it as a success. Since a drone's signal may not be detected due to its energy depletion, the misdetection of attack success is possible.
- **Action Set** ($\mathcal{A}^A$): We define $\mathcal{A}^A = \{a_1, \dots, a_i, \dots, a_n\}$, where a_i refers to AS_i that determines the set of target drones, $S_{\text{target},i}$. When the attacker DRL agent takes action i at round t, we denote it by a_i^t.
- **Reward Function** ($\mathcal{R}_t^A(a_i^t)$): The attacker's immediate reward by taking a_i^t is obtained by $\mathcal{R}_t^A(a_i^t) = 1 - \exp\left(-\frac{1+N_{\text{inactive}}}{T_M^t}\right)$, where N_{inactive} is the number of inactive drones, and T_M^t is the time elapse. This

reward function implies that the attacker aims to maximize the total number of inactive, compromised drones in the system. The attacker DRL agent aims to maximize its discounted sum of rewards over an entire episode represented as $G^A = \sum_{t=0}^{\infty} \gamma^{At} \cdot \mathcal{R}_t^A$ where R_t^A is the attacker's immediate reward at time t and γ^A is the attacker's decay factor.

5.2 Defender Model

5.2.1 Defender's Action Space based on Signal Strengths. We use a defense strategy, $DS_j \in \{DS_1, \ldots, DS_{10}\}$, to control the signal strength of HDs, sg_{HD}. The signal strength of MDs is determined by $sg_{MD} = sg_{HD} - \rho$, where ρ is a predefined integer (see Table 3). We evenly split the signal transmission range from 100m to 1000m based on Eq. (3) and map $sg_{HD} \in \{-20, -7.9, -0.9, 4.0, 7.9, 11.1, 13.8, 16.1, 18.1, 20\}$. We leverage DRL to identify the optimal defense strategy, sg_{HD}, for adjusting MDs' and HDs' signal strength levels.

5.2.2 DRL-based Defense Strategy Selection. The defender DRL agent identifies an optimal signal strength of HDs to maximize its accumulated reward, G^D, by:

- **State Set** ($\mathcal{S}_t^D$): $\mathcal{S}_t^D = \{T_M^t, \mathcal{R}_{MC}^t\}$, where T_M^t represents the duration elapsed in the mission and $\mathcal{R}_{MC}^t$ is the ratio of completed mission tasks at time t.
- **Action Set** ($\mathcal{A}^D$): We define $\mathcal{A}^D = \{a_1, \cdots, a_j, \cdots, a_m\}$, where a_j corresponds to DS_j that determines the HDs' signal strength. We set the signal strength level of MDs based on Section 5.2.1. We denote the defender's action j taken at round t by a_j^t.
- **Reward Function** ($\mathcal{R}_t^D(a_j^t)$): The defender's immediate reward by taking a_j^t is given by: $\mathcal{R}_t^D(a_j^t) = \exp(-\frac{1}{\mathcal{N}_{AC}})$, where N_{AC} is the number of connected non-compromised, active drones. This reward function means that the defender aims to maximize the total number of connected active, non-compromised drones for seamless mission execution. The defender DRL agent aims to maximize its discounted sum of rewards over an entire episode, given by $G^D = \sum_{t=0}^{\infty} \gamma^{Dt} \cdot \mathcal{R}_t^D$ where R_t^D is the defender's immediate reward at time t and γ^D is its decay factor. The energy consumption ($\mathcal{EC}$) is indirectly factored into the reward function via $\mathcal{N}_{AC}$. Higher signal strength improves connectivity but increases energy use, potentially lowering $\mathcal{N}_{AC}$ as drones leave for recharging. This forces the DRL agent to balance connectivity and energy conservation, keeping $\mathcal{N}_{AC}$ high. We considered alternative reward functions that explicitly included $\mathcal{EC}$, but they did not significantly change the results. Thus, we chose the current reward function for its simplicity, as it inherently accounts for energy consumption through its effect on $\mathcal{N}_{AC}$.

The A3C algorithm operates within the RLD to optimize the signal strengths of HDs dynamically. The algorithm runs multiple parallel workers, each interacting with the environment independently, to explore various strategies simultaneously. This parallelism accelerates the convergence of learning and enables the RLD to make real-time, adaptive decisions based on the evolving mission environment. By continuously updating the signal strengths of HDs, the RLD enhances the overall mission performance and effectively mitigates the impact of DoS attacks. The use of A3C also ensures that the system remains resilient and adaptive in the face of complex, dynamic challenges during mission execution.

Fig. 3 describes the high-level overview of our proposed honey drone-based mission system and how DRL agents take attack/defense strategies.

6 Experimental Setup

6.1 Simulation Environment Setup

We implement the Asynchronous Advantage actor-Critic (A3C) agents [23] using the PyTorch deep learning library [51] and use the *gym-pybullet-drones* simulator [52] as our experimental testbed. The *gym-pybullet-drones* is chosen for its multiagent gym-like API commonly used for RL experiments. The *PyBullet Physics Engine* [53]

Table 3. Key Design Parameters and Default Values

Symbol	Meaning	Default
ρ	Signal strength decrement interval when calculating sg_{MD}	5
T_C	Time duration of battery being charged in a drone	30
$T_M^{\max}$	Maximum mission duration	30
N_{MD}	Number of mission drones (MDs)	5
N_{HD}	Number of honey drones (HDs)	2
$[\tau_l, \tau_u]$	The lower bound and upper bound of the maximum number of MDs that an HD can protect simultaneously	[2, 4]
E_P	Energy consumption rate by a drone	7,900 mW
E_C	Energy consumption rate by a camera	4 mW
ζ	Maximum number of targeted drones by the attacker in a single round	5
$P_{dBm}(d_0)$	Signal strength at a reference distance $d_0 = 1$ meter	20 dBm
sg_{max}	Maximum signal strength level	20 dBm
η	Path loss exponent used in the wireless signal attenuation model	4
γ^A	Attacker's decay factor in the DRL reward function	0.92957
γ^D	Defender's decay factor in the DRL reward function	0.59428
lr	Learning rate (lr) for DRL agents, (defender/attacker)	0.001427/0.00192
lr decay	Learning rate decay factor for the learning rate in DRL agents, (defender/attacker)	0.97967/0.98718

by *gym-pybullet-drones* is lightweight, open-source, and compatible with Python and PyTorch. To reduce the training time, we run multiple environments in multiple threads on the multicore CPU. We generate experimental results using twenty-four 128-Threads AMD EPYC 7702 CPUs [54].

