Project Report-Reinforcement Learning in Supply Chain Management

Comparing Traditional Supply Chain Policies with Reinforcement Learning for Platelet Inventory Management in a Single-Echelon Blood Bank Environment

-Joyal Pasricha (DA24C008)
Joint M.Sc Data Science and Artificial Intelligence

The code and graphs for this project have been uploaded on this GitHub repository for reference:

`GITHUB REPO FOR CODE AND VISUALIZATIONS`.

Contents

1 Abstract

- The efficient management of platelet inventory in blood banks is a critical challenge due to the highly perishable nature of platelets and fluctuating demand in healthcare environments. This study investigates the comparative effectiveness of reinforcement learning (RL) methods, specifically Deep Q-Networks (DQN), against traditional inventory policies, namely the (s, Q) policy and the (R, S) policy , in managing a single-echelon supply chain for platelets.
- A simulation framework is developed to emulate a platelet inventory system, capturing real-world constraints such as limited shelf life, lead times, and various cost components, including holding, shortage, wastage, and transport costs.
- Comprehensive experiments were conducted using synthetic datasets, generated to reflect realistic demand distributions and operational scenarios. Results reveal that while traditional policies demonstrated consistent performance in minimizing costs, the DQN agent exhibited superior adaptability to varying demand patterns, albeit with higher initial costs and lower fill rates due to exploration during training.
- This research highlights the potential of reinforcement learning in supply chain management as a scalable, data-driven alternative to traditional methods. It suggests a hybrid approach combining RL with traditional techniques to achieve cost-effective and efficient inventory management for perishable goods as a solution.
- Future work would focus on expanding the framework to multi-level supply chains and using real-world data for validation, enabling practical use in healthcare scenarios.

2 Introduction

Efficient Inventory Management in Healthcare

Efficient inventory management in healthcare is paramount due to the critical nature of resources and the perishable characteristics of products like platelets. Platelets have a short shelf life, which necessitates precise inventory strategies to minimize wastage while ensuring demand fulfillment. This study focuses on a single-echelon stochastic supply chain involving suppliers, blood banks, and hospitals as the end consumers. The supply chain's stochastic nature arises from variability in platelet demand, driven by unpredictable factors such as emergencies and seasonal fluctuations.

Challenges include:

- **Perishability:** Platelets must be utilized within five days (shelf life = 5 days), making wastage costs high.
- **Stochastic Demand:** A daily demand modeled with a half-normal distribution (mean = 30, std = 15) introduces uncertainty.
- **Cost Sensitivity:** Costs are incurred across multiple dimensions: holding (5/unit/day), wastage (50/unit), shortage (100/unit), and transport (150 fixed + 5/unit variable).

This study addresses these challenges by developing and comparing three policies: the SQ ($s,\ Q$) policy, RS ($R,\ S$) policy, and reinforcement learning-based Deep Q-Network (DQN) method. The analysis highlights how these methods optimize costs, minimize wastage, and improve service levels in the context of healthcare.

2.1 Importance of Efficient Supply Chains in Healthcare

Healthcare supply chains are critical to patient care, with inefficiencies directly impacting lives. For platelets, stockouts can lead to severe medical risks, while excess inventory results in wastage due to expiration. A robust supply chain ensures:

- **High Fill Rates:** Meeting demand consistently (target greater than 90%).
- **Reduced Costs:** Balancing inventory holding costs against wastage and shortage costs.
- **Service Reliability:** Maintaining service cycle rates close to 100%.

For instance, this project's simulation shows that a well-optimized ($s,\ Q$) policy achieved a fill rate of 98.62% while maintaining a reasonable total cost of Rs. 189,218.

2.2 Objective

The objective of this project is to evaluate and compare the effectiveness of traditional supply chain policies ($s,\ Q$ and $R,\ S$) and reinforcement learning (DQN) in managing a single-echelon platelet inventory. The specific goals are:

- **Cost Optimization:** Minimize the total costs (holding + wastage + shortage + transport).
- **Performance Analysis:** Assess fill rate, service cycle rate, and wastage under stochastic demand.
- **Policy Comparison:** Highlight the strengths and limitations of traditional policies versus RL-based approaches.

2.3 Scope of the Study

The study is scoped to:

- **Single-Echelon Supply Chain:** Focused on the blood bank level, interfacing with suppliers and hospitals.
- **Fixed Parameter Settings:** Simulated demand variability and fixed costs. Parameters like lead time (2 days) and maximum capacity of the blood bank inventory (300 units) are static.

- **Policy Comparisons:**
  - **Traditional Policies:** Structured and deterministic (e.g., fixed reorder points or intervals).
  - **Reinforcement Learning:** Adaptive and capable of handling dynamic changes.
- **Limitations:**
  - **Simulation Assumptions:** Simplified assumptions such as constant lead times and demand distributions may not fully capture real-world variability to some extent.
  - **Training Constraints:** The DQN model was trained over 100 episodes (= 8 years of data), which may limit its generalization compared to extended training with more epochs and larger datasets.

Despite these limitations, this study provides valuable insights into the potential of reinforcement learning to revolutionize inventory management in healthcare.

3 Literature Review

3.1 Reinforcement Learning in Supply Chains

Reinforcement learning (RL) is a machine learning paradigm that enables agents to learn optimal policies by interacting with an environment through trial and error. In the context of supply chains, RL is particularly adept at addressing challenges such as stochastic demand, dynamic decision-making, and perishable inventory management. Unlike traditional policies, RL algorithms dynamically adapt to changing environments, making them suitable for highly uncertain supply chains.

This study utilizes the **Deep Q-Network (DQN)**, an advanced RL technique that integrates deep learning to approximate the Q-value function. The Q-value represents the expected cumulative reward from taking a specific action in a given state and following the learned policy thereafter. The DQN's architecture, implemented in this project, is designed to handle the large and continuous state spaces associated with platelet inventory management. The neural network architecture includes two fully connected hidden layers of 64 and 32 neurons.

Key aspects of the DQN implementation in this project are as follows:

- **State Representation:** States are defined as vectors comprising:
  - Total inventory across all age categories.
  - Pending orders with lead times (taken as 2 days constantly).
  - Inventory distribution by age (0–6 days).

For example, at any given timestep, the state vector might appear as: Here, 150 is the total inventory, 20 represents pending orders, and subsequent values detail inventory by age.

- **Action Space:** The action space includes discrete ordering decisions from 0 to 300 (max inventory capacity) units in increments of 10. This provides 31 possible actions, allowing fine-grained control over replenishment decisions.
- **Reward Function:** The reward is defined as the negative total cost incurred at each step, i.e., the sum of holding, wastage, shortage, and fixed and variable transportation costs.
- **Training Strategy:** The DQN agent uses an epsilon-greedy strategy to balance exploration (selecting random actions) and exploitation (choosing the best-known actions). Epsilon starts at 1.0 and decays linearly to 0.05 over 100 episodes, ensuring the agent explores various state-action combinations in the early stages of training and exploits learned policies as training progresses.

Implementation Example: Suppose in the first episode, the agent observes an initial state corresponding to 50 total units, no pending orders, and specific age-wise distributions. After selecting an action to order 40 units, the state transitions to include new inventory levels and demand fulfillment. Rewards are computed based on the cost incurred during this step, and the agent updates its Q-value approximations using the Bellman equation.

Through iterative learning, the DQN significantly reduces total costs over 100 training episodes, initially incurring an average cost of and eventually reaching in the final episodes. The agent dynamically adapts its ordering decisions to balance wastage and shortages effectively.

3.2 Traditional Supply Chain Policies

Traditional policies like $(s,\ Q)$ and $(R,\ S)$ are widely used in supply chain management due to their simplicity and effectiveness in structured environments.

$(s,\ Q)$ Policy: The $(s,\ Q)$ policy is a continuous review inventory policy that triggers an order of quantity Q when inventory levels fall below the reorder point s. This study optimizes s and Q using stochastic demand training data. Optimization is performed using scipy optimizers like differential evolution and dual annealing, and optimal s and Q are used on test data.

Its advantage includes the fact that it is effective for items with consistent demand. However, it is not well-suited for stochastic or highly variable demand patterns.

$(R,\ S)$ Policy: The $(R,\ S)$ policy operates on a periodic review basis. Inventory levels are checked at fixed intervals (R), and if required, an order is placed to replenish stock up to a predefined level (S), where:

- **Review Period:** The fixed interval between inventory reviews.

- **Target Inventory Level (S):** The desired inventory level after replenishment.

Implementation in the project includes assessment of inventory level every R days. An order is placed to bring the inventory up to S if the current stock is below it. R and S are optimized to minimize costs.

Advantages of this method include its suitability for predictable demand scenarios and simplification of management by batching orders. However, it is less responsive to sudden demand changes between review periods.

