Group Code - B25AH02

IoT Data Analytics Using DRL

Mentor - Dr.Abhishek Hazara

Our team

- Sushant Gadyal - S20220010218
- Bathina Santosh Kiran - S20220010035
- Dinesh Peddina - S20220010062

Content

Problem Statement

Our focus is to design a system where the Master UAV classifies whether to execute a task locally or offload it, and if offloading is chosen, uses Deep Reinforcement Learning (DRL) to select the optimal external device for task execution

Reduce the energy consumption and latency that is required for task offloading and computation.

Basic Diagram

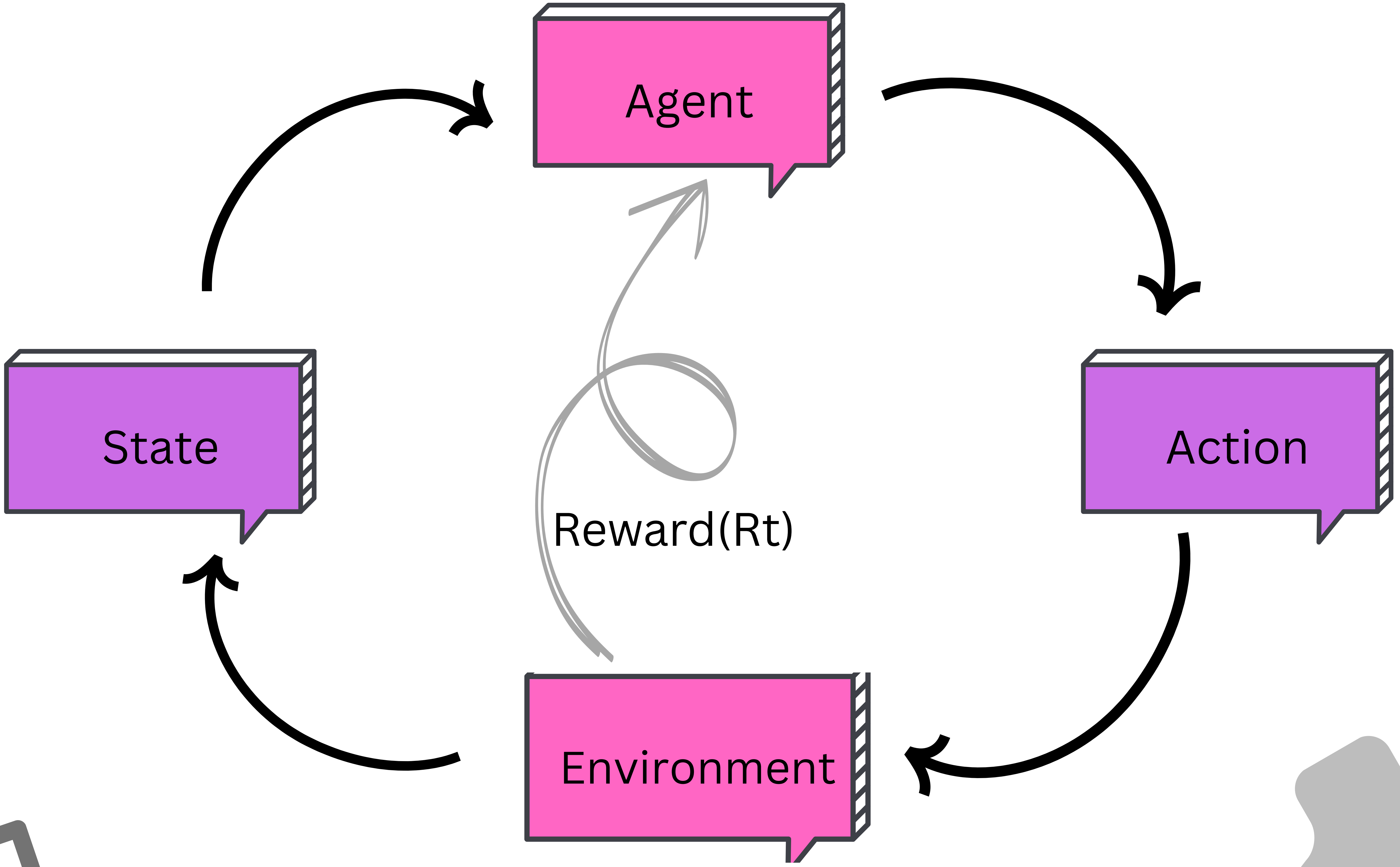
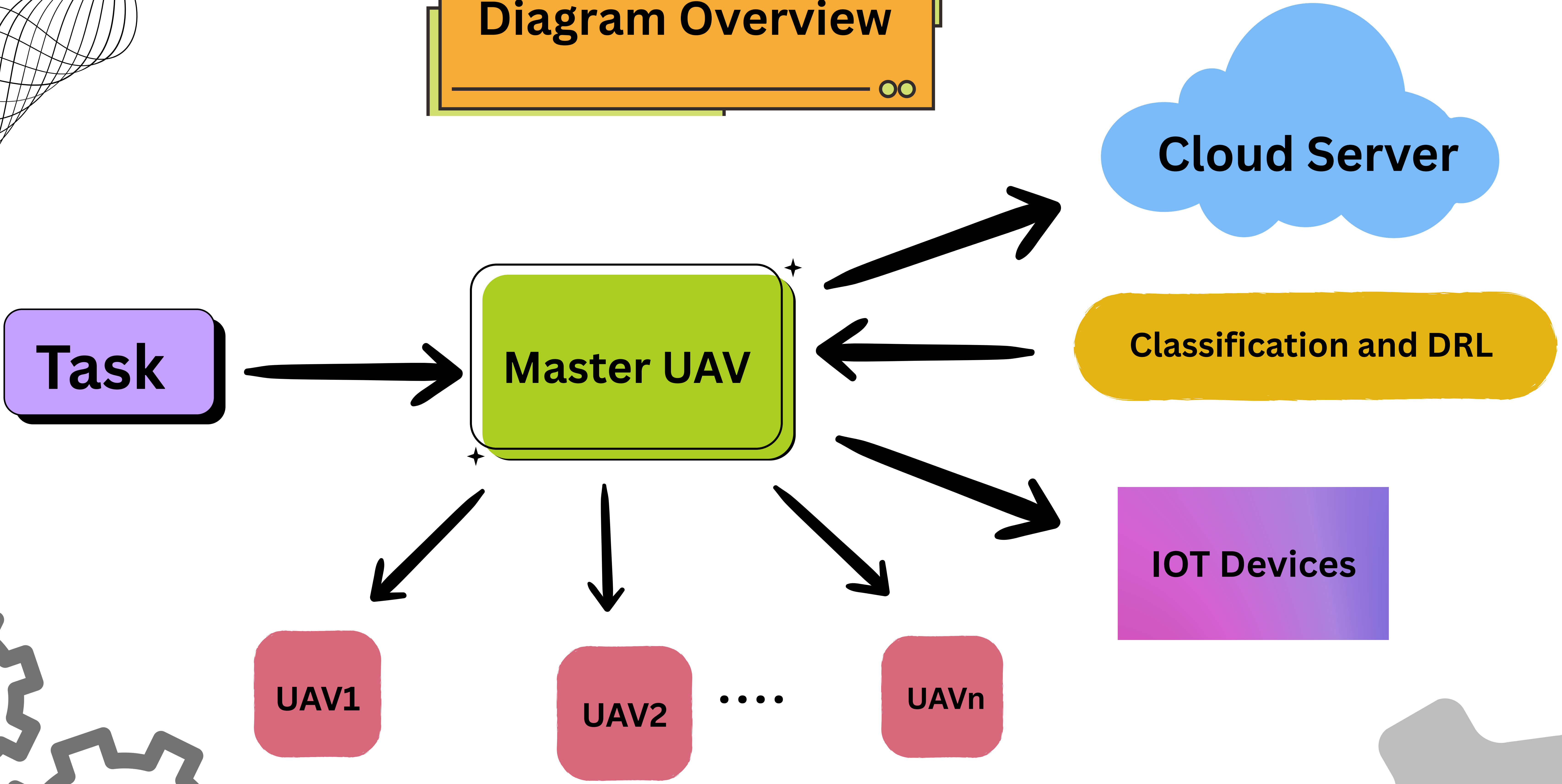
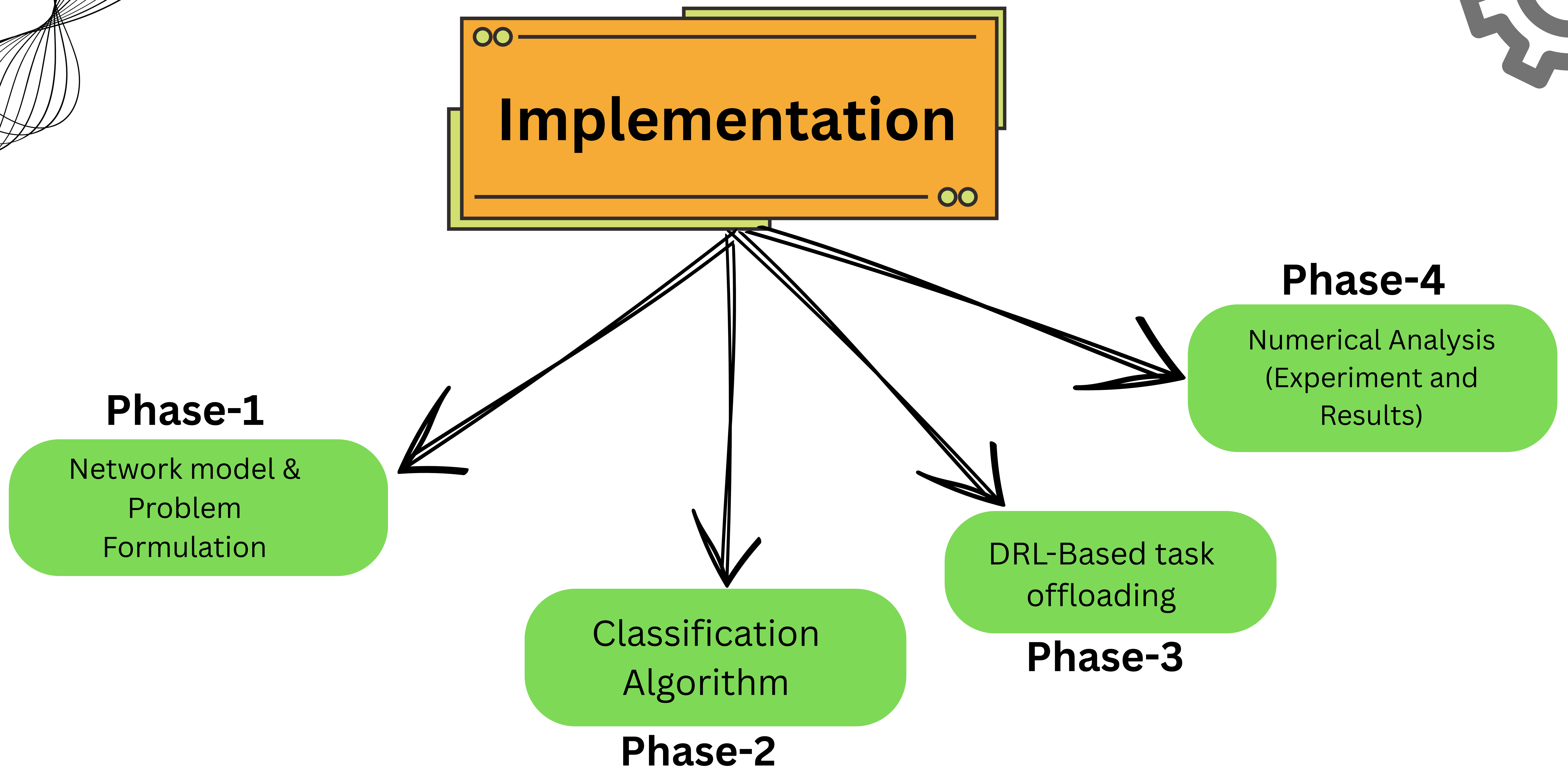


