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

- Introduction
- Motivation
- Literature Survey
- Problem Statement
- Basic Diagram
- Problem Statement Diagram
- Implementation
- Timeline
- References

INTRODUCTION

1. Service deployment is the process of making software or application services available in a network or computing environment.

2. An intelligent service deployment strategy addresses the challenge of efficiently deploying services.

3. Developing a Deep Reinforcement Learning (DRL) can enable accurate decision making .

Motivation

We want to improve the Quality of Service by deploying services through optimizing energy consumption and reducing latency.

The additional motivation arises from the recent effectiveness of Deep Reinforcement Learning (DRL)

Literature Survey

	Key Findings	Master Slave Algorithm	Intelligent Service Deployment	Trade Off Learning (Exploration and Exploitation)
[1]	Task Partitioning and Partial Offloading in UAV Networks	✗	✓	✗
[2]	Service Deployment Policy for Next-Generation Industrial Edge Networks	✗	✓	✓
[3]	Task Partitioning and AI Based Intelligence Service Provision	✓	✓	✗
Our work	Master-Slave Task-Off loading using DRL	✓	✓	✓

Problem Statement

Our focus is on developing an intelligent service deployment system with the help of Deep reinforcement Learning (DRL)

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

Diagram Oveview

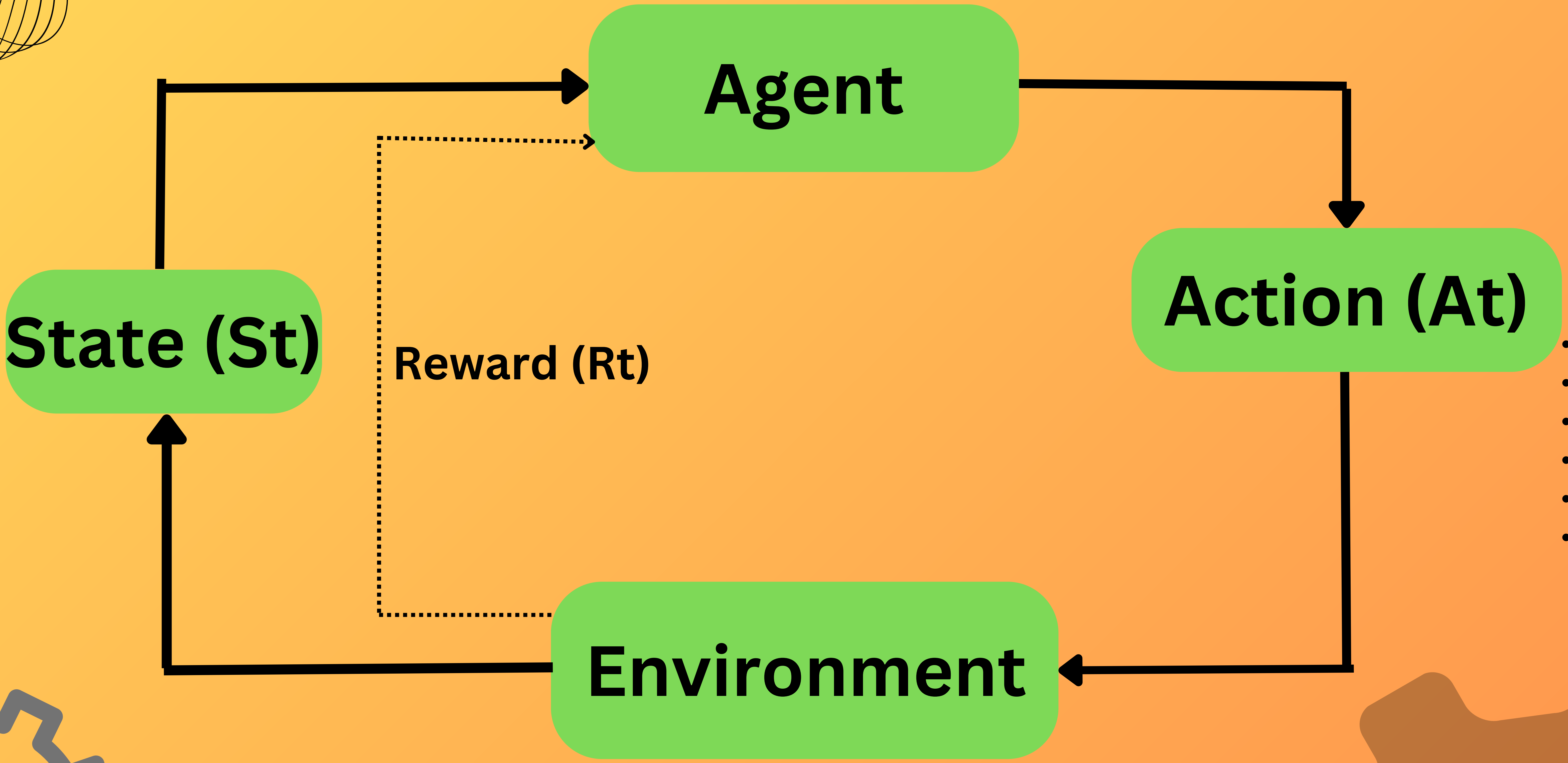
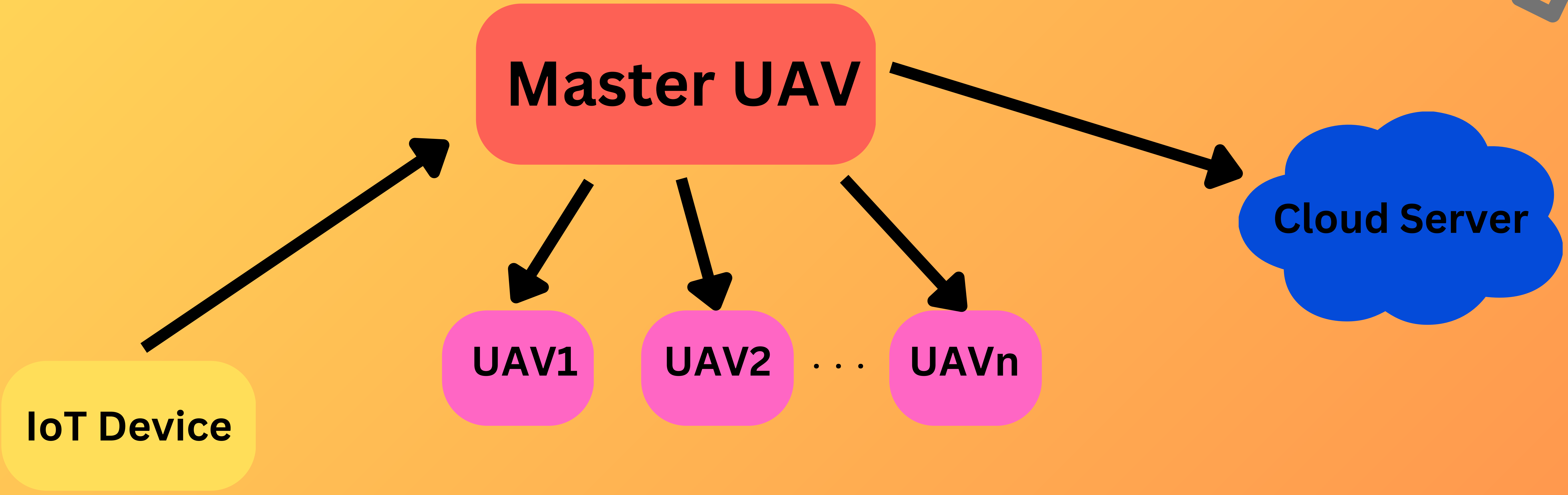


