Intrusion Detection by Border Surveillance using Swarm Robotics

1st Inayat Kaur
Computer Science and Engineering
Indian Institute of Technology
Ropar, India
2020csb1088@iitrpr.ac.in

2nd Kushal Agrawal
Computer Science and Engineering
Indian Institute of Technology
Ropar, India
2020csb1096@iitrpr.ac.in

3rd Pooja Goyal
Computer Science and Engineering
Indian Institute of Technology
Ropar, India
2020csb1108@iitrpr.ac.in

***Abstract*—In the realm of national security, the need for robust border surveillance and the detection of unmanned aerial vehicle (UAV) intrusions is paramount. Traditional surveillance methods face challenges in efficiently covering large border areas. Recent advancements in robotics and artificial intelligence (AI) present an opportunity to address these challenges innovatively.**

In this paper, we propose a mechanism to detect intrusion at the border by using multiple UAVs. They are trained using Reinforcement learning based method to independently tackle path finding tasks.

***Index Terms*—UAV, Surveillance, Reinforcement Learning**

I. Introduction

Unmanned Aerial Vehicles (UAVs), colloquially known as drones, have witnessed a transformative evolution in recent years, emerging as indispensable tools in diverse domains such as surveillance, reconnaissance, and environmental monitoring. As these UAVs become increasingly integrated into our daily lives, the need for sophisticated deployment and coordination strategies becomes paramount, particularly in dynamic and unpredictable environments.

Traditional coverage path planning algorithms have proven effective in guiding UAVs through predefined routes, ensuring comprehensive exploration of a given area. However, these algorithms face significant challenges when confronted with dynamic environments where priorities shift, communication is non-deterministic, and the same location may require repeated visits based on evolving circumstances.

This paper addresses the challenge of deploying and coordinating UAVs in a grid-based environment where each cell is associated with a timer representing its priority. The dynamic nature of this environment, coupled with the limited lifetime and communication constraints of UAVs, necessitates a departure from conventional path coverage approaches. The objective is to develop an algorithm that enables UAVs to adapt autonomously to changing cell priorities, triggered by suspicious activities detected in the environment, investigate communication dynamics in a non-deterministic setting, where UAVs may experience loss of communication and must make decisions independently and incorporate the finite lifetime of UAVs into the algorithm, assessing the impact of battery drainage rates on the overall performance and mission duration.

The outcomes of this research have implications not only for the field of UAV coordination but also for real-world applications such as surveillance, security, and disaster response. As UAVs take on increasingly complex tasks, the ability to navigate dynamic environments with autonomy and adaptability becomes a critical aspect of their operational effectiveness.

This term paper unfolds in a structured manner, beginning with an in-depth exploration of the problem statement, followed by a detailed presentation of the proposed algorithm. The subsequent sections delve into the experimentation methodology, results, and a thoughtful discussion of the implications and future directions of the research.

In navigating the intricacies of UAV deployment in dynamic environments, this term paper contributes to the growing body of knowledge propelling the field of autonomous systems into new frontiers.

II. Background & Related Works

Reference [1] explore the use of swarm robotics for distributed surveillance. The collaborative behavior of a swarm of robots offers advantages in adaptability and coverage.The paper primarily emphasizes on the application of computer vision techniques rather than the path planning of robots. Therefore, we have aligned our research objectives to concentrate specifically on enhancing the path planning aspects of the robots. Various path planning algorithms are investigated for optimizing the trajectories of autonomous robots in surveillance scenarios.

The problem is different from the traditional coverage path problems because of the following reasons:

1) Path coverage algorithms are designed for scenarios where a robot or UAV needs to visit each cell in the environment exactly once. In the UAV intrusion detection problem, the same cell may need to be visited multiple times based on changes in priority. Path coverage algorithms are not naturally suited to handle repeated visits to the same cell.
2) Path coverage algorithms typically generate a fixed path for the robot or UAV, and they do not dynamically adapt to changes in the environment, such as priority changes in specific cells. In the UAV intrusion detection problem,

adapting to dynamic changes in cell priorities is crucial for effective surveillance.