A3C is an RL algorithm using an Advantage Actor-Critic (A2C) neural network architecture consisting of an actor and a critic component. While A2C uses a single agent to learn, A3C (see Fig. 2) deploys multiple workers in parallel to learn from the environment independently and asynchronously. After that, A3C uses an asynchronous gradient descent for network optimization. Based on the parallel feature, A3C can update its policy gradient more frequently, allowing faster convergence. Owing to using independent environments, it also provides a stabilizing effect on the training process.

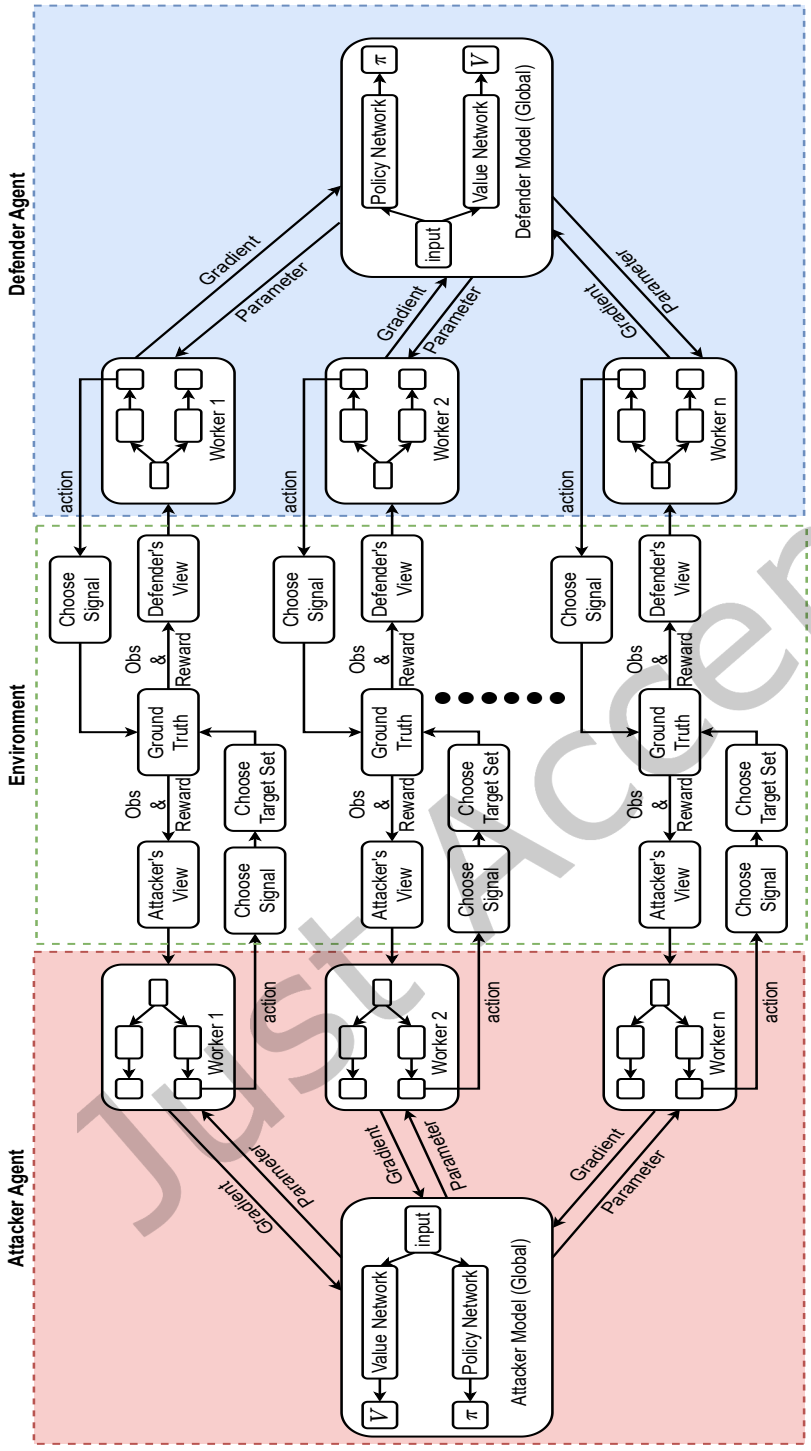


Fig. 2. The DRL framework addresses interactions between attackers and defenders, with both parties selecting strategies via the A3C algorithm within the DRL paradigm. This setup emphasizes the strategic decision-making process informed by advanced AI techniques.

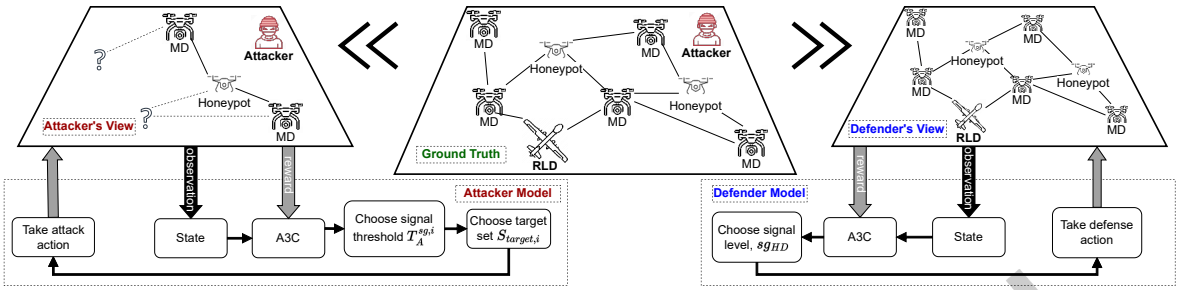


Fig. 3. The proposed system focuses on missions using HDs, with DoS attackers and defenders utilizing DRL to choose strategies, specifically optimal signal strengths, based on their unique, partially observed information. This methodology highlights the use of advanced AI to improve operational resilience and effectiveness against cyber threats.

For efficient RL-based experiments, we use the *Optuna* optimization framework [55] as an AutoML technique allowing the automatic configuration of optimal hyper-parameters, including γ (decay factor in a reward function), a learning rate and its decay, the probability to trigger exploration ϵ, and ϵ decay.