Diagram Oveview



Implementation

Phase 1

Network model & Problem Formulation

Phase2

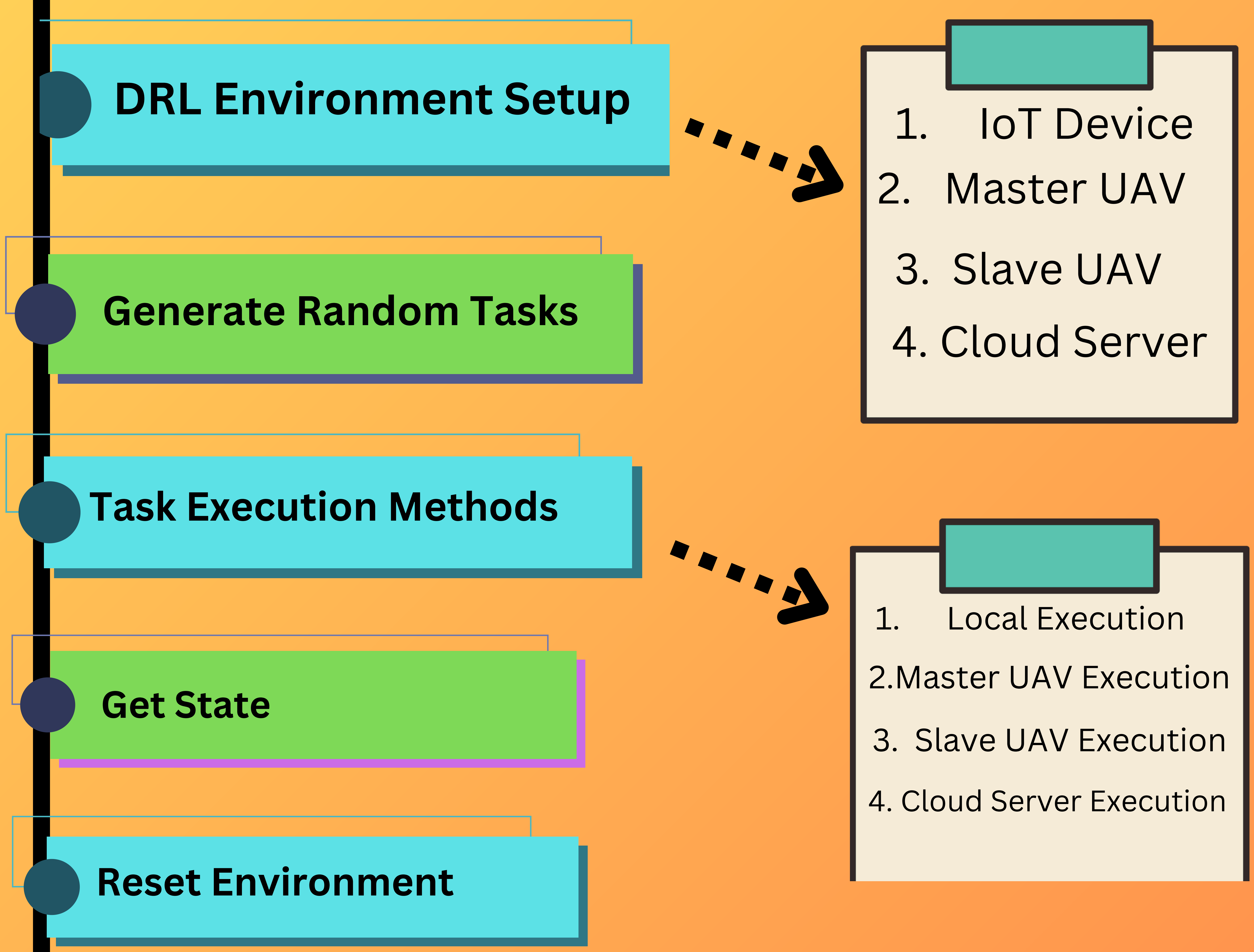
DRL-Based task offloading

Phase3

Numerical Analysis (Experiment and Results)

Phase 1

Network Model and Problem Formulation



DRL Environment Setup

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

Generate Random Tasks

Task Execution Methods

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

Get State

Reset Environment

Phase 2

DRL - Based Task Offloading

- Task Partitioning
- Deep Q - Learning (DQN)
- Algorithm Implementation

Phase 2

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.