3.3 Blood Bank-Specific Inventory Management Challenges

Managing platelet inventory in blood banks is uniquely challenging due to the interplay of perishability, stochastic demand, and high costs of shortages and wastage:

- **Perishability:** Platelets expire in five days, requiring precise demand forecasting and inventory rotation.
- **Stochastic Demand:** The half-normal distribution used in this study captures real-world demand variability.
- **Cost Sensitivity:** Multiple cost components must be balanced. For example:
  - Wastage cost (50/unit) dominates when inventory exceeds demand.
  - Shortage cost (100/unit) penalizes unmet demand.

Simulation data highlights these trade-offs. For instance, the DQN approach initially incurred high costs (943,375) due to exploration but demonstrated adaptability by dynamically adjusting orders to reduce wastage and shortages over time.

4 Methodology

4.1 Problem Formulation

This study focuses on managing platelet inventory in a **single-echelon stochastic supply chain**, where the blood bank acts as the central inventory hub, interfacing with suppliers (upstream) and hospitals (downstream). The problem involves managing perishable inventory (platelets with a 5-day shelf life), stochastic daily demand, and operational constraints to minimize total costs while ensuring high service levels.

Key Metrics:

1. **Lead Time:** A fixed lead time of **2 days** for replenishment orders was simulated to reflect real-world supply chain dynamics.

2. **Demand Generation:**
    - Demand was modeled using a **half-normal distribution** with:
        - Mean (μ) = 30 units/day.
        - Standard Deviation (σ) = 15 units/day.
    - The half-normal distribution helped to avoid negative demand values, essential for practical inventory scenarios.
3. **Transportation Costs:**
    - **Fixed Cost:** 150/order (covers processing and dispatch).
    - **Variable Cost:** 5/unit ordered which is proportional to quantity ordered.
4. **Shelf Life:** Platelets expire after **5 days**, necessitating accurate tracking of age categories (using dictionary with ages as keys and order quantities as corresponding values) to avoid wastage.
5. **Storage Capacity:** The simulation enforced a maximum capacity of **300 units** to model storage constraints effectively.
6. **Cost Components:**
    - **Holding Cost (HC):** 5/unit/day for stored inventory.
    - **Wastage Cost (WC):** 50/unit for platelets that expire.
    - **Shortage Cost (SC):** 100/unit for unfulfilled demand.

7. **Operational Challenges:** The problem formulation incorporates practical hurdles:

- **Demand Variability:** Stochastic demand requires robust policies to handle uncertainty.
- **High Perishability:** The short shelf life demands frequent monitoring and optimal ordering to minimize wastage.
- **Cost Sensitivity:** Balancing multiple cost components while maintaining service levels is critical.
- **Order Delays:** The 2-day lead time adds complexity in predicting demand and replenishment timing.

4.2 Simulation Environment

A simulation model was developed to replicate the dynamics of platelet inventory management. The environment integrates:

1. **Inventory Aging:** Platelets are tracked daily. Units exceeding the 5-day shelf life are considered wasted.

2. **Demand Fulfillment:** A **First-In-First-Out (FIFO)** system fulfills demand using the oldest inventory first.
3. **Pending Orders:** Orders are processed after a 2-day lead time, simulating real-world replenishment delays.
4. **Objective Function:** The primary goal is to minimize total costs, defined as:

$$TotalCost(TTC) = HC(Holdingcost) + WC(Wastagecost) + SC(shortagecost) + TCT(transportcost)$$

4.3 Policies Implemented

4.3.1 (*s, Q*) Policy

Mechanism: Place an order of size Q whenever the total inventory falls below a reorder point s.

4.3.2 (*R, S*) Policy

Mechanism: At fixed intervals R, replenish inventory to a target level S.

4.3.3 Deep Q-Network (DQN)

Mechanism: Uses reinforcement learning to dynamically decide order quantities based on the current state.

Implementation:

- **State Representation:** A vector comprising:
  - Total inventory.
  - Pending orders.
  - Age-wise inventory distribution.
- **Action Space:** Ordering decisions from 0 to 300 units in steps of 10.
- **Reward Function:** Negative total cost: $R = -(HC + WC + SC + TCT)$.
- **Training Process:** Neural network approximates Q-values for all actions. Training episodes were 100 (30 timesteps per episode) with epsilon decay for exploration-exploitation balance.

5 Implementation

5.1 Blood Bank Environment Setup

The code and graphs for this project have been uploaded on this GitHub repository for reference:

`GITHUB REPO FOR CODE AND VISUALIZATIONS`.

The following configurations and features were established:

1. **Initial Inventory Configuration:**
   - **Initial Inventory Level:** Set at 50 units, distributed across different age categories (0 to 6 days old).
   - **Maximum Capacity:** The storage limit was set at 300 units, representing physical space constraints in blood banks.
2. **Simulation Parameters:**
   - **Episodes:** 100 training episodes and 12 testing episodes were conducted.
   - **Steps per Episode:** Each episode consisted of 30 timesteps, simulating daily operations.
3. **Features of the Simulated Environment:**
   - **Inventory Aging:** Platelets age by one day after each timestep. Units exceeding the shelf life of 5 days are discarded as wastage.
   - **Demand Fulfillment:** Daily demand is fulfilled using the FIFO (First In, First Out) approach, prioritizing older inventory.
   - **Pending Orders:** Orders are processed with a fixed lead time of 2 days.
   - **Cost Components:**
     - Holding Cost: Rs.5/unit/day.
     - Wastage Cost: Rs.50/unit for expired platelets.
     - Shortage Cost: Rs.100/unit for unmet demand.
     - Transport Costs: Rs.150 fixed cost + Rs.5/unit variable cost.
4. **Demand Generation:** Modeled using a half-normal distribution with parameters:
   - Mean = 30.
   - Standard Deviation = 15.
5. **Data Generation:**
   - **Training Data:** Simulated for 100 episodes (30 timesteps per episode so approximately 8 years of data) and saved to `training_data.xlsx`.
   - **Testing Data:** Simulated for 12 episodes (30 timesteps per episode so approximately 1 year of data) and saved to `test_data.xlsx`.

5.2 Policy Optimization

(s, Q) Policy Optimization:

- **Optimization:** s and Q were optimized using dual annealing and differential evolution.
- **Results:**
  - Optimized parameters: $s = 76.02$, $Q = 33.60$.
- **Simulation Metrics:**
  - Fill Rate: 98.62%.
  - Total Cost: Rs.189,218.

(R, S) Policy Optimization:

- **Optimization:** Demand from the simulation was used as input to the optimization function. R and S were tuned to minimize costs using dual annealing and differential evolution.
- **Results:** Optimized Parameters: $R = 1$, $S = 148$.
- **Simulation Metrics:**
  - Fill Rate: 97.25%.
  - Total Cost: Rs.201,105

5.3 Reinforcement Learning Training

The DQN-based policy uses a neural network to approximate the Q-value function for decision-making. The network predicts the expected future reward for each possible action in a given state.

Network Architecture:

- Input Size: 9 features (total inventory, pending orders, and age-wise inventory distribution).
- Hidden Layers: Two fully connected layers with 64 and 32 neurons, respectively.
- Output Size: 31 actions (order quantities from 0 to 300 in steps of 10).

Training Process:

1. **State Representation:** States were represented as vectors capturing inventory and pending orders.

2. **Action Selection:** Epsilon-greedy strategy balanced exploration (random actions) and exploitation (optimal actions). Epsilon decayed from 1.0 to 0.05.
3. **Q-Value Update:** The Bellman equation updated Q-values:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[R + \gamma \max_{a} Q(s',a') - Q(s,a) \right]$$

4. **Loss Function:** Mean Squared Error (MSE) between predicted and target Q-values.

Training Metrics:

- Initial Total Cost: Rs. 265,325 (high due to exploration).
- Final Total Cost: Rs. 80,975 after 100 episodes i.e. approx 8 years of data
- Fill Rate = 66.44%.

5.4 Testing and Validation

Testing Methodology:

- Policies were evaluated on 12 episodes (i.e. 1 year of data) using the test dataset.
- Metrics such as total cost, fill rate, service cycle rate, and wastage were analyzed.

Results:

- **(s, Q) Policy:**
  - Total Cost: Rs. 189,218
  - Fill Rate: 98.62%.
- **(R, S) Policy:**
  - Total Cost: Rs. 201,105
  - Fill Rate: 97.25%.
- **DQN Policy:**
  - Total Cost: Rs. 943,375
  - Fill Rate: 66.44%.

6 Results and Analysis

6.1 Performance Metrics

To evaluate the policies implemented, key performance metrics were calculated:

- **Total Cost:** Sum of holding, wastage, shortage, and transport costs for an episode or the entire test period:

$$TC = HC + WC + SC + TCT$$

- **Fill Rate:** The proportion of demand fulfilled during an episode:

$$FillRate = \frac{FulfilledDemand}{TotalDemand}$$

- **Service Cycle Rate:** Fraction of time steps with no unfulfilled demand:

$$ServiceCycleRate = \frac{StepswithZeroShortage}{TotalSteps}$$

- **Wastage:** Total units discarded due to expiration:

$$Wastage = InventoryAgedBeyondShelfLife$$

- **Unfulfilled Demand:** Total units not met during an episode:

$$UnfulfilledDemand = TotalDemand - FulfilledDemand$$

6.2 Training Analysis (DQN)

The DQN model was trained over 100 episodes. The training process was evaluated based on:

- **Episodic Loss:** Loss decreased as the model learned to approximate Q-values.
- **Rewards:** The cumulative rewards improved as the agent optimized inventory decisions.

