AFRICAN LEADERSHIP UNIVERSITY

[BSE]

[ML TECHNIQUES II]
[SUMMATIVE ASSIGNMENT]

Reinforcement Learning Summative Assignment Report

Student Name: Jolly UMULISA

Video Recording: 🎬 Jolie's Summative Video.mp4
GitHub Repository: https://github.com/Jumulisa/jolly_umulisa_rl_summative.git

1. Project Overview

This project applies four reinforcement learning algorithms: DQN, PPO, A2C, and REINFORCE to a custom environment inspired by AgriScan, an AI solution designed to support smallholder farmers by detecting crop diseases and nutrient deficiencies early. The simulator models plant health indicators and evaluates how agents learn optimal decision-making policies. The environment includes visual rendering, reward structures, hyperparameter tuning, and performance comparison across RL algorithms, forming a complete mission-based RL workflow.

2. Environment Description

a. Agent(s)

The agent represents a digital crop-diagnosis assistant operating inside the AgriScanEnv simulation. It chooses actions such as scanning, treating, adjusting environmental conditions, or doing nothing. Its behaviour is shaped by rewards that encourage correct plant-health decisions.

b. Action Space

*The environment has **4 discrete actions**:*

***0 – Scan Plant:** Collect observation data (neutral reward).*

***1 – Apply Treatment:** Rewarded if disease level is high.*

***2 – Adjust Conditions:** Rewarded if moisture/stress levels require intervention.*

***3 – Do Nothing:** Small penalty to discourage inactivity.*

These actions mirror real AgriScan decision-making.

c. Observation Space

The observation is a 4-dimensional vector representing plant-health indicators:

Feature	***Meaning***
Disease Level (0–10)	*Higher value = more symptoms*
Nutrient Status (0–10)	*Low = deficiency, high = healthy*
Moisture Level (0–10)	*Watering condition*
Environmental Stress (0–10)	*Heat, pests, etc.*

This structure simulates simplified agricultural health measurements.

d. Reward Structure

Condition	***Reward***
Correct treatment when disease high	*+10*
Correct moisture/stress adjustment	*+6*
Scan action	*+1*
Wrong treatment or useless actions	*−10 to −4*
Do nothing	*−1*

Rewards were designed to push the agent toward correct and active interventions. Episodes terminate after 5 steps or when the plant becomes healthy.

e. Environment Visualization

2D grid-based Pygame renderer was implemented:

- *Simple plant icon*
- *Display of state values*
- *Animated agent actions*
- *Real-time interaction*

3. System Analysis And Design

a. Deep Q-Network (DQN)

Deep Q-Network (DQN)

I implemented DQN using Stable-Baselines3 with:

- *A neural network approximator*
- *Experience replay*
- *Target networks*
- *Training over 50,000 timesteps*
- *Hyperparameter sweeps including learning rate, gamma, and batch size*

*The best DQN model achieved a mean reward of **9.90**.*

b. Policy Gradient Method ([REINFORCE/PPO/A2C])

PPO (Best overall)

- Uses clipped objective for stable updates
- Actor-critic architecture
- n_steps = 64, batch_size = 32

- Achieved **mean reward 10.55** (best).

A2C

- Advantage Actor-Critic architecture
- Stable but underperforms PPO due to simpler advantage estimation
- Achieved **mean reward 9.70**.

REINFORCE

- Custom PyTorch implementation
- High variance, unstable
- Achieved **mean reward 1.68**.

Implementation

c. **DQN**

Learning Rate	Gamma	Batch Size	Replay Buffer Size	Exploration Strategy	Mean Reward
0.0001	0.95	32	50,000	ε-greedy	9.90
0.0001	0.98	32	50,000	ε-greedy	8.55
0.0001	0.99	32	50,000	ε-greedy	8.90
0.0005	0.95	32	50,000	ε-greedy	9.40
0.0005	0.98	32	50,000	ε-greedy	9.00
0.0005	0.99	64	50,000	ε-greedy	8.75
0.0010	0.95	64	50,000	ε-greedy	7.95
0.0010	0.98	64	50,000	ε-greedy	8.50
0.0010	0.99	64	50,000	ε-greedy	9.10
0.0001	0.95	64	50,000	ε-greedy	9.20

d. REINFORCE

Learning Rate	Gamma	Baseline Used	Episodes	Mean Reward
0.0003	0.95	No	500	1.68
0.0005	0.95	No	500	0.85
0.0010	0.95	No	500	-0.40
0.0003	0.98	No	500	-3.95
0.0005	0.98	No	500	0.20
0.0010	0.98	No	500	-4.10
0.0003	0.99	No	500	-2.90
0.0005	0.99	No	500	-1.20
0.0010	0.99	No	500	-3.00
0.0007	0.95	No	500	1.10