Diagram Overview




Implementation
Phase-1
Network model & Problem Formulation
Classification Algorithm
Phase-2
DRL-Based task offloading
Phase-3
Phase-4
Numerical Analysis (Experiment and Results)

Phase 1

Network Model and Problem Formulation

DRL Environment Setup

1. Master UAV
2. IoT Device
3. Slave UAV
4. Cloud Server

Generate Random Tasks

Task Execution Methods

1. Master UAV Execution
2. IoT Execution
3. Slave UAV Execution
4. Cloud Server Execution

Get State

Reset Environment

Phase 2

Classification Algorithm

- Compute Required Energy
- Compute Required Execution Time
- Check Battery Sufficiency
- Check Time Constraint
- Return Execution Feasibility (True/False)

Phase 3

DRL - Based Task Offloading

Task Partitioning

Deep Q - Learning (DQN)

Algorithm Implementation

Phase 3

DRL - Based Task Offloading

Steps	What DQN does?
Initialize the agent	The agent starts with no knowledge.
Take an Action	The agent offloads a task (Local, Master UAV, SlaveUAV, Cloud).
Recieve a Reward	Based on delay & energy consumption, the agent gets a negative reward.
Store Experience	The agent saves (state, action, reward, next_state).
Train the Neural Network	The agent learns which actions maximize reward.
Reduce Randomness	The agent shifts from random choices to smart decisions.
Improve over Time	After many episodes, the agent chooses the best action every time.

Phase 4

Numerical Analysis

Performance Metrics

Comparison with BenchMarks

Graphs and Analysis

Formulas Used

1. Local Master UAV Execution

$$\mathbb{T}_{i,m}^{\text{Master UAV}} = \frac{\Gamma_{i,m}\mathcal{O}_i\mathcal{D}_i}{\mathscr{C}_m^{\text{CPU}}}$$

$$\mathbb{E}_{i,m}^{\text{Master UAV}} = \Gamma_{i,m}\mathcal{O}_i\mathcal{D}_i k\left(\mathscr{C}_m^{\text{CPU}}\right)^2$$

2. Master UAV to IoT Device Execution

$$\mathbb{T}_{m,d}^{\text{upload}} = \frac{\Gamma_{i,d}\mathcal{O}_i}{\mathbb{R}_{m,d}}$$

$$\mathbb{E}_{m,d}^{\text{upload}} = \frac{\Gamma_{i,d}\mathcal{O}_i\mathscr{P}_m}{\mathbb{R}_{m,d}}$$

$$\mathbb{T}_{i,d}^{\text{IoT}} = \frac{\Gamma_{i,d}\mathcal{O}_i\mathcal{D}_i}{\mathscr{C}_d^{\text{CPU}}}$$

$$\mathbb{E}_{i,d}^{\text{IoT}} = \Gamma_{i,d}\mathcal{O}_i\mathcal{D}_i k\left(\mathscr{C}_d^{\text{CPU}}\right)^2$$

$$\mathbb{T}_i^{\text{IoT}} = \mathbb{T}_{m,u}^{\text{upload}} + \mathbb{T}_{i,u}^{\text{IoT}}$$

$$\mathbb{E}_i^{\text{IoT}} = \mathbb{E}_{m,d}^{\text{upload}} + \mathbb{E}_{i,d}^{IoT}$$

Formulas Used

3. Master UAV to UAV Execution

$$\mathbb{T}_{m,u}^{\text{upload}} = \frac{\Gamma_{i,u}\mathcal{O}_i}{\mathbb{R}_{m,u}}$$

$$\mathbb{E}_{m,u}^{\text{upload}} = \frac{\Gamma_{i,u}\mathcal{O}_i\mathscr{P}_m}{\mathbb{R}_{m,u}}$$

$$\mathbb{T}_{i,u}^{\text{UAV}} = \frac{\Gamma_{i,u}\mathcal{O}_i\mathcal{D}_i}{\mathscr{C}_u^{\text{CPU}}}$$

$$\mathbb{E}_{i,u}^{\text{UAV}} = \Gamma_{i,u}\mathcal{O}_i\mathcal{D}_i k \left(\mathscr{C}_u^{\text{CPU}}\right)^2$$

$$\mathbb{T}_i^{UAV} = \mathbb{T}_{m,u}^{\text{upload}} + \mathbb{T}_{i,u}^{\text{UAV}}$$

$$\mathbb{E}_i^{UAV} = \mathbb{E}_{m,u}^{\text{upload}} + \mathbb{E}_{i,u}^{\text{UAV}}$$

4. Master UAV to Cloud Device Execution

$$\mathbb{T}_{m,c}^{\text{upload}} = \frac{\Gamma_{i,c}\mathcal{O}_i}{\mathbb{R}_{m,c}}$$

$$\mathbb{E}_{m,c}^{\text{upload}} = \frac{\Gamma_{i,c}\mathcal{O}_i\mathscr{P}_m}{\mathbb{R}_{m,c}}$$

$$\mathbb{T}_{i,c}^{\text{Cloud}} = \frac{\Gamma_{i,c}\mathcal{O}_i\mathcal{D}_i}{\mathscr{C}_c^{\text{CPU}}}$$