We consider a target area with a 500m × 500m square, consisting of 25 cells with a size of 100m × 100m each. We place the GCS and CS together, both outside the target area. For the drone fleet, we assign five MDs and two HDs. Table 3 summarizes the meanings and default values of the key design parameters used in our work. In Eq. (1) that describes our energy model in Section 3.3, we set $E_C = 4$ mW, representing the consumption rate of a Himax camera [38]. We also set $E_P \approx 7,900$mW, the consumption rate for a drone platform with SoC, radio, and four motors. Following [41], E_P is set to a 250 mAh, 3.7 V LiPo battery, allowing a drone to fly 7 minutes (≈ 0.117 hr.), resulting in $\frac{250mAh*3.7V}{0.117hr.} \approx 7,900$mW.

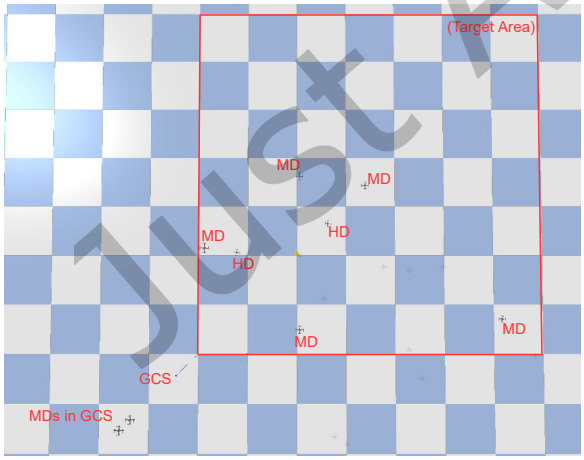


Fig. 4. Drone fleet in Pybullet physical engine environment.

A drone is considered crashed when its height (z-axis) is below 0.1 meters. A drone may crash when: (1) DoS attacks cause errors in its navigation function; (2) Two drones collide; (3) Improper operations (e.g., too fast movement or sudden stop) cause loss of balance; or (4) A drone drains its battery. The path-finding algorithm (see Section 3.2.4) can use only a subset of MDs to perform a mission if more drones are available than the required number of drones for executing a given mission. If some of the drones in mission execution deplete their energy, the remaining drones in GCS can quickly replace the energy-drained drones to complete the mission continuously. If no drones are available in GCS, the mission team must wait for the drone to return after being recharged. If a drone in mission execution crashes, the RLD will rerun the path-finding algorithm and allocate a new drone from GCS. If no drones are available in GCS, the drone's crash will also impact mission performance. Figs. 2 and 4 show the structure of the attacker and defender using A3C and the screenshot of the *gym-pybullet-drones* simulator.

6.2 Metrics

We use the following metrics for our performance analysis:

- **Accumulated Reward** ($\mathcal{G}^A$ or $\mathcal{G}^D$): This is the accumulated rewards, G^A and G^D (see Sections 5.1.2 and 5.2.2), respectively. Recall that the attacker maximizes the total number of inactive, compromised drones while the defenders maximize the number of active, non-compromised drones in the system.
- **Drones' Energy Consumption** ($\mathcal{EC}$): This refers to the total energy consumed by HDs and MDs of the mission system during the entire mission.
- **Number of Attack Successes** ($\mathcal{N}_{AS}$): This indicates the number of attack successes performed by the DoS attacker over drones in the mission system.
- **Ratio of Completed Mission Tasks** ($\mathcal{R}_{MC}$): This measures how many cells are completed among all assigned cells for the surveillance mission by the time the mission finishes at min $\left[T_M^t, T_M^{\max}\right]$ where T_M^t is the mission duration until round t and $T_M^{\max}$ is a predefined maximum mission duration allowed.
- **Simulation Running Time** ($\mathcal{RT}_S$): This measures the time taken for each simulation run in seconds.

6.3 Comparing Schemes

We examine the performance of defenses considered to handle DoS attacks in the considered mission system. We consider the following cases in terms of what approach each party uses for its decision-making, such as selecting a strategy at random (R) or using a DRL approach (A3C) when HDs are used (HD) or not (No-HD). We denote each scheme name as X-Y-Z where X is an attacker's strategy selection method, Y is a defender's strategy selection method, and Z is a defense mechanism used, such as honey drones (HD), no-honey drones (No-HD), contained drone (CD) [28], or intrusion detection system (IDS) [29, 32, 33]. For example, A3C-A3C-HD refers to the case in which the attacker and defender use A3C to select their strategy (i.e., signal strength level), and the defense mechanism used is honey drones. When no honey drones are used, we consider A3C-A3C-No-HD, A3C-IDS, and A3C-CD, where A3C-A3C-No-HD allows the defender to select MDs' signal strength level using A3C while A3C-IDS and A3C-CD mean the attacker only selects signal strength levels to identify target drones to crash. In contrast, the defender uses a fixed signal strength level with a given defense, such as CD or IDS. We selected the CD [28] and IDS [29, 32, 33] because they are defenses that mainly handle DoS attacks in UAV settings.

To ensure the validity of our results, we run our simulations 100 times for each scenario and report the average values from the corresponding results. When HDs are used, we consider five MDs and two HDs. For cases without HDs, we considered only five MDs. When the attacker and defender made decisions using A3C (i.e., A3C-A3C), the A3C model employed multiple parallel workers, known as *local workers*, to interact with the environment. These local workers calculate the parameters' gradients, such as weights and biases, and send them to the global model. The global model then updates its networks and distributes the updated parameters to the local workers. The local workers only have access to a limited view of the environment. The reward functions for the attacker and defender are distinct, as they depend on whether the worker is on the attacker's or defender's side. By implementing this method, we simultaneously engage multiple workers in interacting with the environment. Our experiments utilize a 128-core CPU, creating 128 local workers per process and theoretically increasing the speed of our training by 128 times compared to the A2C. The source code is available at https://github.com/Wan-ZL/ARO-Foureye.