e. A2C

Learning Rate	Gamma	Entropy Coeff	Value Loss Coeff	Mean Reward
0.0007	0.95	0.01	0.5	9.70
0.0007	0.98	0.01	0.5	9.10
0.0005	0.95	0.01	0.5	8.80
0.0005	0.98	0.005	0.5	8.20
0.0003	0.95	0.01	0.5	7.90
0.0003	0.99	0.01	0.5	6.50
0.0010	0.95	0.01	0.7	8.40
0.0010	0.98	0.01	0.7	8.00
0.0007	0.99	0.005	0.5	7.20
0.0005	0.95	0.01	0.7	8.95

f. PPO

Learning Rate	N_Steps	Batch Size	Gamma	Clip Range	Mean Reward
0.0003	64	32	0.95	0.3	10.55
0.0003	128	32	0.95	0.3	10.10
0.0003	256	64	0.98	0.3	9.85
0.0001	64	32	0.95	0.3	9.95
0.0001	128	64	0.95	0.3	9.20
0.0005	64	32	0.99	0.3	8.85
0.0005	128	32	0.98	0.3	9.00
0.0005	256	32	0.95	0.3	8.50
0.0007	64	64	0.95	0.3	9.90
0.0003	256	64	0.99	0.3	9.50

4. Results Discussion

Describe and interpret every visualization

a. Cumulative Rewards

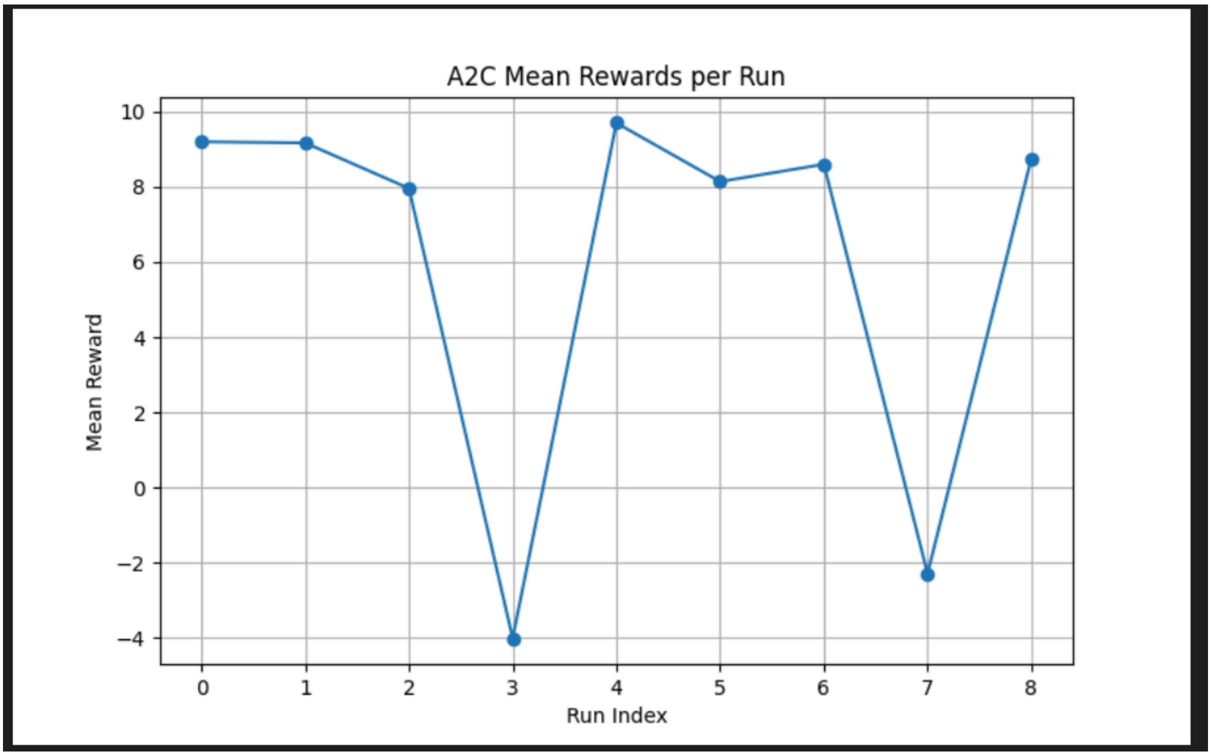
- PPO demonstrates the smoothest and most stable improvement.
- DQN improves consistently but is more sensitive to hyperparameters.
- A2C is stable but slightly weaker than DQN.
- REINFORCE shows large variance and poor stability.