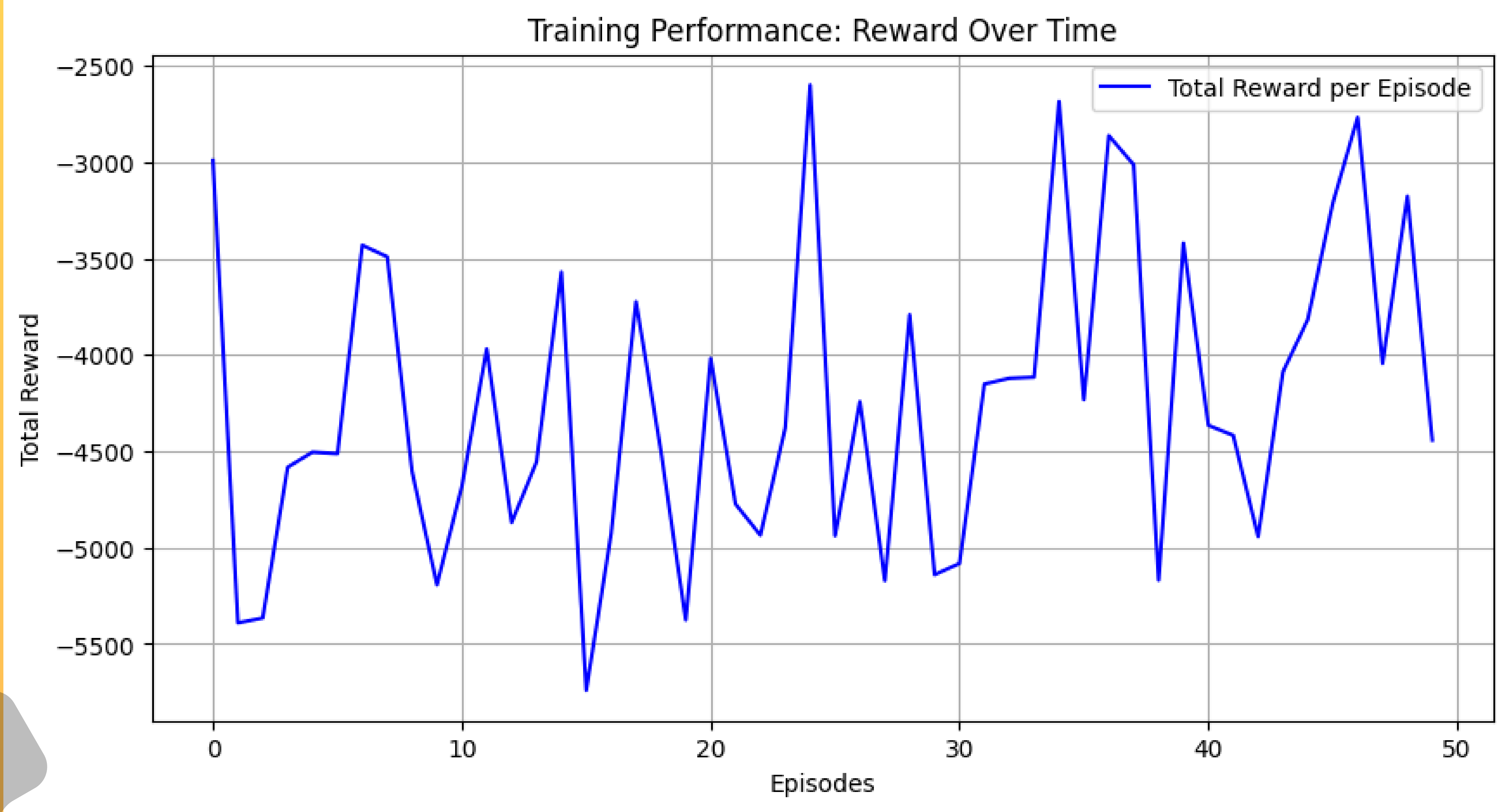
Phase 3

Numerical Analysis

- Performance Metrics
- Comparison with BenchMarks
- Graphs and Analysis

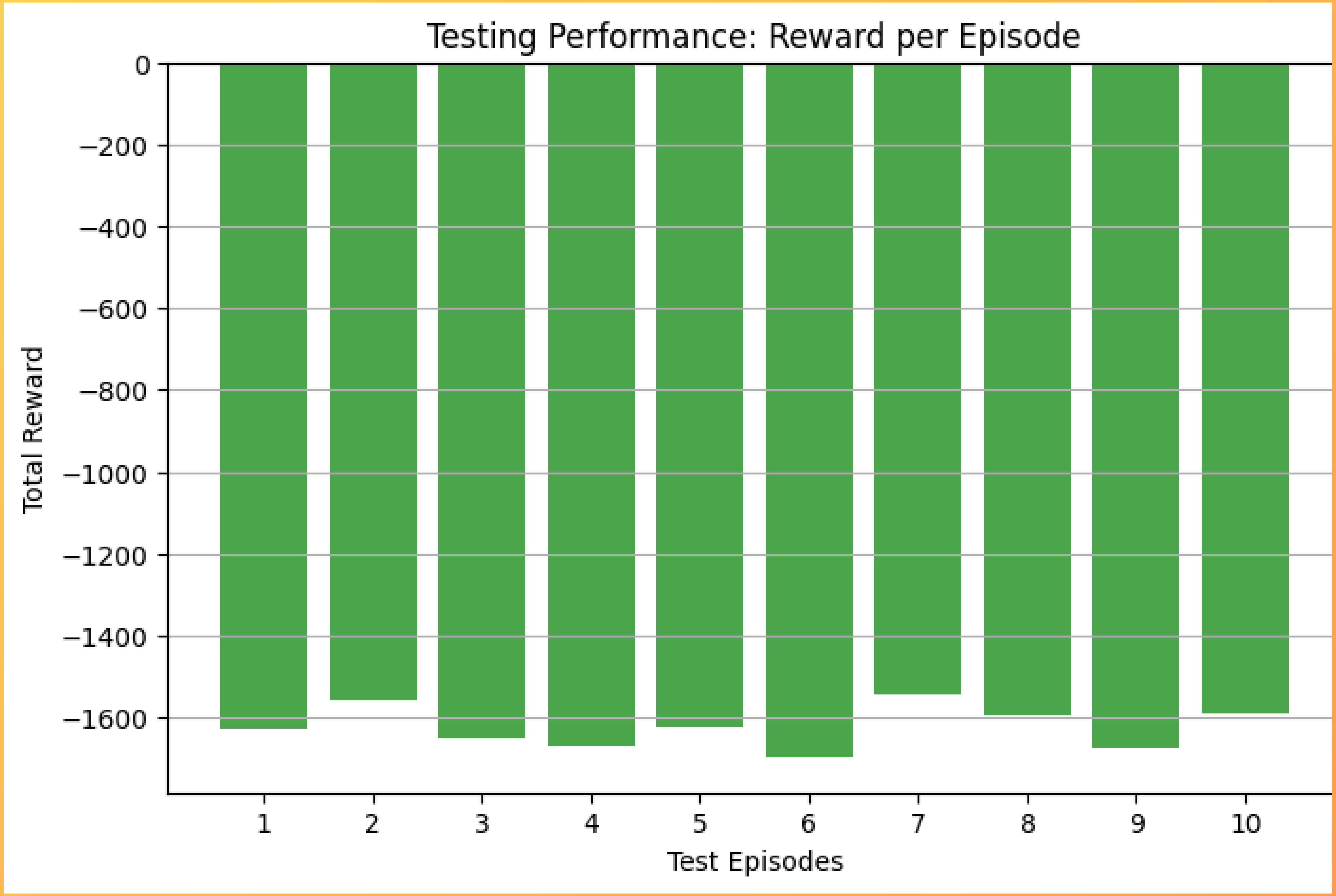
Results Acheived

Training Performance