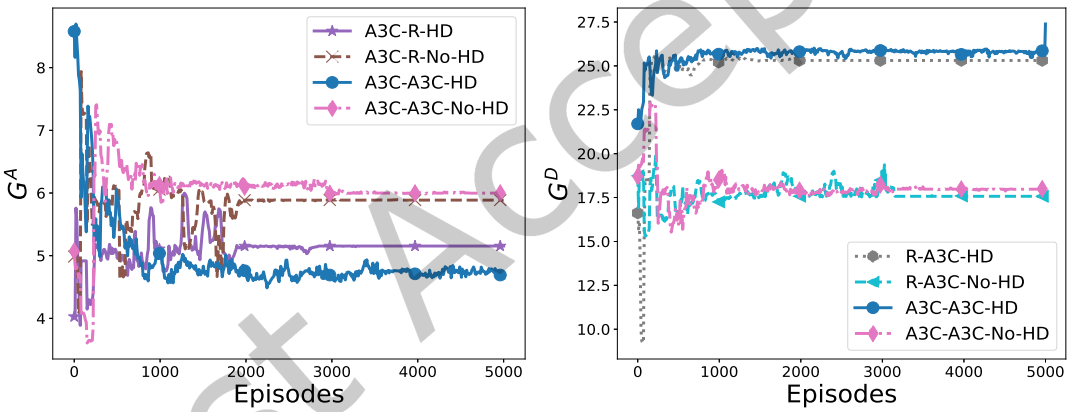
7 Numerical Results and Analyses

For all experimental results, we use the following hyperparameters' settings using the *Optuna* optimization tool [55]. For γ (reward decay factor), the defender uses 0.59428, and the attacker uses 0.92957. For the learning

rate, the defender uses 0.001427, and the attacker uses 0.00192. The decay factor of the learning rate is set to 0.97967 for the defender and 0.98718 for the attacker. The probability of triggering exploration (ϵ) is set to 0.073893 for the defender and 0.21362 for the attacker. For the decay factor of ϵ, we use 0.90245 for the defender and 0.90012 for the attacker.

7.1 Accumulated Reward Analysis

This section analyzes the accumulated rewards of the DRL-based strategies the attacker and defender take in a given mission setting. Fig. 5 illustrates the accumulated rewards (i.e., the discounted sum of rewards over an entire episode) of the attacker and defender, denoted by G^A and G^D in Sections 5.1.2 and 5.2.2, respectively. Fig. 5 shows the following. First, the defender's performance improves with introducing HDs, while the attacker can perform better when the defender does not use HD (No-HD). Notice that A3C-A3C-HD outperforms A3C-A3C-No-HD in Fig. 5b. This implies that using HDs enhances the defender's performance against DoS attacks. Fig. 5a shows that the performance difference between A3C-A3C-HD and A3C-A3C-No-HD is more pronounced than the performance difference between A3C-R-HD and A3C-R-No-HD. This implies that using HDs with DRL-based signal strength selection contributes to significant effectiveness in defense. However, with no HDs, the attacker with DRL can identify the defense pattern and achieve a higher reward. This proves that using HDs provides an additional layer of defense to counteract the attacker's learning strategy.



(a) G^A (b) G^D

Fig. 5. The attacker's and defender's accumulated rewards (G^A, G^D) with and without using honey drones (i.e., no defense).

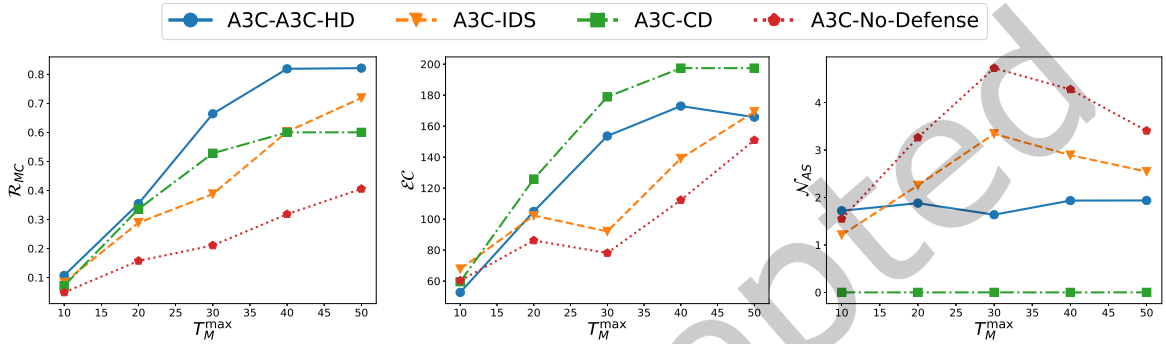
7.2 Comparative Performance Analysis

We compare our proposed defense technique with existing techniques, including intrusion detection systems (IDS) and *ContainerDrone* (CD), based on the metrics of Section 6. As Fig. 6 shows, A3C-A3C-HD outperforms the other strategies. Specifically, Fig. 6a shows that A3C-A3C-HD achieves the highest ratio of completed mission tasks ($\mathcal{R}_{MC}$). This is because the DRL agent learns the opponent agent's pattern and uses the optimal signal strength for HDs to effectively lure the attacker while protecting critical MDs.

Figs. 6c and 6b are well aligned. Greater attack success yields fewer active MDs, leading to lower energy consumption. Notice that A3C-CD has a zero $\mathcal{N}_{AS}$ due to its resource limitation mechanism, preventing MDs from being compromised by DoS attacks. Although using CD allows for greater MD survivability, it does not maintain

a high mission completion ratio $\mathcal{R}_{MC}$ (as shown in Fig. 6a). This is because restricting computing resources with CD can lead to mission suspension, lowering $\mathcal{R}_{MC}$. Also, if more active MDs are present, more energy is consumed ($\mathcal{EC}$), as Fig. 6b shows.

We consider A3C-No-Defense as the baseline with no defense strategy employed in the mission system. IDS improves the mission system security compared to the baseline by reducing the number of successful DoS attacks. However, since IDS gives the same weight to all drones, it cannot prioritize the most critical drones, resulting in a lower mission completion rate ($\mathcal{R}_{MC}$) compared to A3C-A3C-HD.