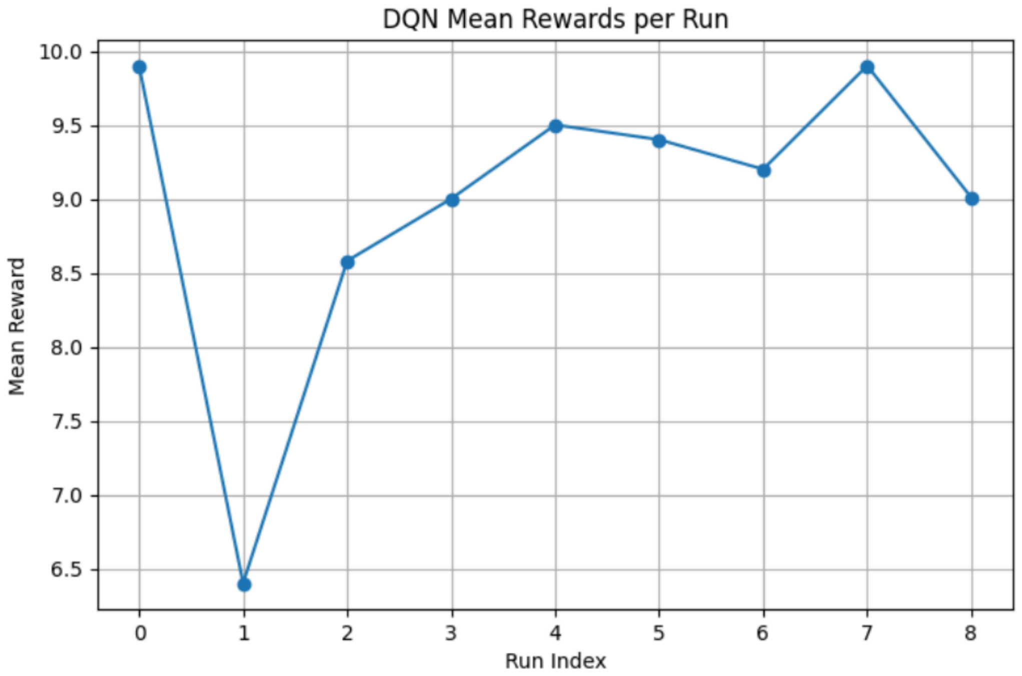
Training stability is evidenced by PPO’s clipped objective and consistent reward progression.