3) The environment in the UAV intrusion detection problem is non-deterministic due to factors like communication loss and changes in suspicion levels. Path coverage algorithms may struggle to handle such uncertainties and may not be able to adjust their paths in real-time based on dynamic events.

Hence, we use reinforcement learning based Deep Q-Learning and compared it with traditional algorithms like Hill Climbing.

III. Methodology

A. Mathematical Modelling

Consider a discrete-time representation of the environment, represented as a grid with cells. Each cell in the grid is associated with a timer that decreases continuously, reflecting the priority of the cell. The objective is to design a systematic approach to deploy and coordinate n Unmanned Aerial Vehicles (UAVs) in a way that each cell is visited by at least one UAV before its associated timer reaches zero.

1) Some of the cells in the grid cannot be visited, representing obstacles in the environment.
2) Since some of the UAVs can go out of communication, the number of available UAVs at a time t is variable.
3) Each of the UAV has a limited lifetime. Its battery gets drained continuously. This also affects the number of availble UAVs at each time.
4) If a suspicious activity is found at a cell, the priority of the cell is increased. The UAVs should adapt to the changing priorities. The suspicious activity is modelled using a guassian distribution.
5) Each cell can have only one UAV at a time.
6) It is possible that due to wind or other factors, UAV is not able to move in the expected direction. To model this, we give probability of 0.9 to move in the expected direction and 0.1 for random move.

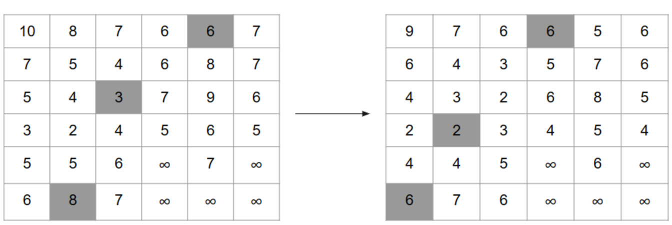

Fig. 1. Grid configuration change after After 1 step

The objective is to design an autonomous network using these robots to optimize the coverage of cells within a specified time frame while considering the priority of each cell. We use the total lateness metric given in (1) for the same.

$$TotalLateness = \begin{cases} 0 & \text{if value} \geq 0 \\ |\text{value}| & \text{else} \end{cases} \tag{1}$$

B. Proposed Solution

To address the intricate challenges posed by dynamic environments, non-deterministic communication, and shifting priorities, our proposed solution leverages a sophisticated algorithm based on Deep Q-Learning (DQN). DQN, a form of reinforcement learning, enables Unmanned Aerial Vehicles (UAVs) to learn and adapt their strategies over time by interacting with the environment.

DQN combines Q-learning, a traditional reinforcement learning technique, with deep neural networks to handle high-dimensional state spaces. In our context, the state space encapsulates the spatial and temporal information of the environment, including cell priorities, UAV positions, and the number of available UAVs at a time.

The core of the DQN algorithm lies in its neural network, which approximates the Q-function. Our implementation employs a deep neural network with multiple layers, including input, hidden, and output layers. The input layer receives a vectorized representation of the state, and the output layer produces Q-values corresponding to different actions. The model structure is given in Fig. 2.