(a) Ratio of completed mission tasks ($\mathcal{R}_{MC}$)

(b) Energy consumption ($\mathcal{EC}$)

(c) Number of attack successes ($\mathcal{N}_{AS}$)

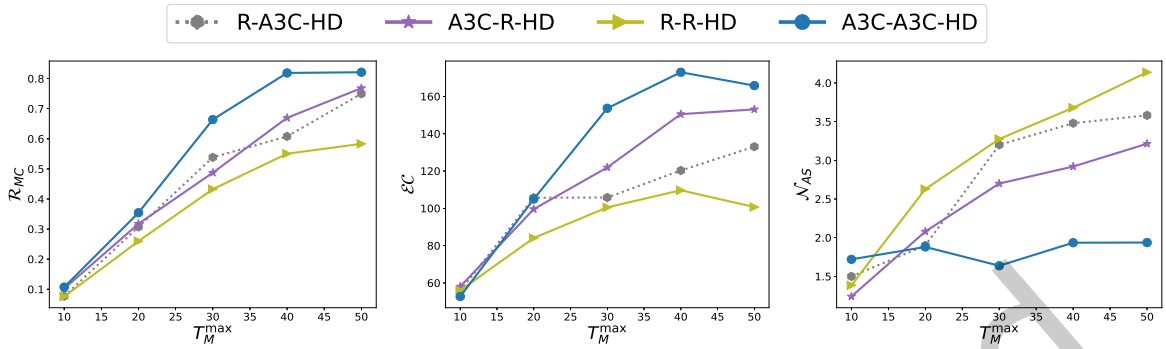
Fig. 6. Performance comparison of the four defenses against DoS attacks where our approach is A3C-A3C-HD, a baseline model A3C-A3C-No-HD, and two state-of-the-art competitive defenses, A3C-IDS and A3C-CD with respect to varying attack severity in terms of the maximum number of drones to attack in one round (ζ).

7.3 Ablation Study of HD-based Defenses

We evaluate mission performance with HDs, considering four schemes (R-R-HD, R-A3C-HD, A3C-R-HD, and A3C-A3C-HD) under different maximum mission duration thresholds, $T_M^{\max}$, and maximum attack targets, ζ.

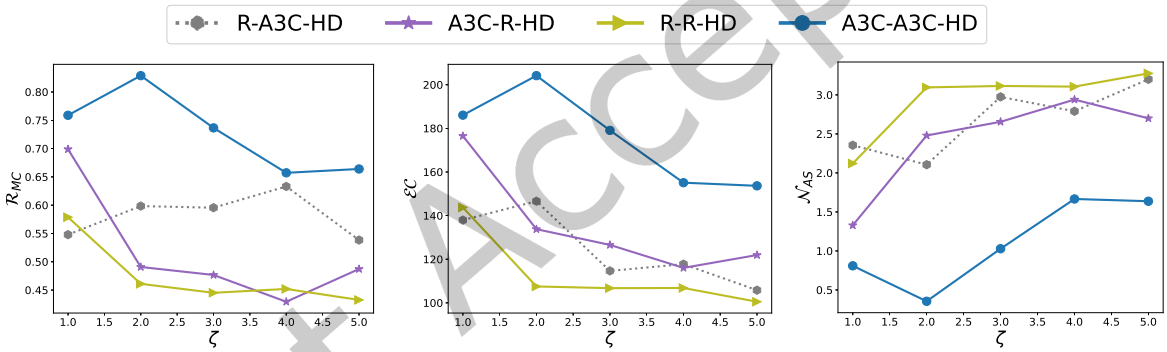
7.3.1 Effect of Maximum Mission Duration. Fig. 7 reports the influence of maximum mission duration ($T_M^{\max}$) on the mission completion ratio $\mathcal{R}_{MC}$, energy consumption $\mathcal{EC}$, and the number of successful attacks $\mathcal{N}_{AS}$. Fig. 7a demonstrates that A3C-A3C-HD beats A3C-R-HD, which beats R-A3C-HD. This indicates that DRL can detect an opponent's tactics more effectively when the opponent selects strategies with a clear pattern based on DRL rather than using random strategies. More time allows the exploration and scanning of larger areas: thus, as $T_M^{\max}$ increases, the completion mission ratio increases for all schemes. Fig. 7b reveals increasing energy consumption as more mission drones (MDs) are available for longer missions. Energy consumption aligns with increasing completion since they both increase with time. Consequently, a team achieves higher completion when it consumes more energy by using more drones. Fig. 7c supports the effectiveness of our approach against DRL-based attacks. Our approach becomes more effective when $T_M^{\max} > 20$, and the gap widens for a longer maximum duration, suggesting that longer missions provide more opportunities for DoS attacks that other schemes fail to defend against.

7.3.2 Effect of Varying Attack Budget. Fig. 8 reports the impact of varying the attack budget, i.e., the maximum number of drones targeted in one round (ζ), on the three metrics, $\mathcal{R}_{MC}$, $\mathcal{EC}$, and $\mathcal{N}_{AS}$. Fig. 8a shows that mission



(a) Ratio of completed mission tasks ($\mathcal{R}_{MC}$) (b) Energy consumption ($\mathcal{EC}$) (c) Number of attack successes ($\mathcal{N}_{AS}$)

Fig. 7. Comparative Performance Analysis: Ratio of completed mission tasks ($\mathcal{R}_{MC}$), total energy consumed ($\mathcal{EC}$), and number of attack successes ($\mathcal{N}_{AS}$) when maximum mission duration ($T_M^{\max}$) varies.



(a) Ratio of completed mission tasks ($\mathcal{R}_{MC}$) (b) Energy consumption ($\mathcal{EC}$) (c) Number of attack success ($\mathcal{N}_{AS}$)

Fig. 8. Comparative Performance Analysis: Ratio of completed mission tasks ($\mathcal{R}_{MC}$), total energy consumed ($\mathcal{EC}$), and the number of attack successes ($\mathcal{N}_{AS}$) when the maximum number of drones to attack in one round (ζ) varies.

completion falls as the attack budget increases, in essence, because fewer drones are available for the mission. Under A3C-A3C-HD, the defender performs the best when ζ is smaller (i.e., target fewer drones). However, with higher ζ, the attack strength becomes stronger, and accordingly, $\mathcal{R}_{MC}$ is significantly lowered. That is, more MDs are compromised, leading to a decrease in the number of active MDs available for completing mission tasks and resulting in a lower $\mathcal{R}_{MC}$. In Fig. 8b, we investigate the effect of ζ on energy consumption ($\mathcal{EC}$). We observe that the general trends in energy consumption under the four cases are very well aligned with the performance in $\mathcal{R}_{MC}$. Recall that the defender does not mainly aim to minimize energy consumption. Rather, the defender mainly aims to ensure more active, non-compromised nodes are connected in the mission team network and maximize mission performance while the adjusted signal strengths can introduce saving energy. Hence, higher