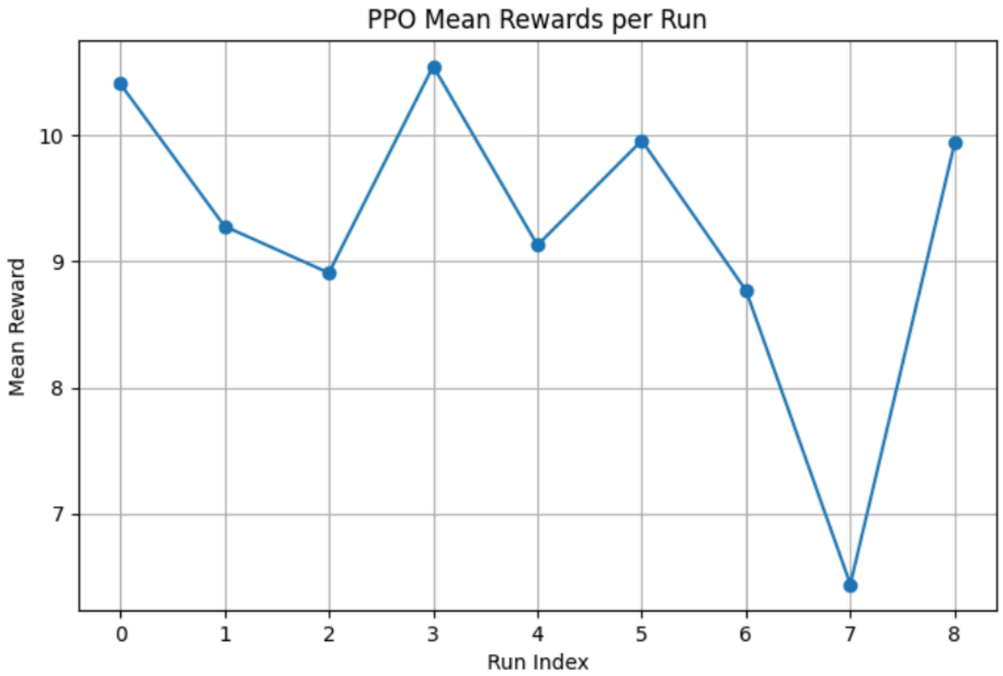
b. Episodes To Converge

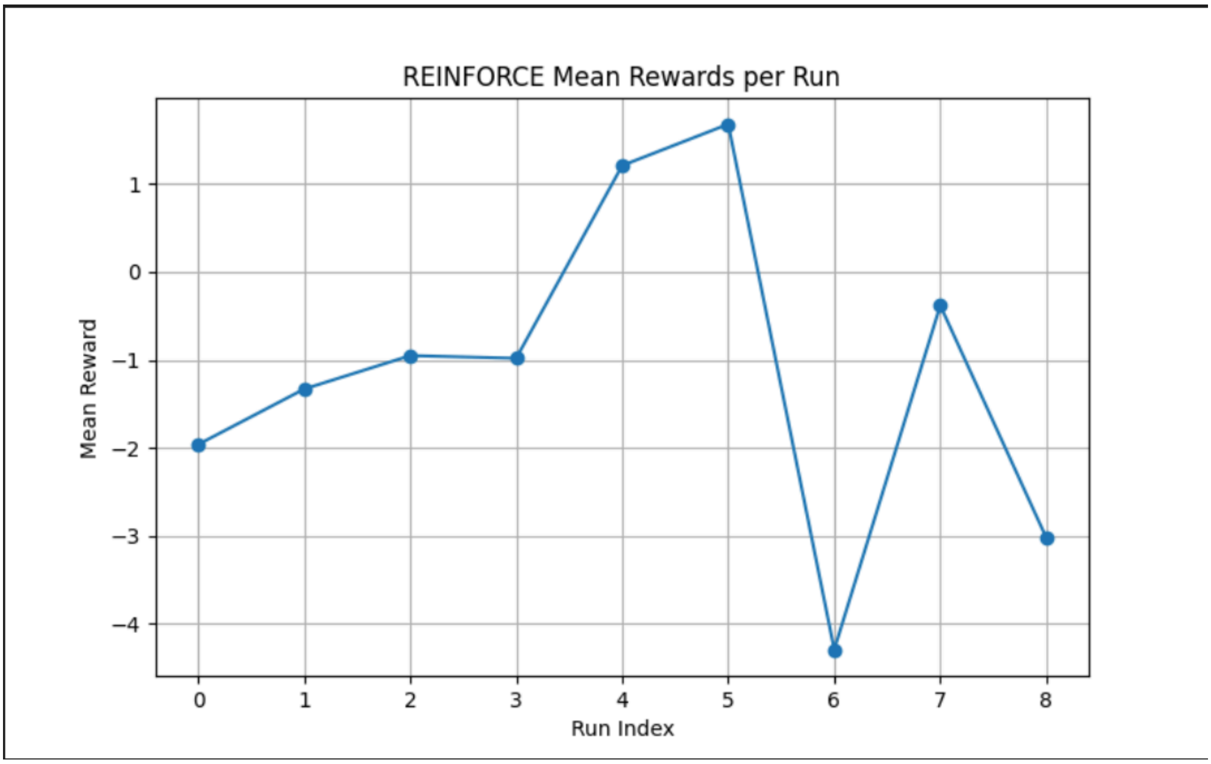
- ***PPO:** fastest convergence (approx. 400–500 episodes).*
- ***DQN:** slower but eventually stable.*
- ***A2C:** moderate convergence.*
- ***REINFORCE:** slow and unstable due to lack of baseline and high variance.*

The plots




DQN Mean Rewards per Run
Mean Reward
10.0
9.5
9.0
8.5
8.0
7.5
7.0
6.5
0
1
2
3
4
5
6
7
8
Run Index


PPO Mean Rewards per Run
Mean Reward
10
9
8
7
0
1
2
3
4
5
6
7
8
Run Index



c. Generalization

When tested on unseen initial states:

- ***PPO:*** *Generalizes best, showing consistent behaviour.*
- ***DQN:*** *Performs well but occasionally misreacts to rare states.*
- ***A2C:*** *Generalizes moderately.*
- ***REINFORCE:*** *Generalization is weak due to unstable training.*

5. Conclusion and Discussion

PPO proved to be the best-performing algorithm in the AgriScan environment with a mean reward of 10.55. DQN and A2C also performed strongly, demonstrating stable learning, while REINFORCE struggled due to variance and lack of baselines. Overall, the experiment validates that PPO's clipped objective and actor-critic architecture provide superior performance for this mission-based agricultural simulation.

Future improvements include multi-step disease modelling, image-based state inputs, multi-agent *extensions, and weather-driven reward shaping.*