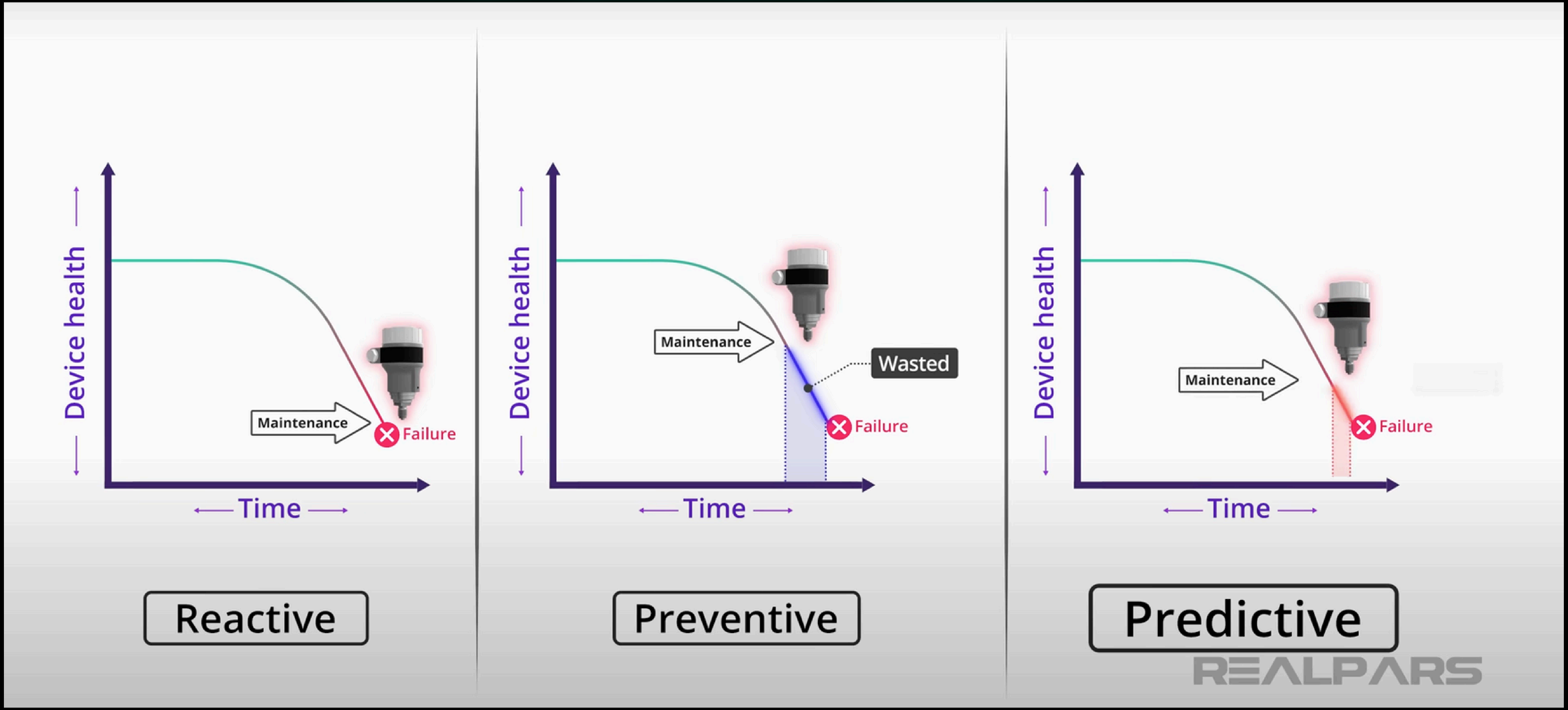
YZU - Introduction to Smart Manufacturing