$\mathcal{R}_{MC}$ means more MDs available in the system, and the mission team naturally consumes more energy. Lower energy consumption with higher ζ is because higher ζ can quickly compromise MDs and lead to fewer drones consuming energy. In Fig. 8c, we study the effect of ζ on the number of attack successes. Under A3C-A3C-HD, the defender significantly outperforms and shows the lowest number of attack successes. The reason is that DRL-based defense can better detect highly intelligent attacks (e.g., using A3C) rather than random attacks. If the attack severity with ζ does not exceed two, which is manageable by the defender with five MDs and two HDs, the mission system can handle the DoS attacks effectively. In particular, it is noticeable that under A3C-A3C-HD, the defender performs very well for $\zeta \leq 3$.

8 Conclusion & Future Work

In this work, we developed a honey drone-based surveillance mission system that effectively executes a given mission while safeguarding mission drones by luring Denial-of-Service (DoS) attacks towards honey drones. Through extensive experiments using a physical engine-based realistic testbed, we validated the effectiveness of honey drones with dynamic signal strength levels as proposed. The system leverages deep reinforcement learning (DRL) to allow both the attacker and defender to autonomously determine optimal strategies. Specifically, we employed the Asynchronous Advantage Actor-Critic (A3C) algorithm, which provides parallel processing capabilities that expedite the learning process.

8.1 Key Findings

Our findings indicate that the proposed approach is highly effective in both luring opponents and protecting the drone fleet from DoS attacks. First, our proposed approach is highly effective in luring opponents and protecting the drone fleet from DoS attacks. This technique using HDs and DRL-based dynamic signal selection outperforms the existing techniques of IDS and ContainedDrone (CD) in completed mission ratio and attack success rate. Second, using HDs with DRL-based strategy selection in the mission system significantly increases mission performance by minimizing security vulnerabilities to DoS attacks. Third, when both the attacker and defender utilize DRL algorithms like A3C, the results show a marked improvement over non-DRL counterparts, especially in accumulated rewards for each agent. DRL-based agents were notably more adept at detecting an opponent's patterns when the opponent employed intelligent strategies like A3C, compared to random strategies. Fourth, DRL-based defenses using honey drones were found to outperform non-DRL-based methods, particularly in longer mission durations. These defenses were especially effective when the number of drones targeted by the attacker did not exceed half of the total drones in the mission system, resulting in fewer attack successes. Finally, the mission system's energy consumption was heavily influenced by the number of drones available for mission execution. However, the use of A3C in defense strategies offered a slight advantage in reducing energy consumption while maintaining high mission performance.

8.2 Limitation & Future Work

Despite the promising results, the current work has several limitations that can be addressed in future research. While we considered a single attacker using DoS attacks, real-world UAV-based surveillance mission systems may encounter multiple attackers employing various types of cyberattacks. The effectiveness of the honey drone technique against a diverse range of threats remains unexplored, and future research should consider scenarios involving multiple attackers and various cyberattack methods. This approach would enable a comprehensive evaluation of the honey drone technique's upper limits and help identify optimal deployment settings in more complex and realistic threat landscapes.

Another limitation of the current work is the interpretability of the optimal strategies identified using DRL. While the DRL approach successfully determines effective strategies, the rationale behind these strategies is not

clearly explained. Understanding the factors that contribute to a particular strategy's effectiveness is crucial for trust and adoption in real-world applications. Future work should incorporate Explainable AI (XAI) techniques to provide clearer insights into the utility of the identified strategies. XAI methods could uncover the key features and decision-making processes that lead to the selection of specific strategies, thereby enhancing the interpretability and trustworthiness of the honey drone system.

Furthermore, while this work focuses on using DRL (A3C) for autonomous strategy selection by both the attacker and defender, it does not explore the potential of game theory for strategy selection. Game theory could offer a more intuitive understanding of the attacker-defender dynamics and help identify equilibrium strategies. Future research should investigate the performance of game theory-based approaches compared to DRL to evaluate how honey drones can further enhance system performance and security in various environments. This comparison could provide valuable insights into the strengths and weaknesses of each approach, helping to determine the most suitable method for different scenarios.

Lastly, the current evaluation relies on a physical engine-based realistic testbed, which, while providing valuable insights, may not fully capture the complexities and challenges of real-world deployments. Future work should involve real-world testbed evaluations of the honey drone system to validate its efficacy and practical feasibility. Such evaluations should consider environmental conditions, hardware limitations, and communication constraints to ensure the system's robustness and reliability in real-world scenarios. These evaluations could also identify potential implementation challenges and guide further improvements to the honey drone system.

By addressing these limitations, future research aims to develop a more comprehensive and robust honey drone-based surveillance mission system that can effectively handle various security threats and operate efficiently in real-world environments. The insights gained from these future works will contribute to the advancement of UAV security and the development of more secure and reliable UAV-based applications.

References

[1] Yong Zeng, Rui Zhang, and Teng Joon Lim. 2016. Wireless communications with unmanned aerial vehicles: Opportunities and challenges. *IEEE Communications Magazine* 54, 5 (2016), 36–42.

[2] Lav Gupta, Raj Jain, and Gabor Vaszkun. 2015. Survey of important issues in UAV communication networks. *IEEE Communications Surveys & Tutorials* 18, 2 (2015), 1123–1152.

[3] C Stöcker, F Nex, M Koeva, and M Gerke. 2019. UAV-based Cadasral mapping: an assessment of the impact of flight parameters and ground truth measurements on the absolute accuracy of derived orthoimages. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences* XLII-2/W13 (2019), 613–617.

[4] Raja Naeem Akram, Konstantinos Markantonakis, Keith Mayes, Oussama Habachi, Damien Sauveron, Andreas Steyven, and Serge Chaumette. 2017. Security, privacy and safety evaluation of dynamic and static fleets of drones. In *The 36th Digital Avionics Systems Conference (DASC)*. IEEE, 1–12.