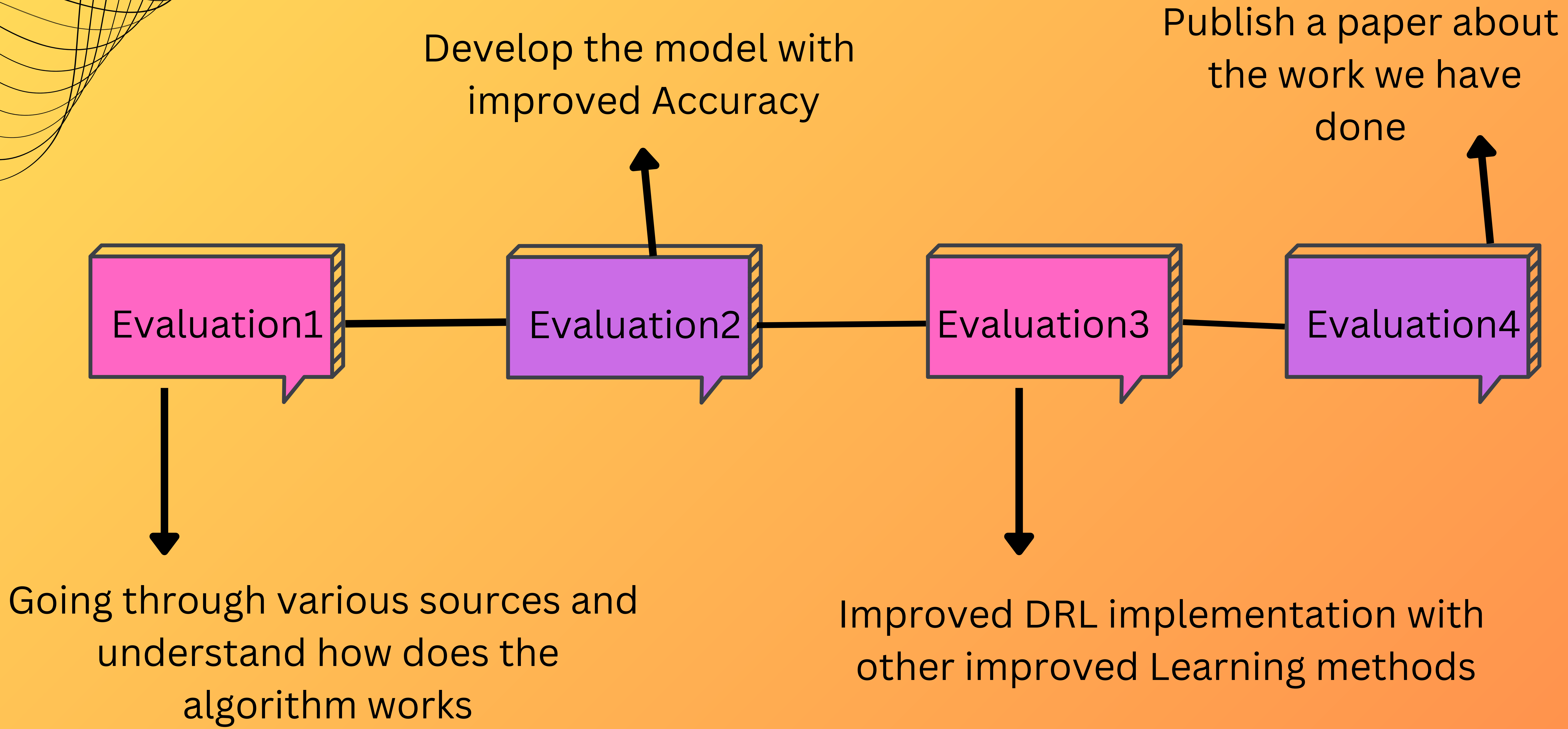


Results Acheived

Testing Performance



TimeLine



Evaluation1

Going through various sources and understand how does the algorithm works

Evaluation2

Develop the model with improved Accuracy

Evaluation3

Improved DRL implementation with other improved Learning methods

Evaluation4

Publish a paper about the work we have done

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!