Training Curves:

- **Loss Curve:** Demonstrates convergence.
- **Reward Curve:** Reflects increasing rewards with training.

Training Metrics:

Insights:

- Loss decreased steadily, reflecting improved Q-value approximations.
- Rewards increased over episodes as the agent learned effective policies.

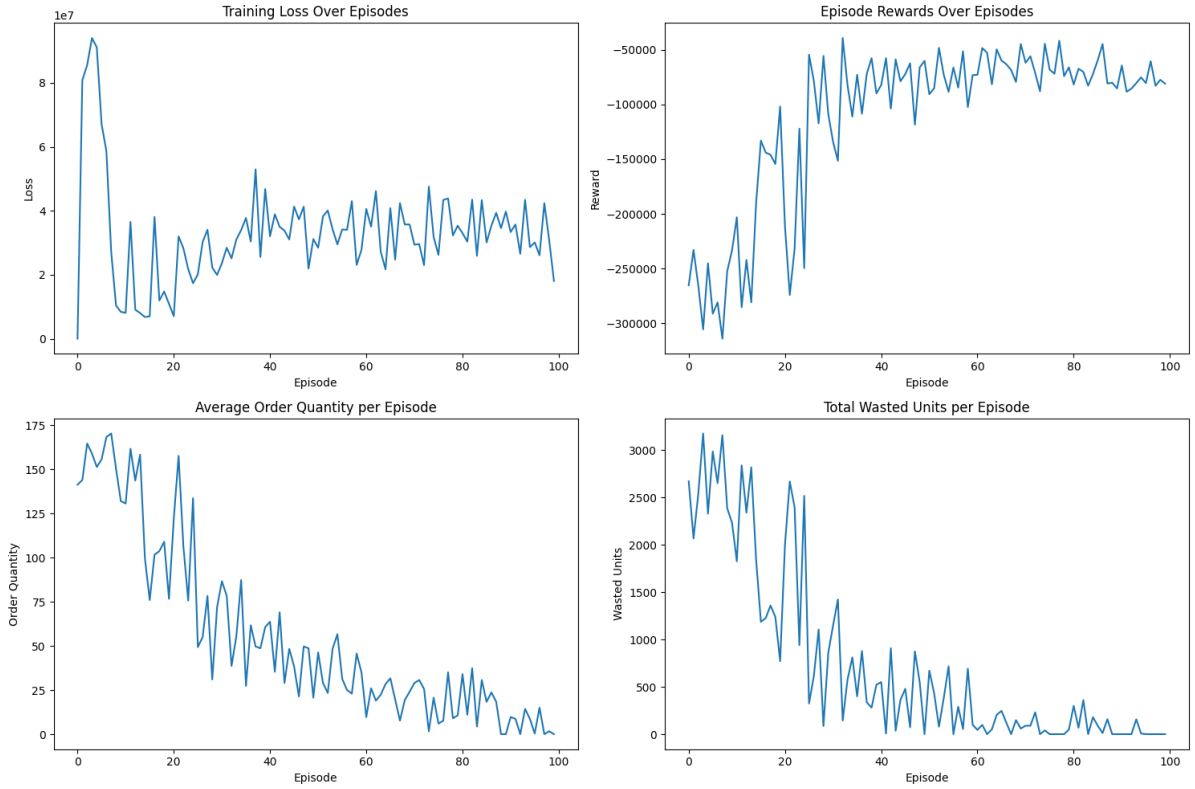

Figure 1: DQN training results

Metric	Episode 1	Episode 100
Total Cost (Rs.)	265,325	80,975
Average Reward	-8,844.17	-2,699.17