[5] Muhammad Abrar, Ushna Ajmal, Ziyad M Almohaimeed, Xiang Gui, Rizwan Akram, and Roha Masroor. 2021. Energy efficient UAV-enabled mobile edge computing for IoT devices: A review. *IEEE Access* 9 (2021), 127779–127798.

[6] Maher Aljehani and Masahiro Inoue. 2019. Performance evaluation of multi-UAV system in post-disaster application: Validated by HITL simulator. *IEEE Access* 7 (2019), 64386–64400.

[7] M Anwar Ma'Sum, M Kholid Arrofi, Grafika Jati, Futuhal Arifin, M Nanda Kurniawan, Petrus Mursanto, and Wisnu Jatmiko. 2013. Simulation of intelligent unmanned aerial vehicle (UAV) for military surveillance. In *2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. IEEE, 161–166.

[8] Milan Erdelj, Enrico Natalizio, Kaushik R Chowdhury, and Ian F Akyildiz. 2017. Help from the sky: Leveraging UAVs for disaster management. *IEEE Pervasive Computing* 16, 1 (2017), 24–32.

[9] Fadi Al-Turjman. 2022. A novel approach for drones positioning in mission critical applications. *Transactions on Emerging Telecommunications Technologies* 33, 3 (2022), e3603.

[10] Cristiano Bonato Both, Joao Borges, Luan Gonçalves, Cleverson Nahum, Ciro Macedo, Aldebaro Klautau, and Kleber Cardoso. 2022. SYSTEM INTELLIGENCE FOR UAV-BASED MISSION CRITICAL SERVICES WITH CHALLENGING 5G/B5G CONNECTIVITY. (2022).

[11] Riham Altawy and Amr M. Youssef. 2016. Security, privacy, and safety aspects of civilian drones: A survey. *ACM Transactions on Cyber-Physical Systems* 1, 2 (2016), 1–25.

[12] Ahmad Y Javaid, Weiqing Sun, Vijay K Devabhaktuni, and Mansoor Alam. 2012. Cyber security threat analysis and modeling of an unmanned aerial vehicle system. In *Conference on Technologies for Homeland Security (HST)*. IEEE, Waltham, MA, USA, 585–590.
[13] South China Morning Post. 2021. (2021). https://www.youtube.com/watch?v=S15OsuW4eGE
[14] Zelin Wan, Jin-Hee Cho, Mu Zhu, Ahmed H Anwar, Charles Kamhoua, and Munindar Singh. 2023. Deception in Drone Surveillance Missions: Strategic vs. Learning Approaches. In *Proceedings of the Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*. 382–387.
[15] Jorg Daubert, Dhanasekar Boopalan, Max Mühlhäuser, and Emmanouil Vasilomanolakis. 2018. HoneyDrone: A medium-interaction unmanned aerial vehicle honeypot. In *NOMS: Network Operations and Management Symposium*. IEEE, Taiwan, China, 1–6.
[16] Ender Cetin, Cristina Barrado, Guillem Muñoz, Miquel Macias, and Enric Pastor. 2019. Drone navigation and avoidance of obstacles through deep reinforcement learning. In *The 38th Digital Avionics Systems Conference (DASC)*. IEEE, San Diego, CA, USA, 1–7.
[17] Ronglei Xie, Zhijun Meng, Lifeng Wang, Haochen Li, Kaipeng Wang, and Zhe Wu. 2021. Unmanned aerial vehicle path planning algorithm based on deep reinforcement learning in large-scale and dynamic environments. *IEEE Access* 9 (2021), 24884–24900.
[18] Gyeong Taek Lee and Chang Ouk Kim. 2020. Autonomous control of combat unmanned aerial vehicles to evade surface-to-air missiles using deep reinforcement learning. *IEEE Access* 8 (2020), 226724–226736.
[19] Wenhong Zhou, Zhihong Liu, Jie Li, Xin Xu, and Lincheng Shen. 2021. Multi-target tracking for unmanned aerial vehicle swarms using deep reinforcement learning. *Neurocomputing* 466 (2021), 285–297.
[20] Letian Jing, Xiangdong Jia, Yaping Lv, and Nini Wan. 2021. Maximizing the average secrecy rate for UAV-assisted MEC: A DRL method. In *2021 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. IEEE, 2514–2518.
[21] Yu Zhang, Zhiyu Mou, Feifei Gao, Jing Jiang, Ruijin Ding, and Zhu Han. 2020. UAV-enabled secure communications by multi-agent deep reinforcement learning. *IEEE Transactions on Vehicular Technology* 69, 10 (2020), 11599–11611.
[22] Yu Zhang, Zirui Zhuang, Feifei Gao, Jingyu Wang, and Zhu Han. 2020. Multi-agent deep reinforcement learning for secure UAV communications. In *2020 Wireless Communications and Networking Conference (WCNC)*. IEEE, 1–5.
[23] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning (ICML)*. PMLR, USA, 1928–1937.
[24] Yuntao Wang, Zhou Su, Abderrahim Benslimane, Qichao Xu, Minghui Dai, and Ruidong Li. 2023. Collaborative honeypot defense in uav networks: A learning-based game approach. *IEEE Transactions on Information Forensics and Security* (2023).
[25] Fangyuan Hou, Jian Sun, Qiuling Yang, and Zhonghua Pang. 2022. Deep reinforcement learning for optimal denial-of-service attacks scheduling. *Science China Information Sciences* 65, 6 (2022), 162201.
[26] Chenglin Yang, Adam Kortylewski, Cihang Xie, Yinzhi Cao, and Alan Yuille. 2020. Patchattack: A black-box texture-based attack with reinforcement learning. In *European Conference on Computer Vision*. Springer, 681–698.
[27] Hossein Mohammadi Rouzbahani, Hadis Karimipour, and Lei Lei. 2023. Multi-layer defense algorithm against deep reinforcement learning-based intruders in smart grids. *International Journal of Electrical Power & Energy Systems* 146 (2023), 108798.
[28] Jiyang Chen, Zhiwei Feng, Jen-Yang Wen, Bo Liu, and Lui Sha. 2019. A container-based DoS attack-resilient control framework for real-time UAV systems. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 1222–1227.
[29] Hichem Sedjelmaci, Sidi Mohammed Senouci, and Nirwan Ansari. 2017. A hierarchical detection and response system to enhance security against lethal cyberattacks in UAV networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48, 9 (2017), 1594–1606.
[30] Devaprakash Muniraj and Mazen Farhood. 2017. A framework for detection of sensor attacks on small unmanned aircraft systems. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 1189–1198.
[31] Said Ouiazzane, Malika Addou, and Fatimazahra Barramou. 2022. A multiagent and machine learning based denial of service intrusion detection system for drone networks. *Geospatial Intelligence: Applications and Future Trends* (2022), 51–65.
[32] Jean-Philippe Condomines, Ruohao Zhang, and Nicolas Larrieu. 2019. Network intrusion detection system for UAV ad-hoc communication: From methodology design to real test validation. *Ad Hoc Networks* 90 (2019), 101759.
[33] Khaista Rahman, Muhammad Adnan Aziz, Ahsan Ullah Kashif, and Tanweer Ahmad Cheema. 2022. Detection of Security Attacks Using Intrusion Detection System for UAV Networks: A Survey. In *Big Data Analytics and Computational Intelligence for Cybersecurity*. Springer, 109–123.
[34] Charan Gudla, Md Shohel Rana, and Andrew H Sung. 2018. Defense techniques against cyber attacks on unmanned aerial vehicles. In *Proceedings of the International Conference on Embedded Systems, Cyber-physical Systems, and Applications (ESCS)*. 110–116.
[35] Hichem Sedjelmaci, Sidi Mohammed Senouci, and Mohamed-Ayoub Messous. 2016. How to detect cyber-attacks in unmanned aerial vehicles network?. In *2016 Global Communications Conference (GLOBECOM)*. IEEE, 1–6.
[36] Geon-Hwan Kim, Jae-Choong Nam, Imtiaz Mahmud, and You-Ze Cho. 2016. Multi-drone control and network self-recovery for flying Ad Hoc Networks. In *2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, Vienna, Austria, 148–150.
[37] KR Chevli, PY Kim, AA Kagel, DW Moy, RS Pattay, RA Nichols, and AD Goldfinger. 2006. Blue force tracking network modeling and simulation. In *MILCOM 2006-2006 Military Communications Conference*. IEEE, Washington, DC, USA, 1–7.