AI Driven Predictive Maintenance

1113534 古弘恩 Hantz
1113535 田雷西 Max
1113548 范瑞德 Arridson

january 09, 2025

Introduction

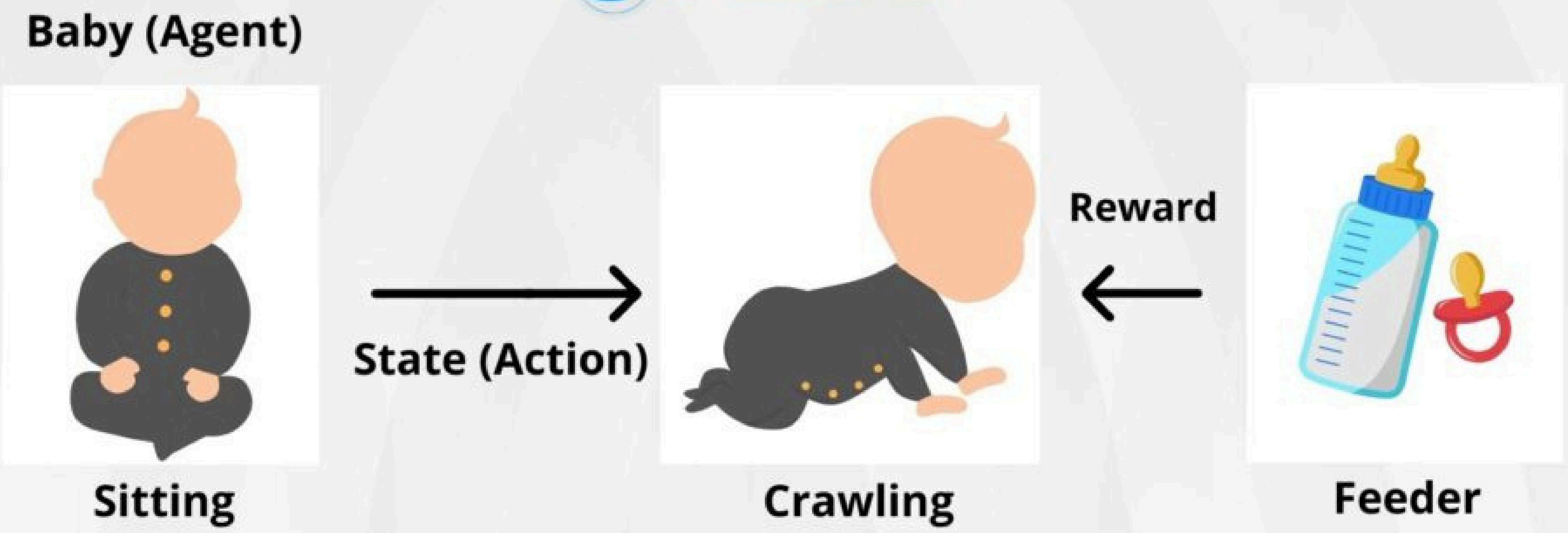


- Predictive Maintenance

REINFORCEMENT LEARNING

Reinforcement learning is a machine learning paradigm that focuses on how agents learn to interact with an environment to maximize cumulative rewards.