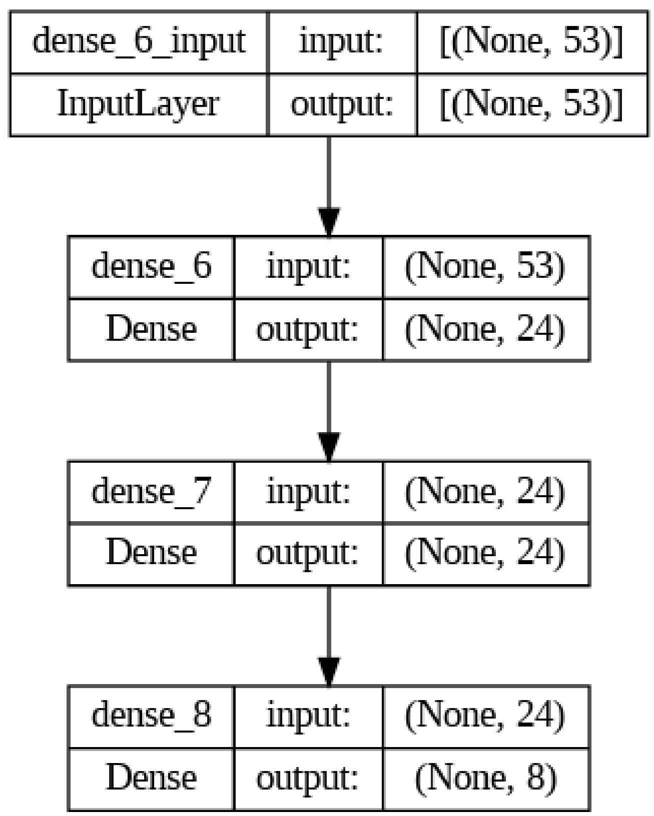

Fig. 2. Neural Network structure used in DQN

A critical aspect of the algorithm is the formation of states, representing the current snapshot of the environment. Each state includes information such as UAV positions, the number of available UAVs at a time, and the lateness associated with different zones in which the whole grid is divided. This holistic representation allows the UAVs to make informed decisions based on the evolving dynamics of the environment.

The algorithm incorporates a mechanism for UAVs to adapt autonomously to changing cell priorities. When suspicious activity is detected, the base station broadcasts updated priorities, triggering the UAVs to reevaluate their actions. The DQN algorithm facilitates efficient learning and decision-making, ensuring timely responses to shifting priorities.

In a non-deterministic communication setting, where UAVs may experience loss of communication, the algorithm enables each UAV to make decisions independently based on its local observations and learned policies. This decentralized approach enhances the robustness of the system, allowing UAVs to navigate the environment even in the absence of global communication.

The finite lifetime of UAVs is a crucial factor in mission planning. The algorithm accounts for battery drainage rates, influencing the UAVs' decision-making processes. This consideration ensures that the deployment strategy aligns with the available energy resources, optimizing the overall mission duration.

By combining the power of DQN with a tailored neural network structure and a comprehensive state representation, our proposed solution provides a versatile framework for UAV coordination in dynamic and priority-sensitive environments.

A hill climbing algorithm is also implemented for comparative analysis. Here, the UAVs move greedily to the cell which has the highest priority among the neighbouring cells.

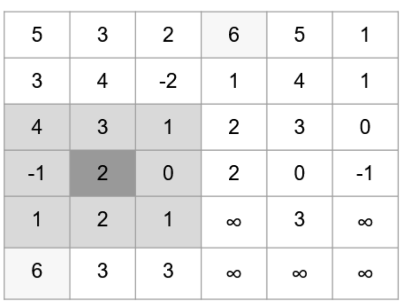

Fig. 3. Hill Climbing Algorithm Representation

C. Experimentation

The experimentation phase involved the implementation of a simulation environment to model the proposed algorithm for deploying and coordinating Unmanned Aerial Vehicles (UAVs) in a grid-based environment. The simulation was designed to address the challenges posed by changing cell priorities, communication losses, limited drone lifetime, and the dynamic nature of suspicious activity detection.

The simulation was implemented using a modular structure with key classes representing drones, the base station, the environment, a trainer for reinforcement learning, and a global simulator clock. Each class encapsulated specific functionalities, contributing to a cohesive and extensible codebase.

Drone Class: The Drone class represented individual UAVs in the simulation. Each drone was equipped with Q-learning for algorithm decision-making, information about cell priorities, current position, last communication time, and battery levels. The drone class interacted with the environment, made decisions on movement, and adapted to changing cell priorities.

BaseStation Class: The BaseStation class served as the central coordination hub. It managed cell priorities, received data about robot movements and suspicious locations, operated the trainer for reinforcement learning, and made decisions on flying new drones. The base station also broadcasted updates on cell priorities.