[38] Inc Himax Technologies. 2015. HM01B0 Ultralow Power CIS. (2015). https://www.himax.com.tw/en/products/cmos-image-sensor/always-on-vision-sensors/hm01b0/ Accessed: 03-08-2022.
[39] Steffen Liebergeld, Matthias Lange, and Collin Mulliner. 2013. Nomadic honeypots: A novel concept for smartphone honeypots. In *Workshop on Mobile Security Technologies (MoST'13)*, Vol. 4. Citeseer, Germany, 1–4.
[40] Michael Hooper, Yifan Tian, Runxuan Zhou, Bin Cao, Adrian P Lauf, Lanier Watkins, William H Robinson, and Wlajimir Alexis. 2016. Securing commercial WiFi-based UAVs from common security attacks. In *MILCOM 2016-2016 Military Communications Conference*. IEEE, Baltimore, MD, USA, 1213–1218.
[41] Bitcraze. 2022. Bitcraze's Crazyflie 2.x nano-quadrotor. (2022). https://www.bitcraze.io, accessed: 03-08-2022.
[42] Shane Daniel Kent, Ryan Weideman, and Nicholas Kimball. 2018. Dynamic video streaming for nano quadcopters. (2018).
[43] Maria Alvarez Custodio. 2019. Autonomous Recharging System for Drones: Detection and Landing on the Charging Platform. (2019).
[44] Daniele Palossi, Nicky Zimmerman, Alessio Burrello, Francesco Conti, Hanna Müller, Luca Maria Gambardella, Luca Benini, Alessandro Giusti, and Jérôme Guzzi. 2021. Fully onboard AI-powered human-drone pose estimation on ultralow-power autonomous flying nano-UAVs. *IEEE Internet of Things Journal* 9, 3 (2021), 1913–1929.
[45] Azade Fotouhi, Haoran Qiang, Ming Ding, Mahbub Hassan, Lorenzo Galati Giordano, Adrian Garcia-Rodriguez, and Jinhong Yuan. 2019. Survey on UAV cellular communications: Practical aspects, standardization advancements, regulation, and security challenges. *IEEE Communications Surveys & Tutorials* 21, 4 (2019), 3417–3442.
[46] Mohamed Khalil Baazaoui, Ilef Ketata, Ahmed Fakhfakh, and Faouzi Derbel. 2023. Modeling of Packet Error Rate Distribution Based on Received Signal Strength Indications in OMNeT++ for Wake-Up Receivers. *Sensors* 23, 5 (2023), 2394.
[47] Anand Gachhadar, Ram Krishna Maharjan, Surendra Shrestha, Nanda Bikram Adhikari, Faizan Qamar, Syed Hussain Ali Kazmi, and Quang Ngoc Nguyen. 2023. Power Optimization in Multi-Tier Heterogeneous Networks Using Genetic Algorithm. *Electronics* 12, 8 (2023), 1795.
[48] Irene Joseph, Prasad B Honnavalli, and BR Charanraj. 2022. Detection of DoS attacks on Wi-Fi networks using IoT sensors. In *Sustainable Advanced Computing: Select Proceedings of ICSAC 2021*. Springer, 549–558.
[49] Weixing Xue, Weining Qiu, Xianghong Hua, and Kegen Yu. 2017. Improved Wi-Fi RSSI measurement for indoor localization. *IEEE Sensors Journal* 17, 7 (2017), 2224–2230.
[50] Martin Sauter. 2010. *From GSM to LTE: An introduction to mobile networks and mobile broadband.* John Wiley & Sons.
[51] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems* 32 (2019), 1–12.
[52] Jacopo Panerati, Hehui Zheng, SiQi Zhou, James Xu, Amanda Prorok, and Angela P Schoellig. 2021. Learning to fly—a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control. *2021 International Conference on Intelligent Robots and Systems (IROS)* 0 (2021), 7512–7519.
[53] Erwin Coumans and Yunfei Bai. 2016. Pybullet, a Python module for physics simulation for games, robotics and machine learning. (2016).
[54] Tinkercliffs, ARC's flagship resource. ([n. d.]). https://www.docs.arc.vt.edu/resources/compute/00tinkercliffs.html
[55] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2623–2631.

Received 14 March 2024; revised 25 August 2024; accepted 26 September 2024