DatabaseTown



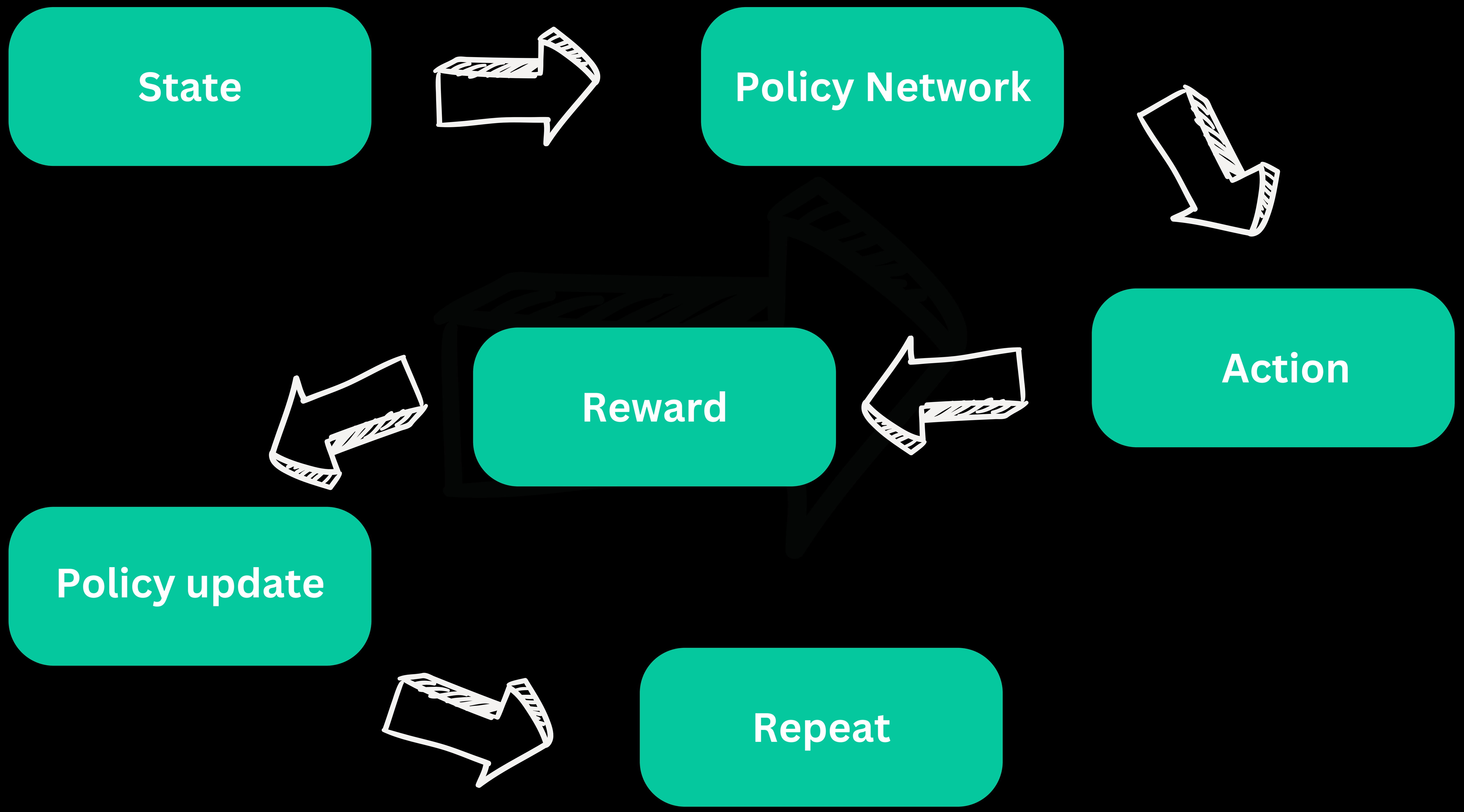
Algorithms and Approaches in Reinforcement Learning

- Q-learning
- Deep Q-networks (DQN)
- Policy Gradients Methods
- Proximal Policy Optimization (PPO)