Environment Class: The Environment class simulated the grid-based environment. It tracked flying drones, generated suspicious locations, and managed drone addition and removal based on battery levels. The suspicion value at a location follows a normal curve with mean 0 and standard deviation 5 i.e. N(0,5). These values are then mapped to some priority value for the counters.

Trainer Class: The Trainer class handled the training aspect of the simulation. It interacted with the DQNAgent for reinforcement learning, received data from the environment and base station, and trained drones to make informed decisions.

Simulator Class: The Simulator class orchestrated the entire simulation. It managed the global clock, controlled the simulation flow, and facilitated communication between different components. The simulation involved a warm-up phase, a main simulation loop, training intervals, and data collection.

IV. Results

We conduct simulations based on 200 simulation steps for both Hill Climbing and DQN algorithms. Additionally, DQN was subjected to 300 warm-up steps. Each step involves one move by each drone. The results corresponding to different grid sizes and number of robots are summarized in Table I.

TABLE I
COMPARISON OF LATENESS BETWEEN ALGORITHMS

Lateness Table	**Algorithms**	
Configurations	***DQN***	***Hill Climbing***
Area=50x50, n=15	402361	454553
Area=100x50, n=35	879783	915033
Area=200x100, n=50	3647908	3672569

Overall, the results indicate that the DQN algorithm consistently outperformed Hill Climbing in terms of lateness across various simulation and warmup scenarios. DQN demonstrated lower lateness values, suggesting its effectiveness in optimizing the system compared to Hill Climbing.

It is important to note that the observed efficiency of DQN in minimizing lateness could potentially be further enhanced through additional training iterations. Increasing the number of training steps may lead to a more fine-tuned model, resulting in even lower lateness values and improved overall system performance. This highlights the adaptability of the DQN algorithm and its potential for continued refinement to achieve optimal results in latency-sensitive applications.

V. Conclusion

The term paper delved into the intricate realm of deploying and coordinating Unmanned Aerial Vehicles (UAVs) in a dynamic grid-based environment. Addressing the unique challenges posed by the problem statement, such as the need for repeated visits to the same cell based on priority, required a departure from conventional path coverage algorithms. The algorithm's success in handling this distinct characteristic sets it apart in the realm of coverage path planning. In the rapidly evolving landscape of UAV technology, the presented algorithm contributes a valuable perspective on the challenges associated with dynamic environments. This term paper, while providing insights into the proposed algorithm's efficacy, serves as a stepping stone for future endeavors in the exciting intersection of UAVs, dynamic environments, and autonomous decision-making.

VI. Future Work

We would like to further improvise on the neural network by using hyperparameter tuning. This refinement process will allow us to fine-tune the model and extract more promising results. We would also like to compare our approach with some variations of deterministic path coverage algorithms to gauge the effectiveness and efficiency of our proposed method. Additionally, we aim to explore alternative reinforcement learning techniques and assess their applicability and performance in the context of drone path planning.

References

[1] W. Mahjoub, C. Nakkach and T. Ezzedine, "Design of Autonomous Wireless Sensor Network Using Mobile Robots for Intrusion Detection and Border Surveillance," 2023 International Wireless Communications and Mobile Computing (IWCMC), Marrakesh, Morocco, 2023, pp. 476-481, doi: 10.1109/IWCMC58020.2023.10183288.

[2] C. V. Mahamuni and Z. M. Jalauddin, "Intrusion Monitoring in Military Surveillance Applications using Wireless Sensor Networks (WSNs) with Deep Learning for Multiple Object Detection and Tracking," 2021 International Conference on Control, Automation, Power and Signal Processing (CAPS), Jabalpur, India, 2021, pp. 1-6, doi: 10.1109/CAPS52117.2021.9730647.

[3] A. I. Panov, K. S. Yakovlev, and R. Suvorov, "Grid path planning with deep reinforcement learning: Preliminary results," Procedia Comput. Sci., vol. 123, pp. 347–353, 2018.

[4] B. Jang, M. Kim, G. Harerimana and J. W. Kim, "Q-Learning Algorithms: A Comprehensive Classification and Applications," in IEEE Access, vol. 7, pp. 133653-133667, 2019, doi: 10.1109/ACCESS.2019.2941229.