Table 1: Training Metrics for DQN

6.3 Training Visualizations for Comparison

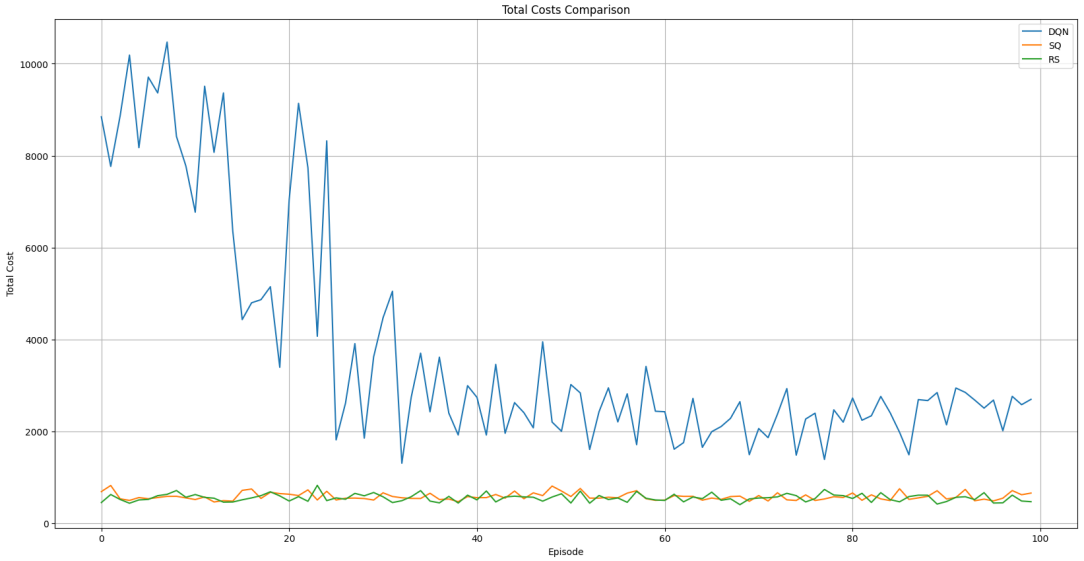

Figure 2: Total costs comparison during training

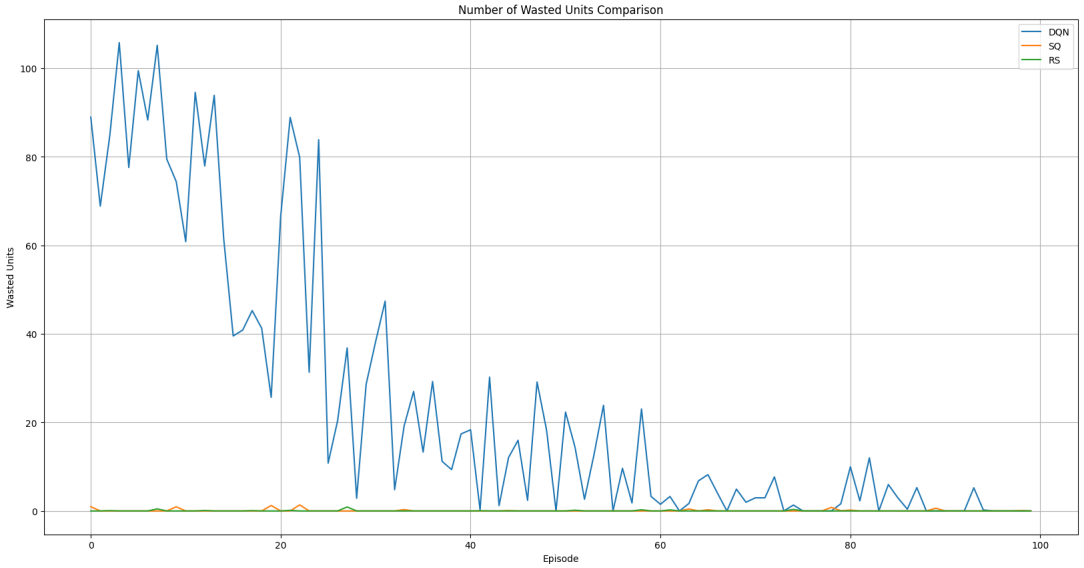

Figure 3: Wasted units comparison during training

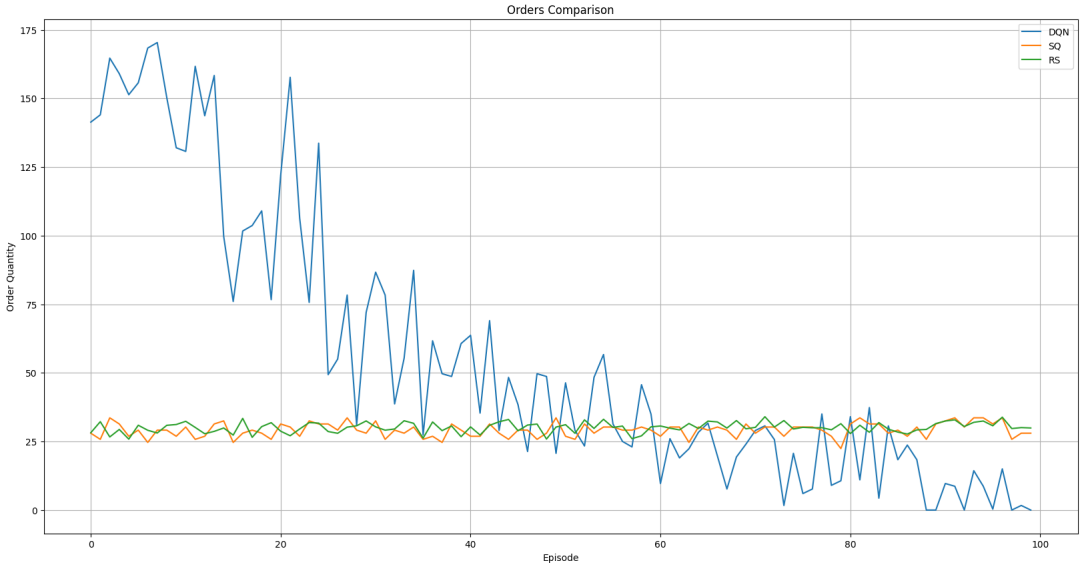

Figure 4: Order comparison during training

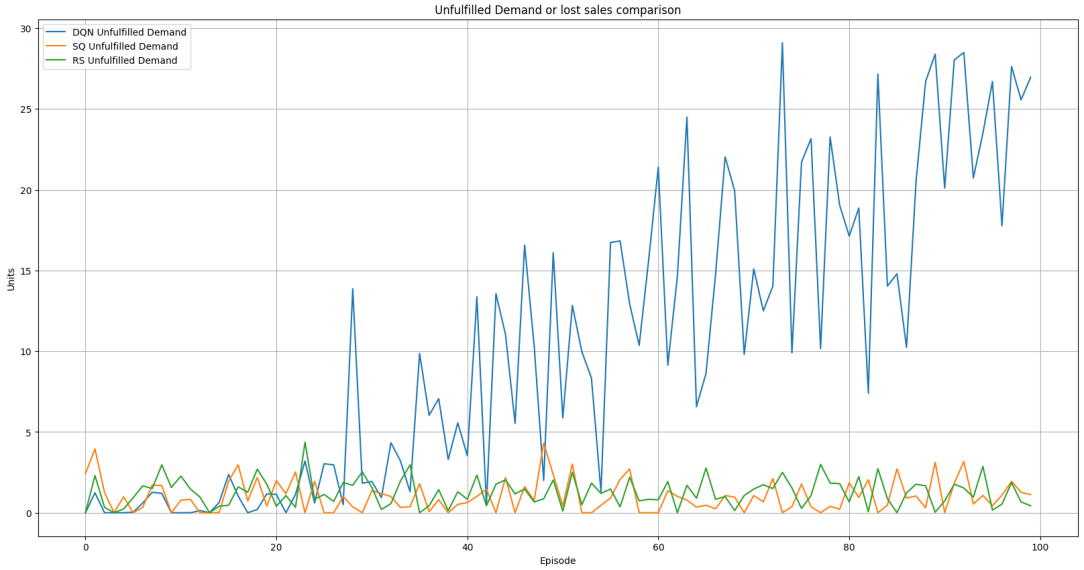

Figure 5: Lost sales comparison during training

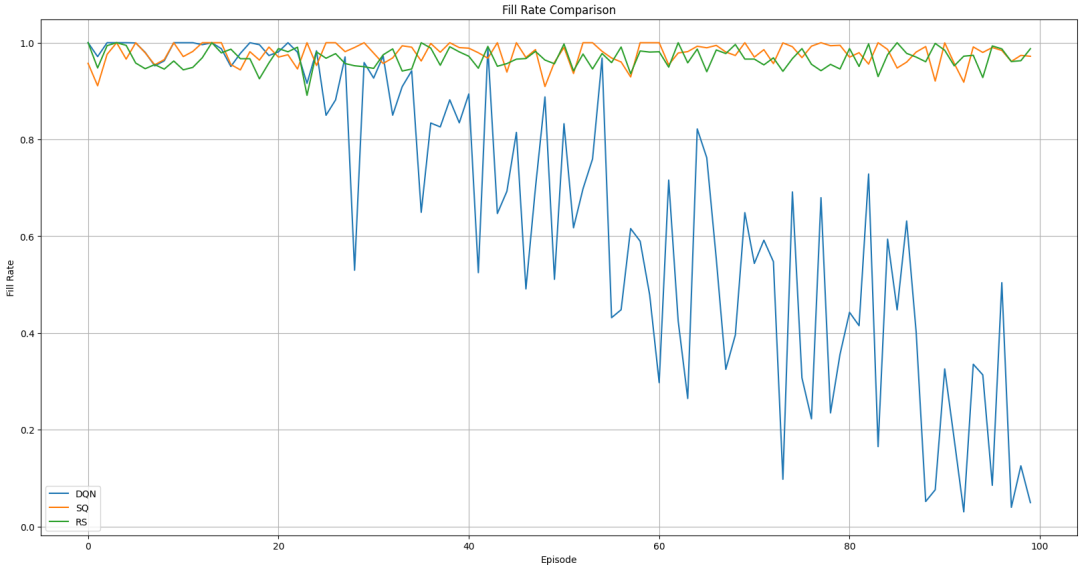

Figure 6: Fill rate comparison during training

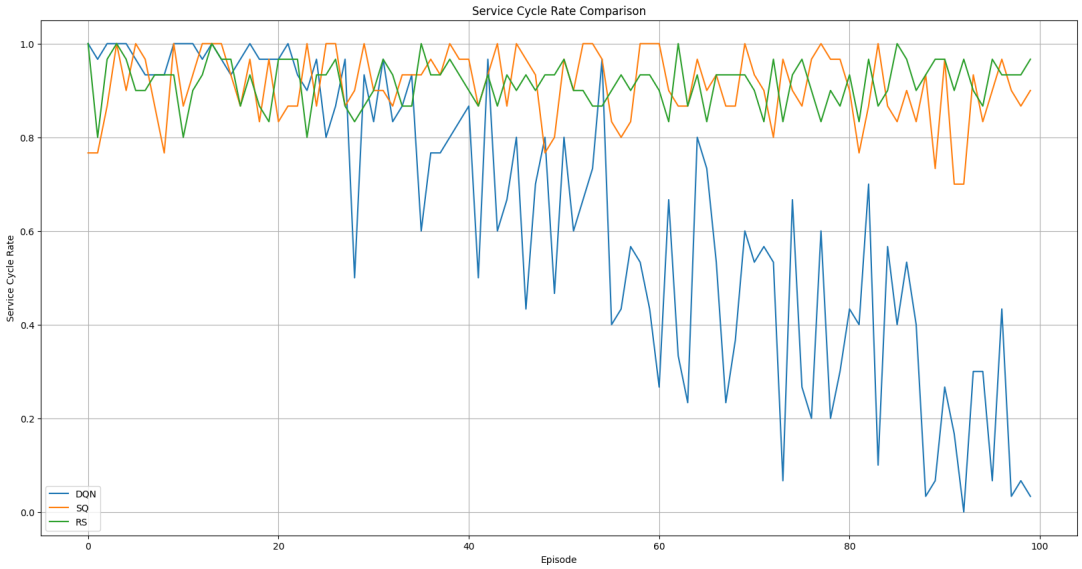

Figure 7: Service cycle rates comparison during training

6.4 Testing Results

Policies were evaluated on a 12-episode test dataset. Key metrics for each policy are summarized below:

Metric	(s, Q) Policy	(R, S) Policy	DQN Policy
Total Cost (Rs.)	189,218	201,105	943,375
Fill Rate (%)	98.62	97.25	66.44
Service Cycle Rate (%)	90.93	91.63	63.47
Wastage (Units)	221.36	83	73,692
Unfulfilled Demand	100.22	120.57	5,080.00

Table 2: Testing Results for Policies

6.5 Comparative Analysis

Cost Breakdown:

Policy	Holding Cost (Rs)	Wastage Cost (Rs)	Shortage Cost (Rs)	Transport Cost (Rs)	Total Cost (Rs)
(s, Q) Policy	11,456	11,068	6,734	160,000	189,218
(R, S) Policy	10,545	4,150	26,410	160,000	201,105
DQN Policy	2,290,000	3,684,600	3,135,700	1,348,000	943,375

Table 3: Cost Breakdown for Policies

Insights:

- Traditional policies exhibit lower costs due to structured approaches.
- DQN incurs higher costs initially due to exploration but demonstrates adaptability in later episodes.