Model Training Using PPO

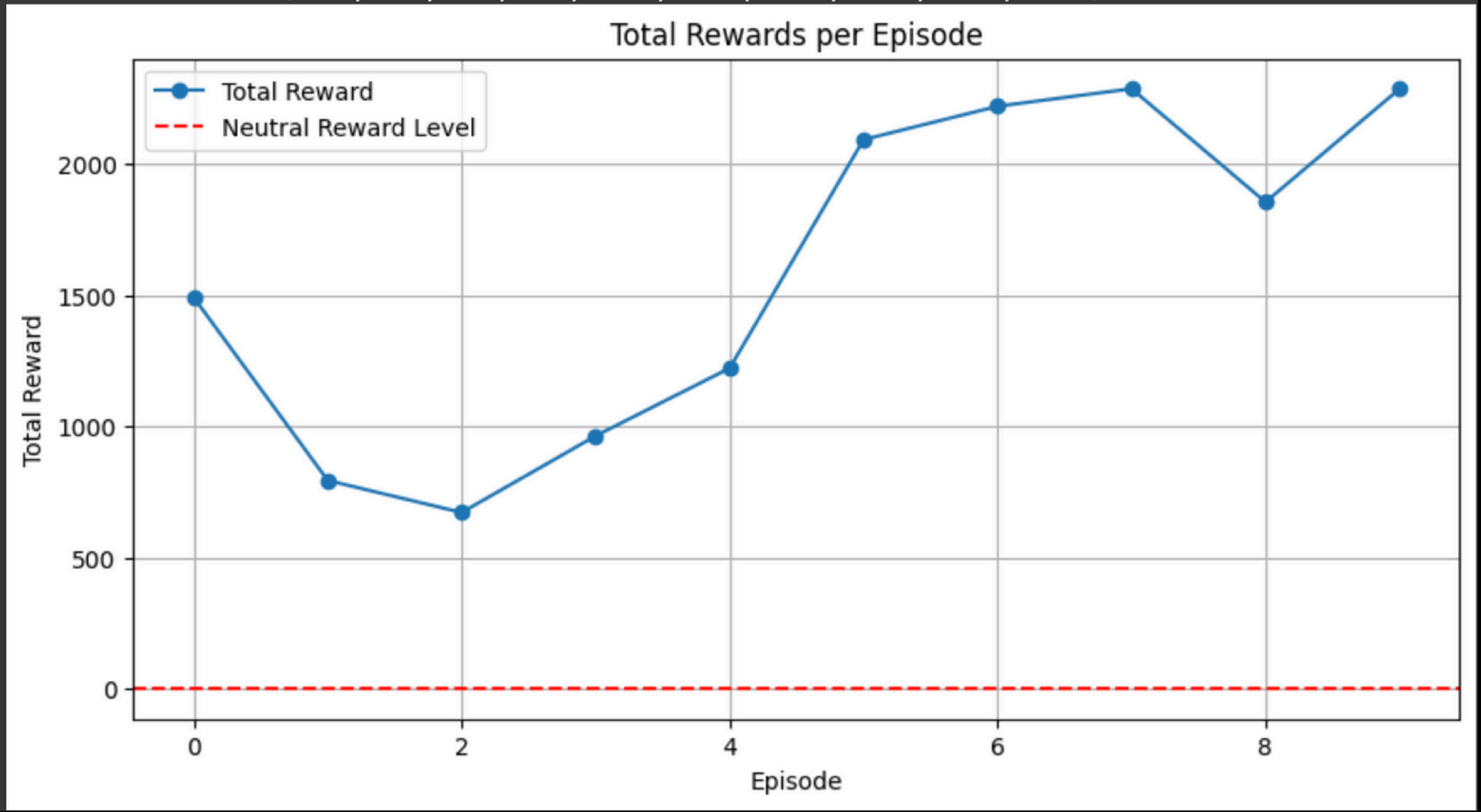
- **Algorithm: Proximal Policy Optimization (PPO)**
  - PPO was selected for its robustness and scalability in reinforcement learning tasks.
  - It balances exploration and exploitation during training using a ***clipped surrogate objective.***
- **Hyperparameters**:
  - ***Entropy coefficient (ent_coefent_coefent_coef):*** Set to 0.01 to encourage the agent to explore more diverse actions.
  - ***Learning Rate:*** Default value optimized for PPO.
  - ***Timesteps:*** 10,000 total training timesteps.
- **Code Explanation:**
  - The custom environment EquipmentEnv was instantiated with the training dataset (X_train, y_train).
  - The PPO model was defined and trained as follows:

```
env = EquipmentEnv(X_train, y_train)
model = PPO("MlpPolicy", env, verbose=1, ent_coef=0.01)  # Entropy for expl
model.learn(total_timesteps=10000)
```