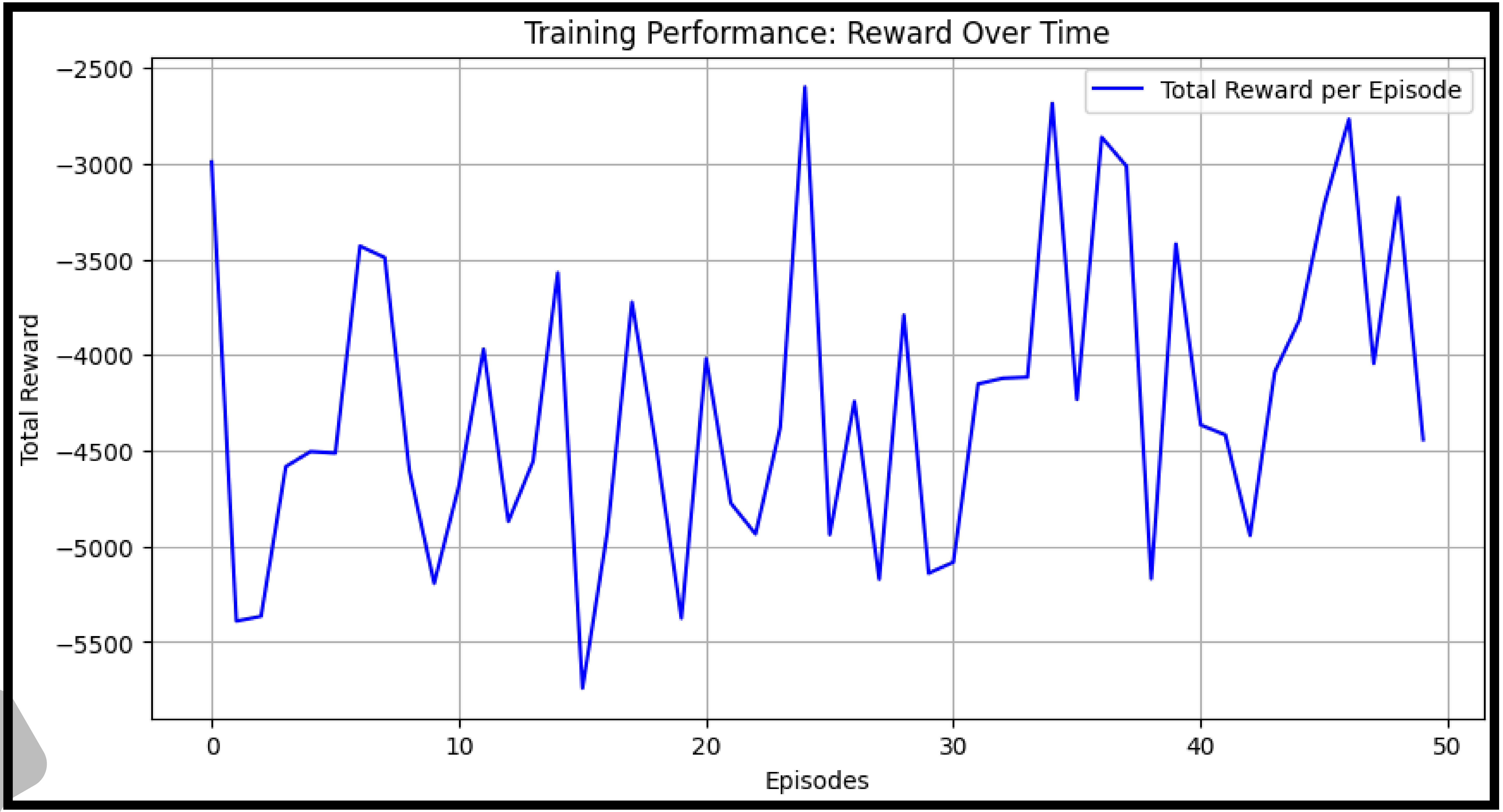
$$\mathbb{E}_{i,c}^{\text{Cloud}} = \Gamma_{i,c}\mathcal{O}_i\mathcal{D}_i k \left(\mathscr{C}_c^{\text{CPU}}\right)^2$$

$$\mathbb{T}_i^{Cloud} = \mathbb{T}_{m,c}^{\text{upload}} + \mathbb{T}_{i,c}^{\text{Cloud}}$$

$$\mathbb{E}_i^{Cloud} = \mathbb{E}_{m,c}^{\text{upload}} + \mathbb{E}_{i,c}^{\text{Cloud}}$$