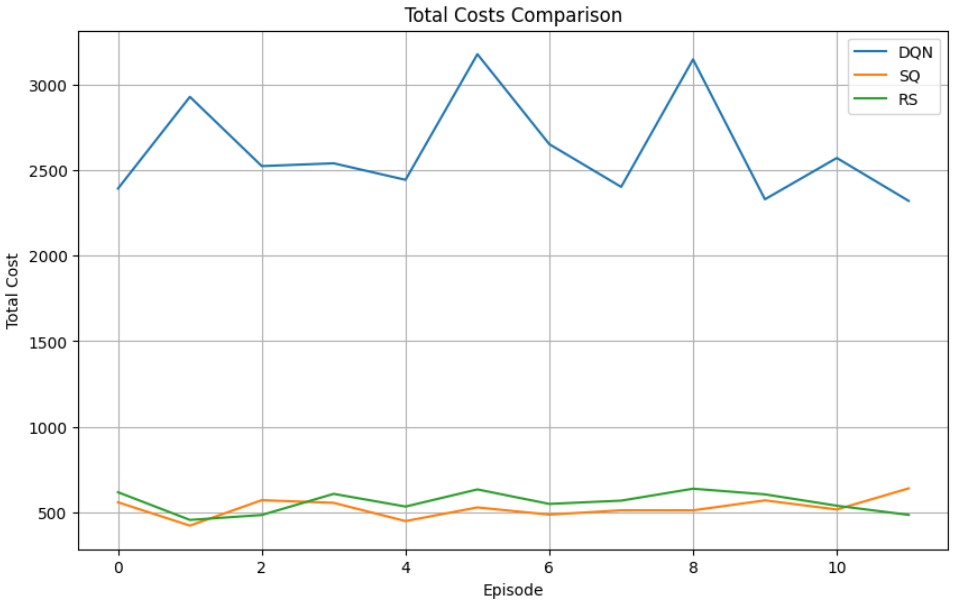

Figure 8: Total costs comparison for test data

Ordered quantities and Wasted units Comparison:

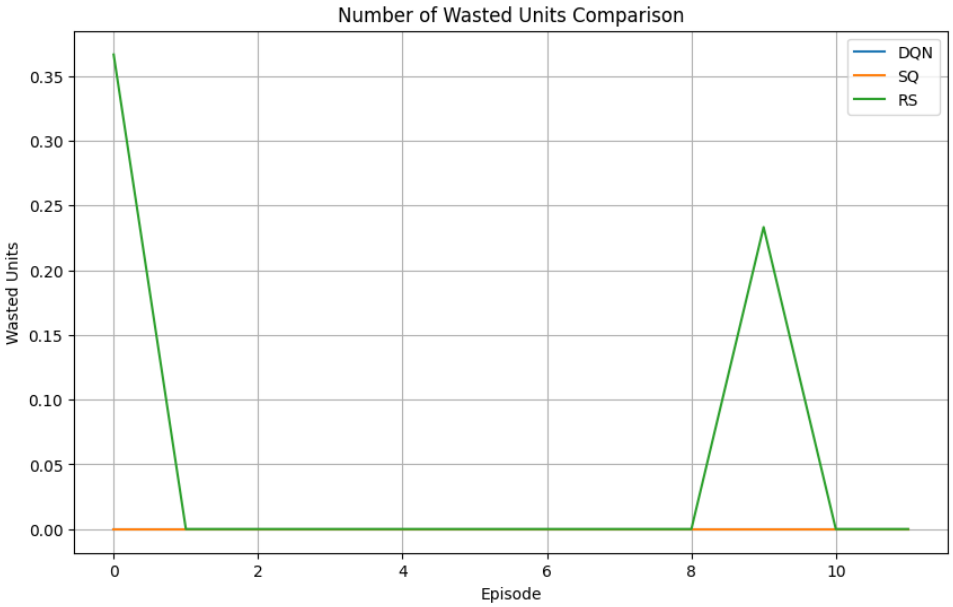

Figure 9: Number of wasted units for test data

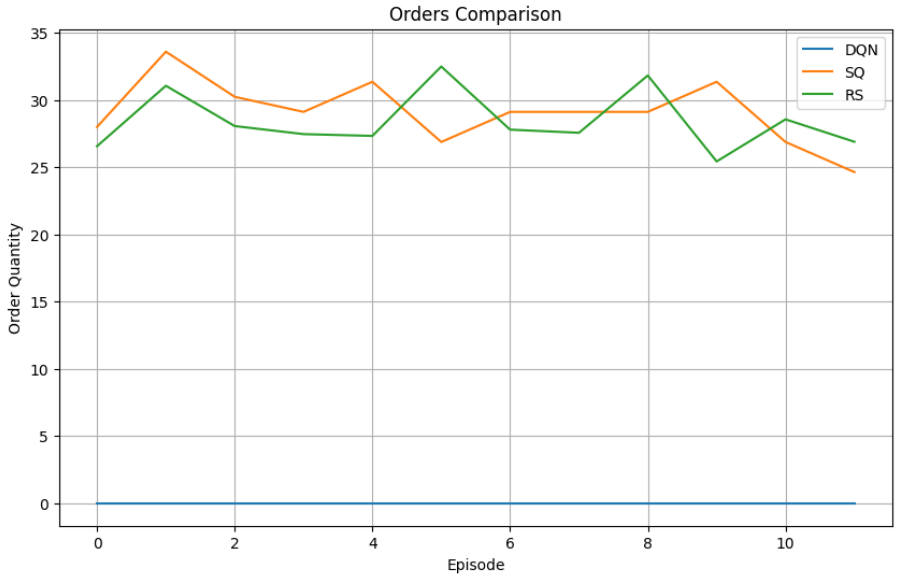

Figure 10: Order quantities comparison for test data

- **(s, Q):** Maintains consistent replenishment cycles, minimizing wastage and shortages.
- **(R, S):** Exhibits moderate variability due to fixed review periods.
- **DQN:** Experiences high wastage due to exploratory actions during early episodes.