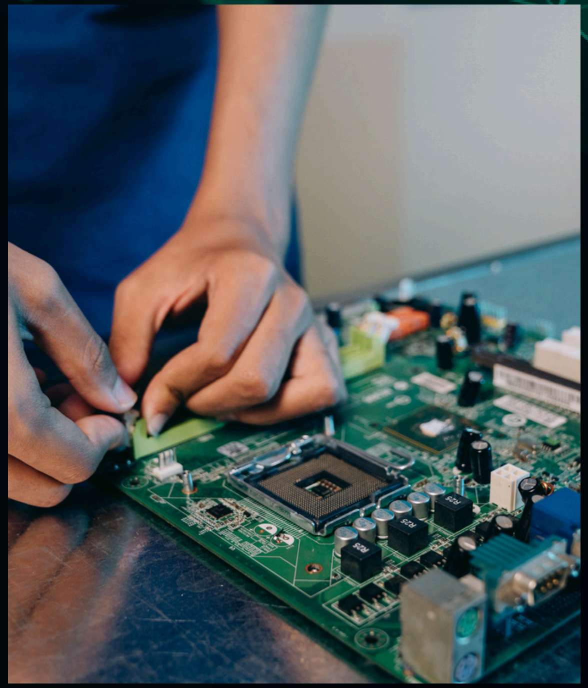

State
Policy Network
Action
Reward
Policy update
Repeat

Reward Trends Across Episodes

1. **Total Rewards Per Episode:**
   a. The graph displays the total rewards accumulated by the agent in each episode.
   b. X-axis: Episodes.
   c. Y-axis: Total Rewards.
2. **Observations from the Graph:**
   a. ***Initial Variability:*** The first few episodes show a noticeable drop in total rewards.
      i. The agent tries random actions and hasn't yet learned effective strategies.
   b. ***Gradual Improvement:*** From episode 3 onward, rewards begin to increase steadily.
      i. The agent is learning the optimal policy and improving its decisions over time
   c. ***Plateau and Minor Variations:*** After episode 5, rewards stabilize and reach a near-peak performance.
      i. Minor fluctuations (e.g., a dip in episode 7), which may correspond to occasional suboptimal actions or exploration.


Total Rewards per Episode
Total Reward
Neutral Reward Level
2000
1500
1000
500
0
0
2
4
6
8
Episode
Total Reward

Agent Decision-Making Analysis



Actions Taken by the Agent:

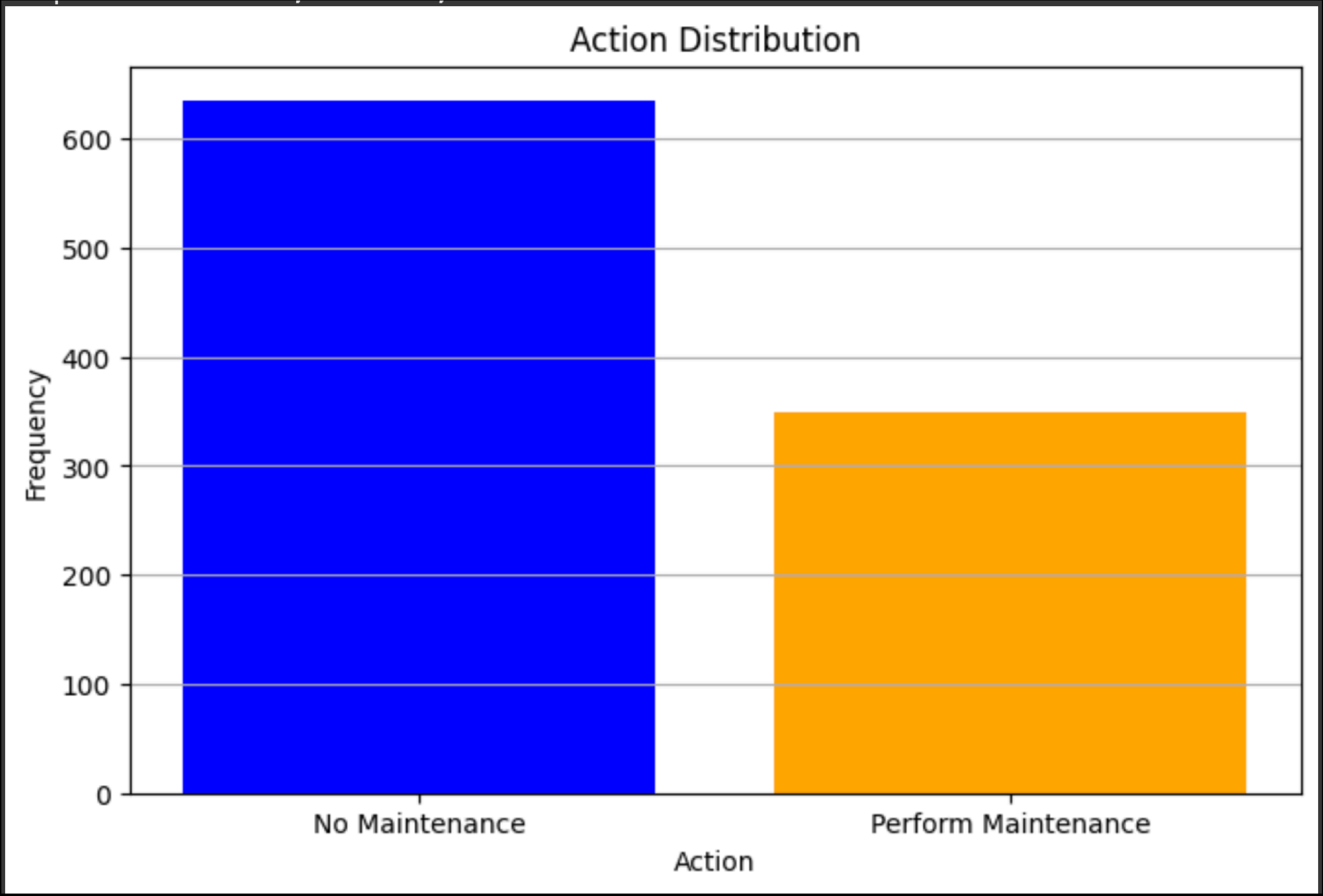
- **Action 0:** No maintenance.
- **Action 1:** Perform maintenance.

Observed Patterns:

- The agent favors Action 0 (No Maintenance) when the equipment is unlikely to fail.
- Action 1 (Perform Maintenance) is chosen more frequently when failures are likely, demonstrating the agent's understanding of the environment.

Interpretation:

- The agent prioritizes "No Maintenance" to maximize rewards.
- "Perform Maintenance" is chosen selectively to avoid penalties.


Action Distribution
Frequency
0
100
200
300
400
500
600
No Maintenance
Perform Maintenance
Action

Challenges and Limitations

Reward balance

- Small Penalties: Discourage undesirable actions without overly punishing the agent.
- Large Rewards: Strongly incentivize desirable actions, guiding the agent toward optimal behavior.

Why Is Reward Balance Important?

- Exploration vs. Exploitation
- Learning Stability
- Real-World Relevance

Future Work

- Incorporate real-world datasets for practical validation.
- Explore other DRL algorithms like DQN.

The Promise of DRL in Predictive Maintenance

- Integration with IoT
- Cross-Industry Applications
- Data-Driven Efficiency

Thank You