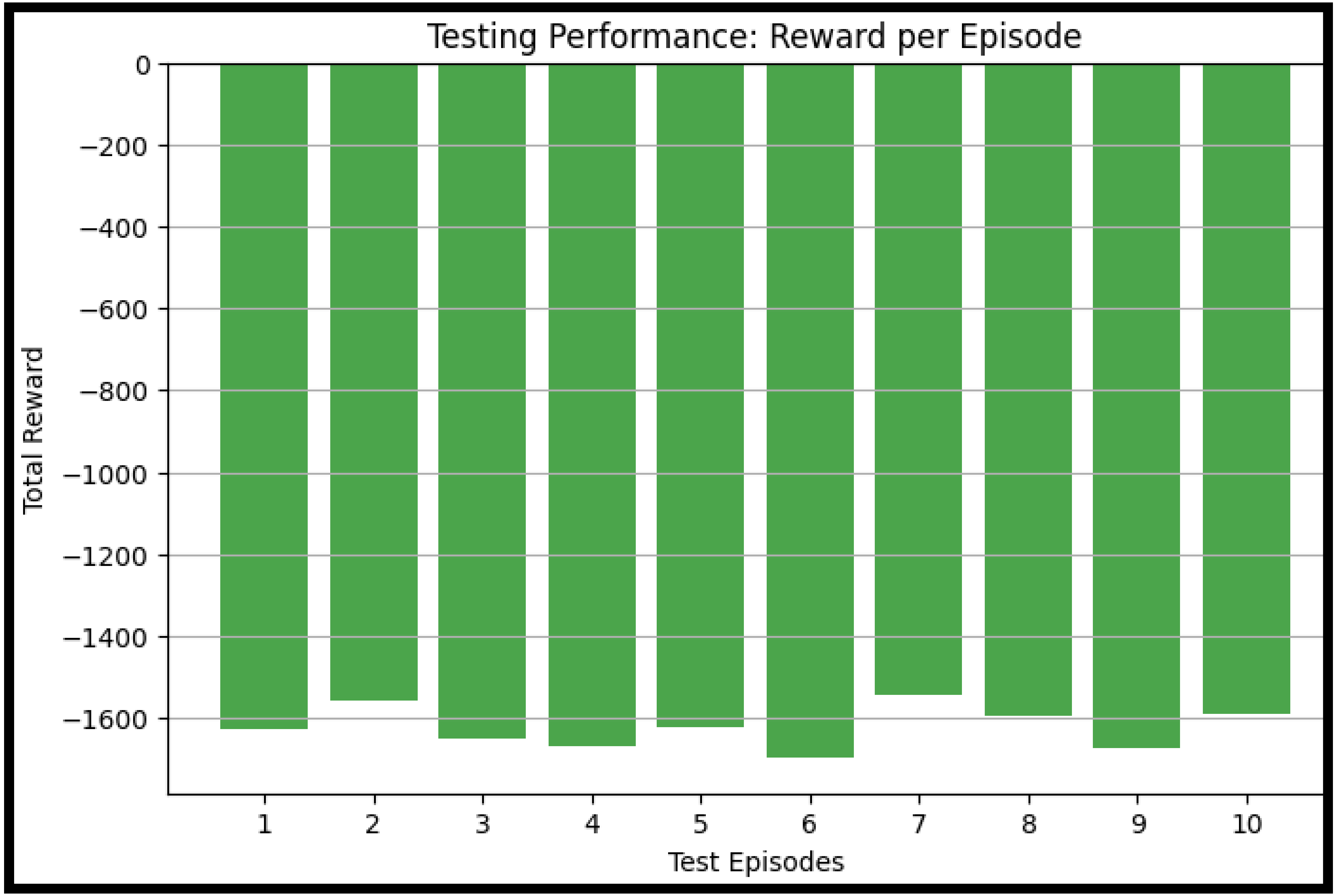
Previous Results Acheived

Training Performance



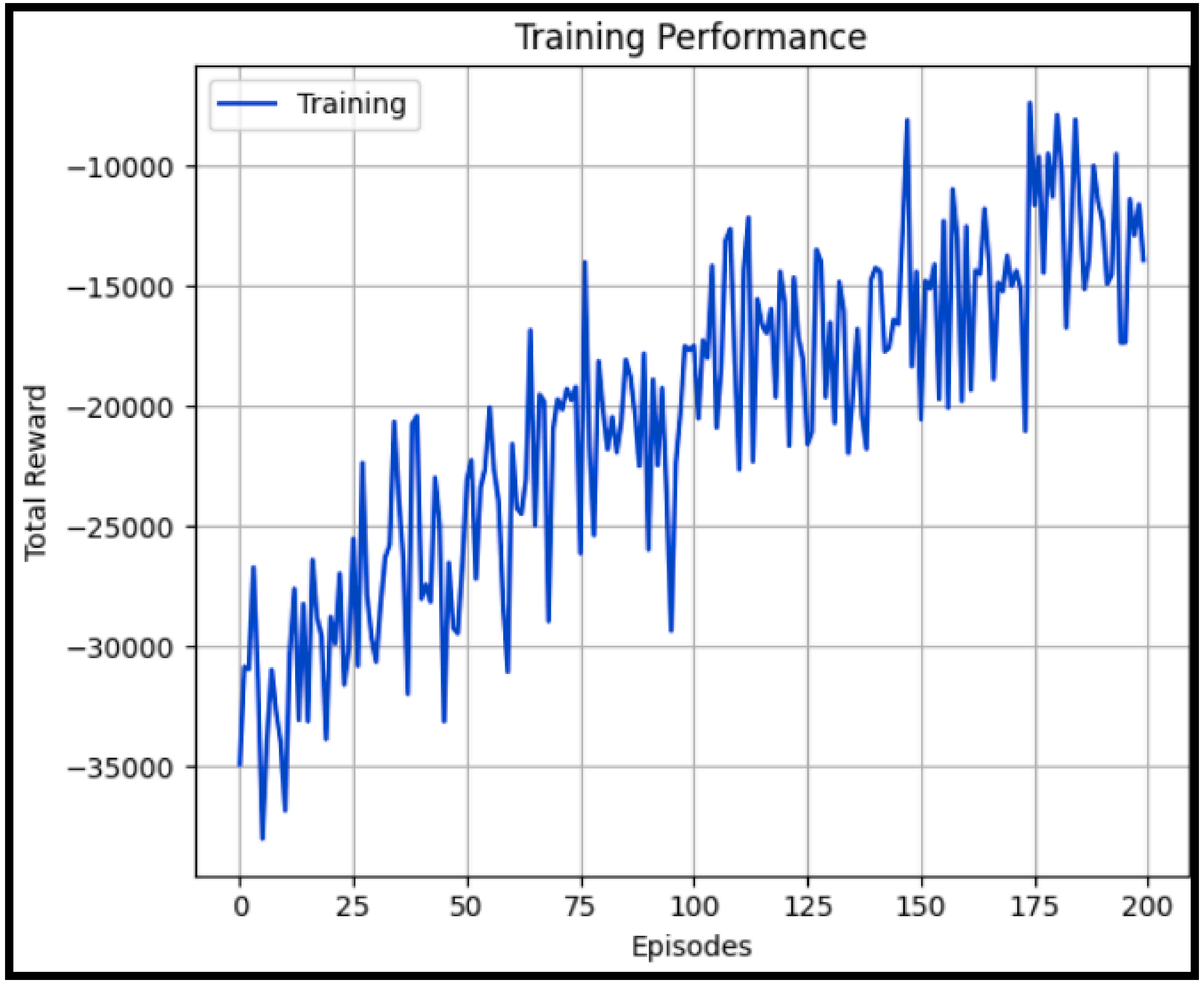
Previous Results Acheived

Testing Performance



Current Results Acheived

Training Performance



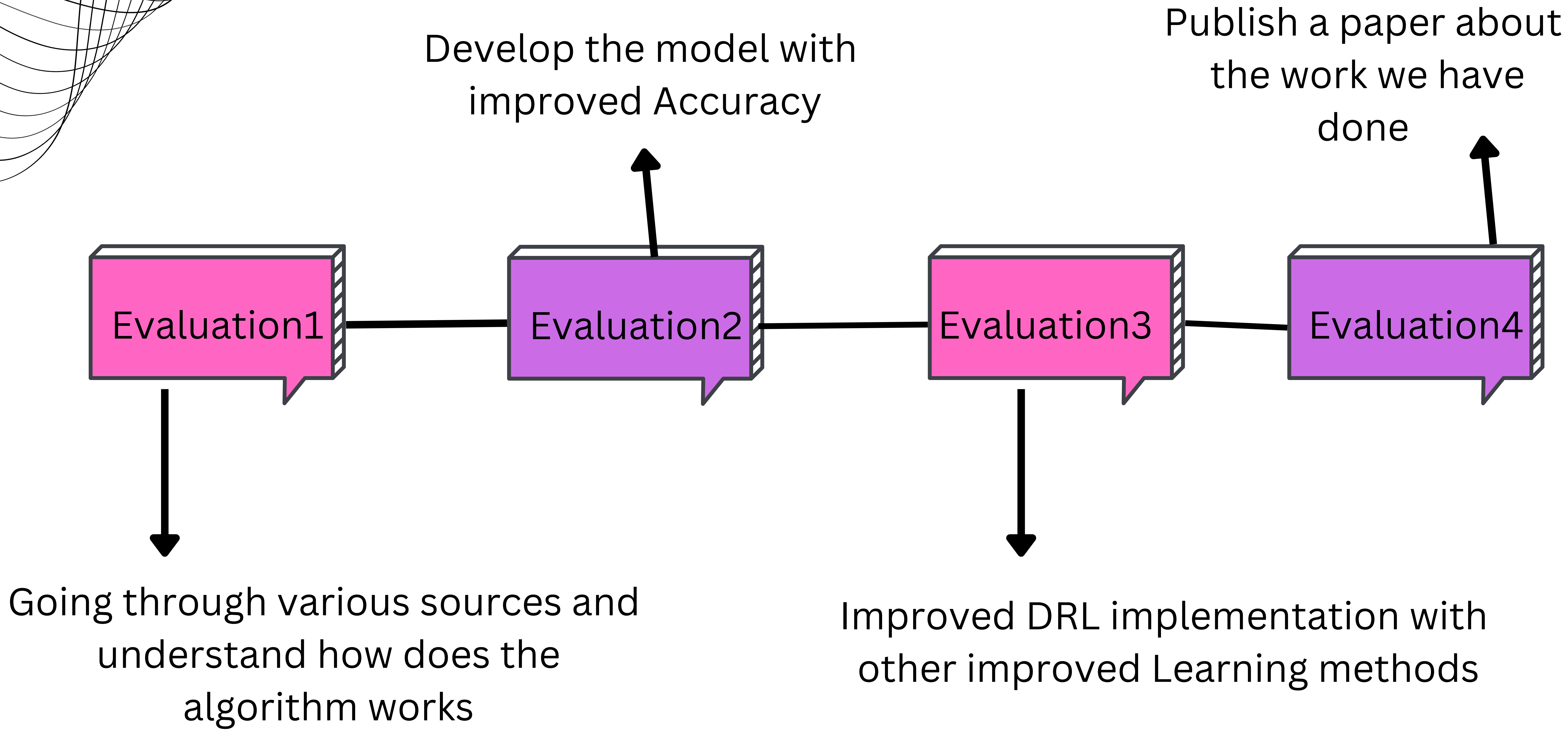
Epsilon (ε) is decreasing from 1.0 to 0.01 .

Current Results Acheived

Testing Performance

Epsilon (ε) is 0.

TimeLine


Develop the model with improved Accuracy
Publish a paper about the work we have done
Evaluation1
Evaluation2
Evaluation3
Evaluation4
Going through various sources and understand how does the algorithm works
Improved DRL implementation with other improved Learning methods

References

[1]

Deep Reinforcement Learning for Task Partitioning and Partial Offloading in UAV Networks

Srivikas Varasala, Veera Manikantha Rayudu Tummala, Suhas N Reddy, Sampath Kumar Talada, Abhishek Hazra, Mohan Gurusamy

[2]

Intelligent Service Deployment Policy for Next-Generation Industrial Edge Networks

Abhishek Hazra, Mainak Adhikari, Tarachand Amgoth, Satish Narayana Srirama

[3]

Collaborative AI-enabled Intelligent Partial Service Provisioning in Green Industrial Fog Networks

Abhishek Hazra, Mainak Adhikari, Tarachand Amgoth, Satish Narayana Srirama

Thank you!