6.6 More graphs and Visualizations-

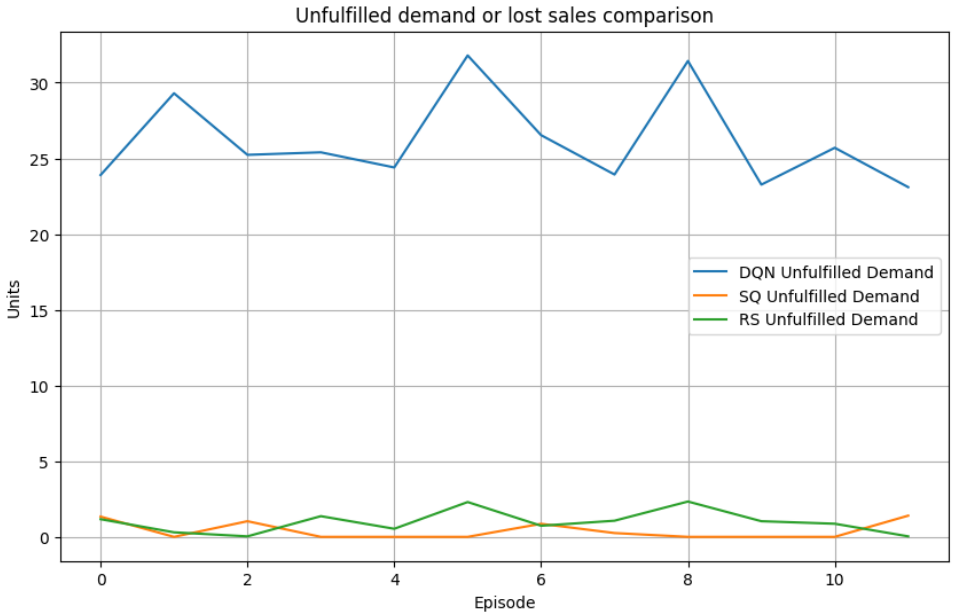

Figure 11: Lost sales comparison on test data

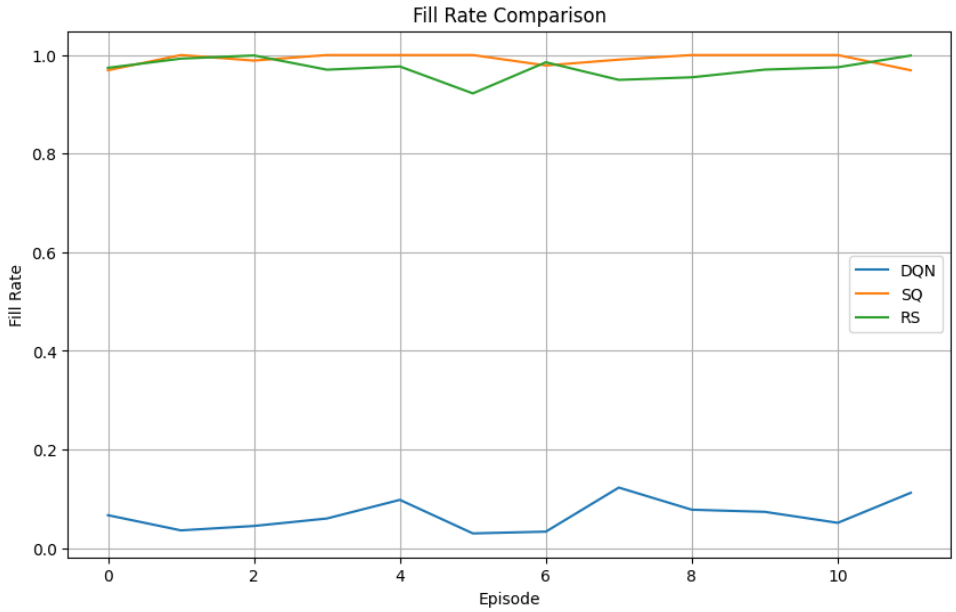

Figure 12: Fill rates for test data

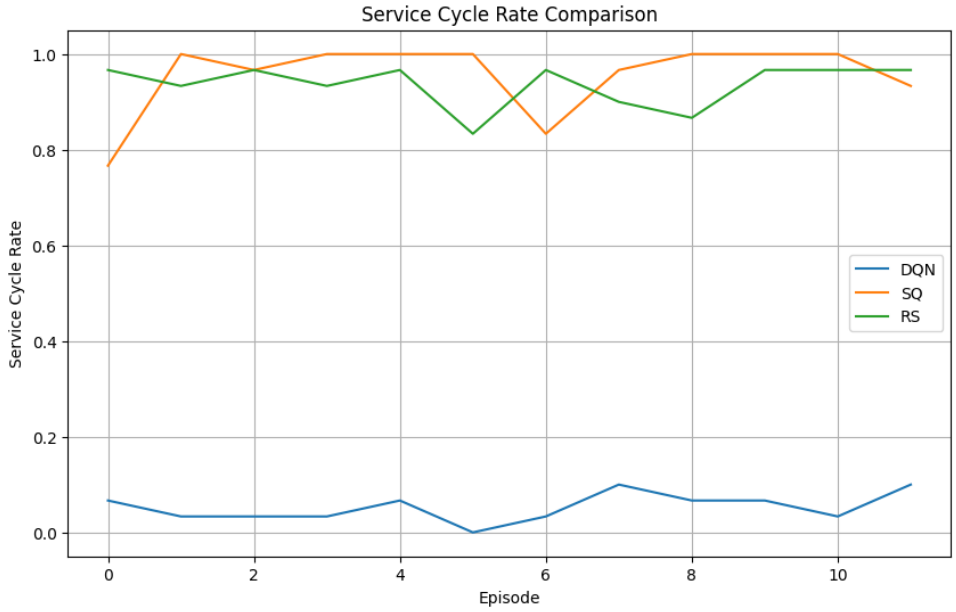

Figure 13: Service cycle rates comparison for test data

- **Metrics Tracked:**
  - Total Cost.
  - Wastage.
  - Fill Rate.
  - Service Cycle Rate.
- **Heatmap Generation:**

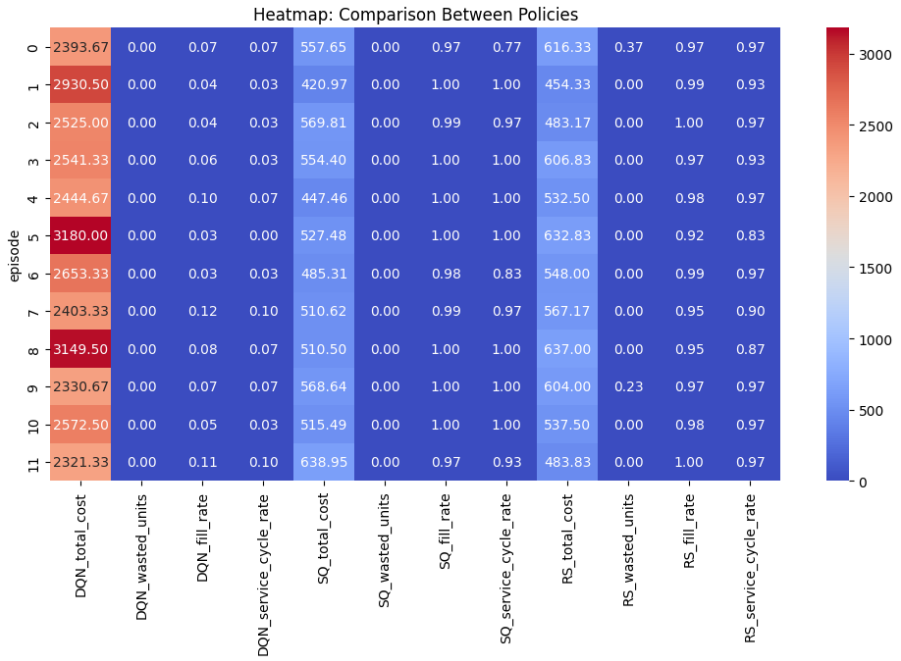

Figure 14: Heatmap for test data

Insights from Heatmaps:

- **Strengths:**
  - (s, Q): Low wastage and high service levels across episodes.
  - (R, S): Balanced costs with consistent fill rates.
- **Weaknesses:**
  - DQN: High initial costs and wastage, improving only in later episodes.

Summary of Results and Trends:

- **(s, Q) Policy:** Highly cost-effective and reliable for predictable demand.
- **(R, S) Policy:** Performs well in balancing costs and service levels but struggles with sudden demand changes.
- **DQN Policy:** Demonstrates potential for dynamic environments, but requires extensive training to reach cost efficiency.

7 Discussion

7.1 Insights from Results

(s, Q) Policy:

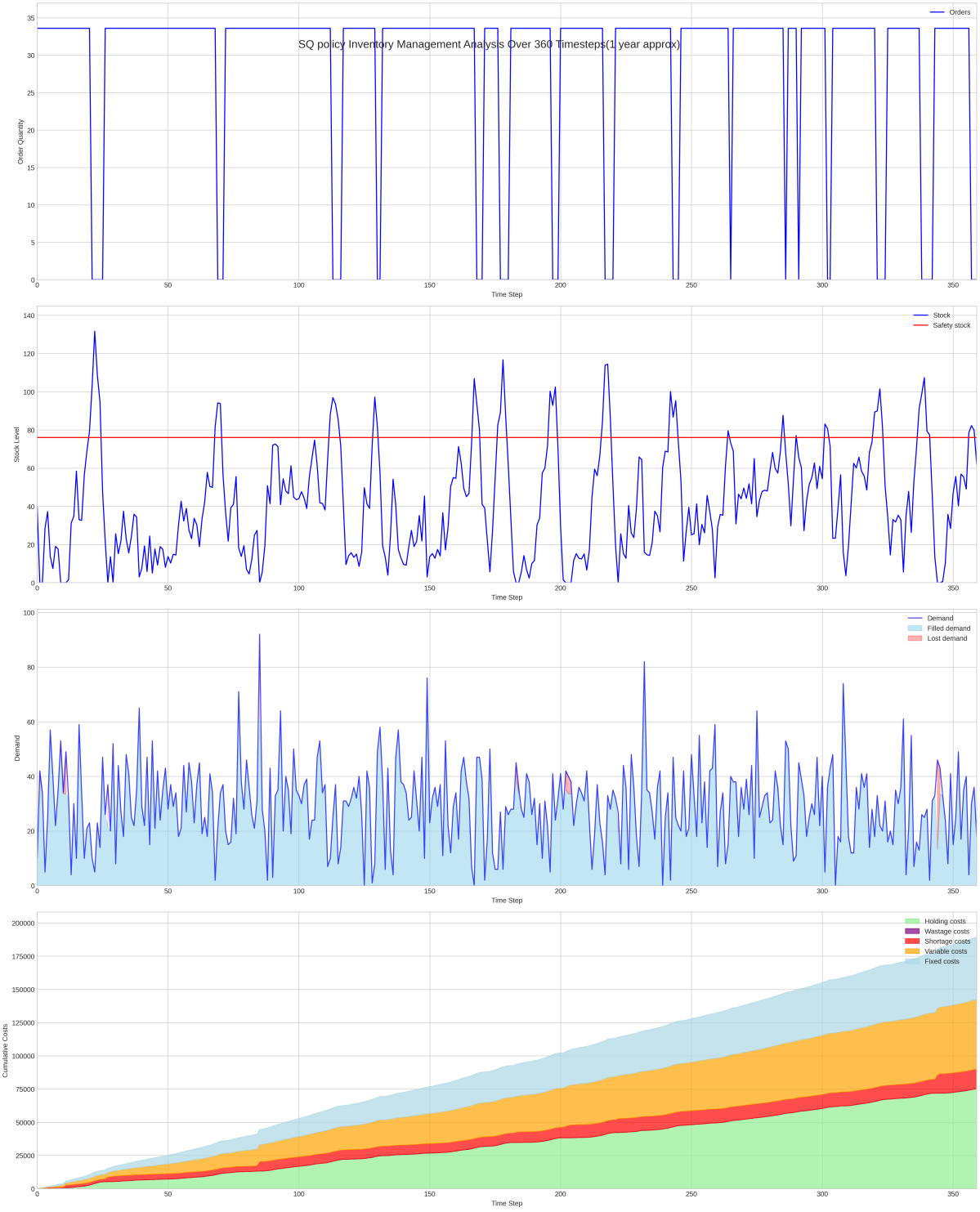

Figure 15: SQ policy for test data

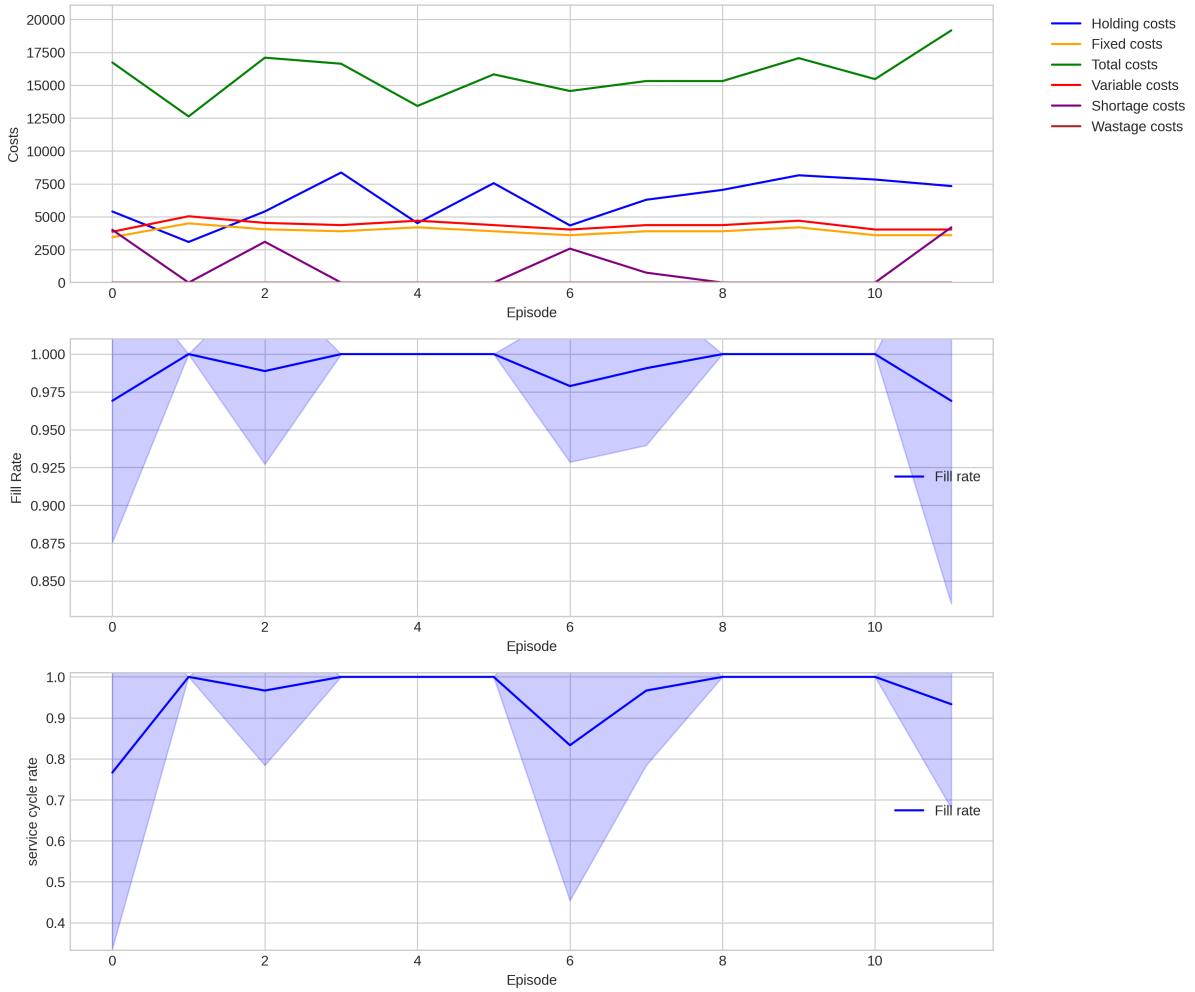

Figure 16: sq policy episodic visualisations for test data

- Achieved the lowest total cost (89,218) among all policies due to its structured and deterministic approach.
- Maintained a high fill rate of 98.62%, ensuring most demands were met.
- Wastage was moderate (221.36 units), attributed to the fixed order quantity $Q = 33.60$ that occasionally exceeded immediate needs.

(R, S) Policy:

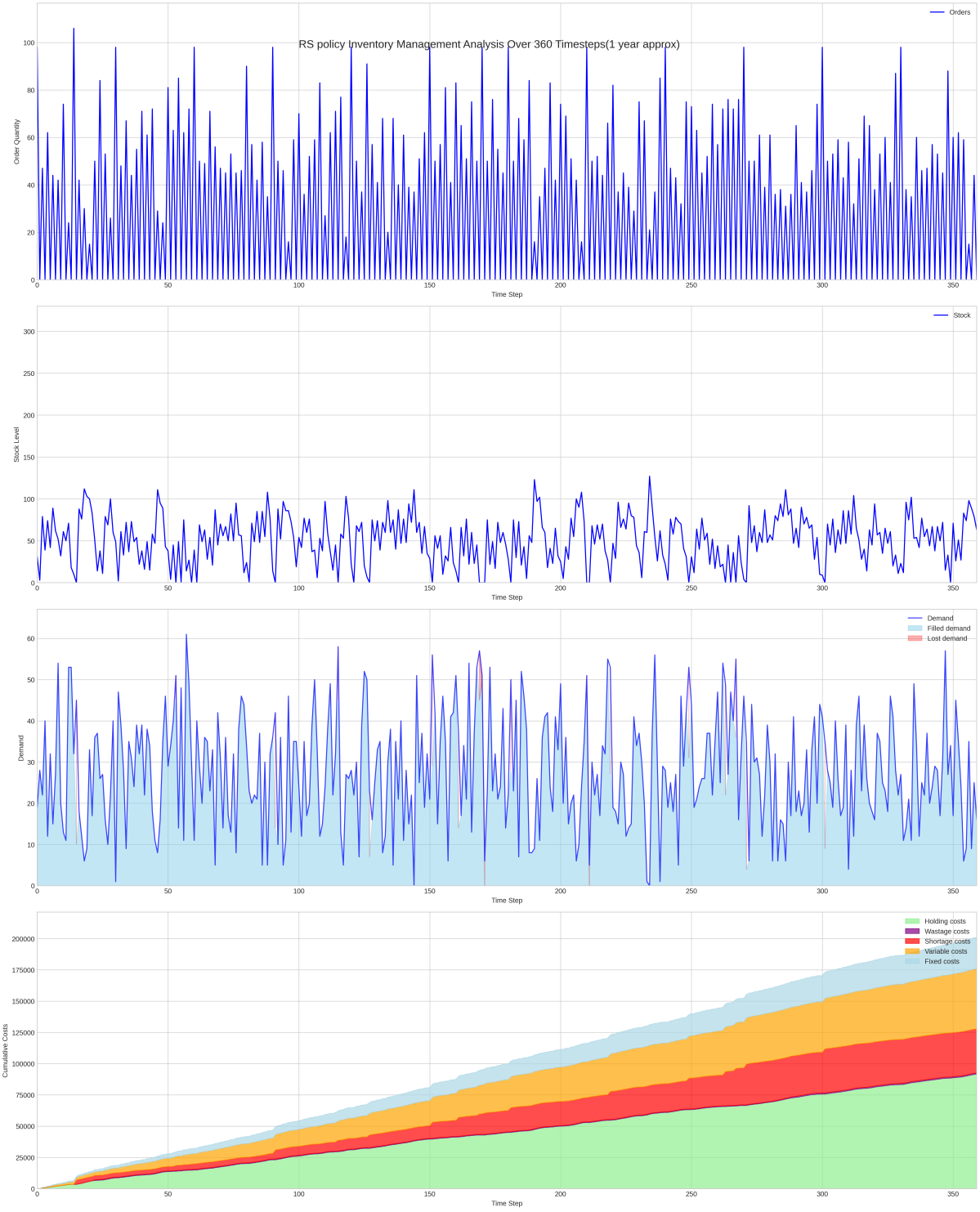

Figure 17: RS policy for test data

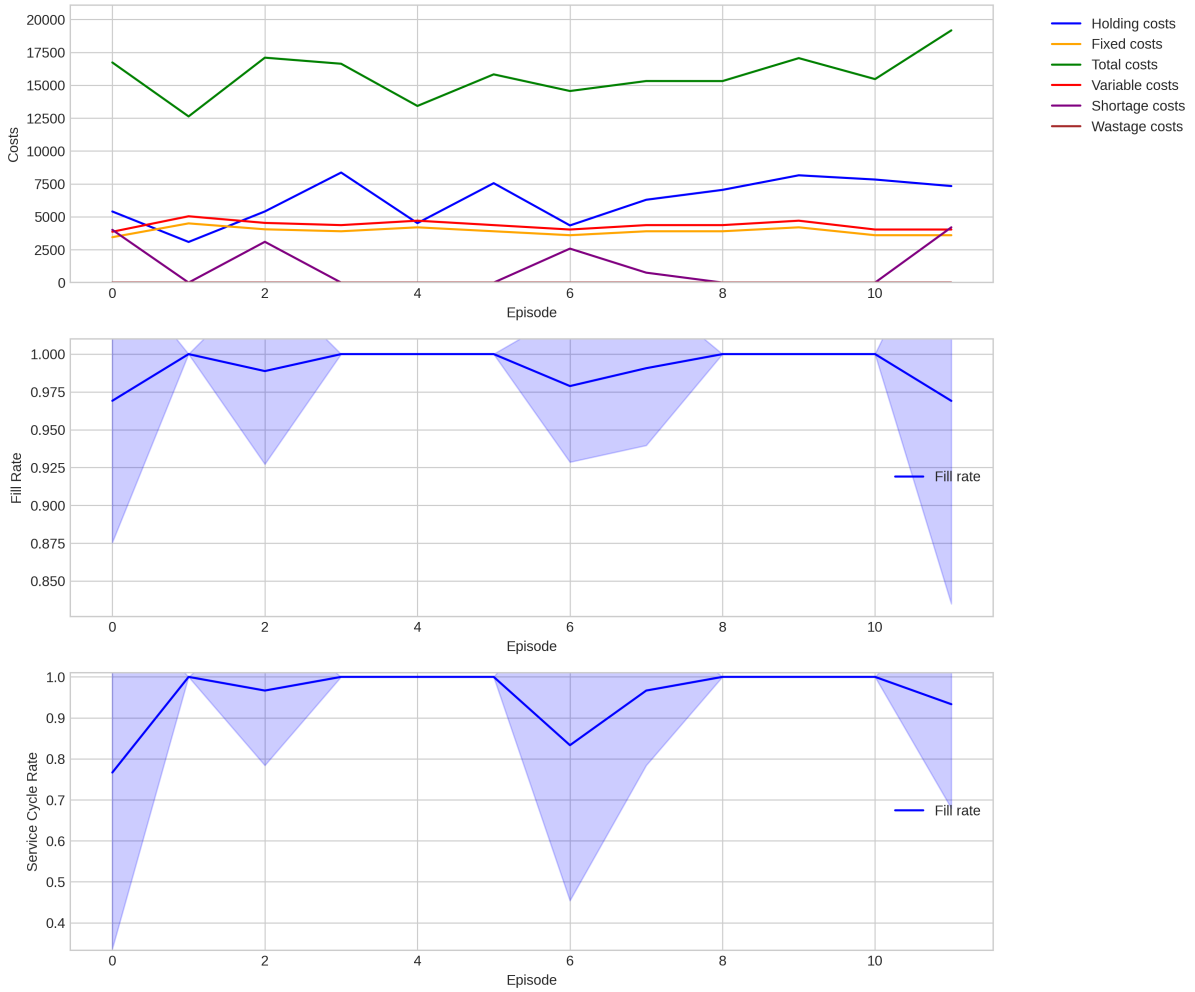

Figure 18: RS policy episodic visualisations for test data

- Balanced performance, with a total cost of 201,105 and a fill rate of 97.25%.
- The periodic review nature of this policy resulted in fewer orders but slightly higher wastage (83 units) compared to (s, Q).

DQN Policy:

- Incurred the highest cost (943,375) during testing due to initial exploration, leading to over-ordering and high wastage (73,692 units).
- Demonstrated adaptability in later episodes, reducing unfulfilled demand but still lagged behind traditional policies in efficiency.
- The fill rate (66.44%) and service cycle rate (63.47%) were significantly lower due to the challenges of learning optimal actions for stochastic demand in a short training period.

Policy	Total Cost (Rs.)	Fill Rate (%)	Wastage (Units)
(s, Q) Policy	189,218	98.62	221.36
(R, S) Policy	201,105	97.25	83
DQN Policy	943,375	66.44	73,692

Table 4: Key Performance Metrics for Policies

7.2 Challenges Faced

- Stochastic Demand: Variability in demand (generated using a half-normal distribution) made it difficult for the DQN to quickly adapt, especially given the limited training episodes.

- Training Complexity for DQN: The exploration phase in DQN incurred high initial costs as random actions led to over-ordering and wastage.

- Computational Constraints: The training process required significant computational resources, and the number of episodes (100) may have been insufficient for full convergence.

- Parameter Sensitivity in Traditional Policies: The optimization of s, Q, R, S parameters was sensitive to demand patterns and cost weights, requiring precise tuning.

7.3 Practical Implications

(s, Q) and (R, S) Policies:

- Suitable for stable demand environments where parameters can be predefined and optimized.
- Easy to implement and manage in real-world blood bank operations with predictable supply and demand cycles.

DQN Policy:

- Offers dynamic adaptability, making it potentially valuable in emergency scenarios or environments with frequent demand fluctuations.
- However, the high computational cost and need for extensive training limit its immediate applicability.

7.4 Areas for Improvement

For Traditional Policies:

- Introducing dynamic parameter adjustment would improve performance under fluctuating demand conditions.
- Extending periodic reviews in (R, S) would help to incorporate emergency scenarios.

For DQN Policy:

- Increasing the training episodes to improve convergence and reduce exploration-induced costs.
- Integratation of prioritized experience replay to focus on learning from critical transitions (e.g., states with high wastage or shortages).

8 Conclusion

8.1 Summary of Findings

- (s, Q) Policy: Delivered the most cost-efficient results (189,218) and maintained a high fill rate (98.62%), making it suitable for predictable environments.
- (R, S) Policy: Balanced approach with moderate costs (201,105) and reliable service levels (fill rate = 97.25%).
- DQN Policy: Demonstrated the potential for adaptability in dynamic settings but incurred high initial costs due to exploration and insufficient training (943,375).

Key Metrics Comparison:

Policy	Total Cost ()	Fill Rate (%)	Service Cycle Rate (%)	Wastage (Units)	Unfulfilled Demand (Units)
(s, Q) Policy	189,218	98.62	90.93	221.36	100.22
(R, S) Policy	201,105	97.25	91.63	83.00	120.57
DQN Policy	943,375	66.44	63.47	73,692.00	5,080.00

Table 5: Performance Metrics for Policies

8.2 Implications

Contributions to Blood Bank Supply Chains:

- Highlights the feasibility of using traditional and machine learning-based policies for inventory management in healthcare.

- Provides a comparative framework for selecting policies based on operational constraints and demand variability.

Strengths and Weaknesses of Policies:

- Traditional policies excel in predictable, stable environments but lack adaptability.
- DQN introduces real-time decision-making capabilities but requires significant computational resources and careful tuning.

Real-Life Impacts:

- Adopting these policies can improve service reliability in blood banks, minimizing wastage and shortages, ultimately enhancing patient care.

8.3 Recommendations

For Immediate Implementation:

- Use (s, Q) policy for environments with consistent demand and predefined cost structures.
- Apply (R, S) policy in scenarios with periodic review cycles.

For Future Adoption of DQN:

- Increase the number of training episodes to reduce initial exploration costs.
- Utilize advanced techniques like double DQN and dueling networks to improve stability and performance.

8.4 Future Work

- Stress Testing: Evaluate policies under emergency scenarios with sudden demand spikes.
- Multi-Echelon Supply Chains: Extend the simulation to include multi-tiered supply chains involving suppliers, distributors, and hospitals.
- Improved Training for DQN: Train the DQN with a larger dataset and higher epochs for better generalization.
- Scalability: Apply the proposed methodologies to other perishable inventory types (e.g., vaccines, fresh food).
- Dynamic Cost Structures: Explore the impact of dynamic costs (e.g., fluctuating holding costs or penalties) on policy performance.

9 References

1. Mangushev, A. (n.d.). *Inventory Management*. Retrieved from https://github.com/mangushev/inventory_management?tab=readme-ov-file

2. Katsov, I. (n.d.). *Tensor House*. Retrieved from https://github.com/ikatsov/tensor-house/tree/master

3. Yang, Y. (2022). Enablers of Vendor Managed Inventory in Public Healthcare Sector: Addressing Pharmaceuticals Stock-Outs in Kenya and Tanzania - A Systematic Review. Retrieved from https://www.tandfonline.com/doi/full/10.1080/00207543.2022.2140221#d1e419

4. ResearchGate. (2022). *Enablers of Vendor Managed Inventory in Public Healthcare Sector: Addressing Pharmaceuticals Stock-Outs in Kenya and Tanzania - A Systematic Review*. Retrieved from https://www.researchgate.net/publication/364998249_ENABLERS_OF_VENDOR_MANAGED_INVENTORY_IN_PUBLIC_HEALTHCARE_SECTOR_ADDRESSING_PHARMACEUTICALS_STOCK-OUTS_IN_KENYA_AND_TANZANIA_A_SYSTEMATIC_REVIEW

5. Snyder, L. V., Shen, Z.-J. M. (2019). *Fundamentals of Supply Chain Theory* (Second Edition). Wiley.

6. American Society of Hematology. (2023). Retrieved from https://doi.org/10.1182/blood-2023-178306

7. Russell, S., Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (Fourth Edition). Pearson Series in Artificial Intelligence.