|  |  |  |
| --- | --- | --- |
|  | Contents lists available at ScienceDirect Intelligent Systems with Applications journal homepage – [www.elsevier.com/locate/smhl](http://www.elsevier.com/locate/smhl) |  |

Multi-Echelon Supply Chain Management System: Using Reinforcement Learning  
  
Pranav Ha, Sri Harshini Muppavarapua, Tarun Rachuria, [[1]](#footnote-1)\* and Nippun Kumaar A.Aa

a Department of Computer Science and Engineering Amrita School of Computing, Bengaluru, Amrita Vishwa Vidyapeetham, India

|  |  |
| --- | --- |
| ARTICLE INFO | ABSTRACT |
| **Keywords**:  Multi Echelon Supply ChainSupply chain management Reinforcement Learning Deep Reinforcement Learning  Twin Delayed DDPG (TD3) Deep Deterministic Policy Gradient (DDPG) | Supply chain management (SCM), is the process of control ling the flow of products, services, money, and information from suppliers to customers. Reinforcement learning (RL) can improve SCM by using self-learning techniques to make better decisions over time. By learning from operations, RL optimizes transportation, inventory, and production, creating a cost-efficient and time-saving mechanism. This work uses TD3algorithm to manage supply chain operations. The algorithm selects actions and computes performance metrics based on cost efficiency and delivery times. The reward function optimizes overall supply chain performance, focusing on a network of factories and warehouses. The RL agent’s performance, evaluated on parameters including cost efficiency, delivery times, profitability, accuracy, rewards, and loss metrics, achieved a profit of 6078 units. It effectively navigated supply chain complexities, enhancing profitability. To validate performance, the agent was tested on real-world Walmart e-commerce product data, using a calendar quarter and three product categories, achieving a 6.02% profit increase compared to manually calculated profit. |

1. Introduction

In today’s dynamic business environment, effective supply chain management is essential for organizations to remain competitive and profitable. With the proliferation of global markets and the increasing complexity of operations, traditional inventory management methods are inadequate for modern supply chains ([Nair PR. 2019, Nair, Prashant R., and S. P. Anbuudayasankar.2016](#_Contents_lists_available)). There is a growing interest in advanced technologies ([Krishna, G. Sai Radha, and P. Rekha.2022 , Rahimi, and et al 2020](#_Contents_lists_available)), such as reinforcement learning ([Giannoccaro and et al 2002., Kumaar, AA Nippun, and et al 2023](#_Contents_lists_available" \o "References)) which has infiltrated fields from Robotics to Natural Language Processing [(Jiang and et al 2009),](#_Contents_lists_available) to optimize supply chain management. This work aims to tackle supply chain complexities by designing and deploying a sophisticated multi-echelon supply chain management system. Leveraging reinforcement learning, this system will navigate a multi-node supply chain network, encompassing manufacturing facilities and multiple warehouses. The primary objective is to enhance profitability by efficiently orchestrating the flow of goods, minimizing inventory costs, and maximizing service levels. Transportation logistics perform a crucial role in the supply process, ensuring timely, efficient delivery of goods. Seasonal demand requires the system to control production and inventories to avoid stock outs and excess stock. Proper control of manufacturing costs is essential because optimal production costs sustain market competitiveness. Through complex algorithms and data analysis, the system is expected to decrease costs and improve processes, achieving higher effectiveness in operations. In summary, using a reinforcement learning approach for a multi echelon SCM has great potential for enhancing performance in SCM, managing the complexities of contemporary supply chains, and improving key performance indicators to support sustainable growth and competitive advantage.

1.1 The Key Contributions of this Paper

1. Developing multi-echelon end to end supply chain management system to maximize profits and minimize costs, across the Manufacturing, Shipping and Warehousing domains.
2. A Robust Agent learnt over 12 years of data which ensures long term profitability.
3. Incorporation of Deep Reinforcement Learning in order to ensure the agent stays up to date and improves itself with every new datapoint.

Section 2 of this paper contains, a review of the cutting-edge research in the domain of SCM, along with the gaps in them which this research tries to resolve. Section 3 describes the problem statement that this research deals with. Section 4 contains the dataset that was used for this purpose, and Section 5 covers the proposed methodology. Section 6 contains the results and analysis, while Section 7 contains the conclusion regarding the method of implementation, as well as future work.

**2 Related works**

For today’s business world, SCM is used to manage inventory throughout stages to improve the operation and the return. Traditional systems ([Ravulapati, Kiran Kumar, Jaideep Rao, and Tapas K. Das. 2008](#_Contents_lists_available" \o "References)) have problems with demand variation and lead time. New technologies and optimization meth ods are solutions to these problems. This review aims at discussing how DRL and optimization models enhance multi-stage supply chains, based on three articles ([Chaharsooghi, S. Kamal, Jafar Heydari, and S. Hessameddin Zegordi 2008](#_Contents_lists_available" \o "References)) showing the effectiveness of the strategies. Exploring DRL for multi-echelon supply chains ([Shar, Ibrahim El et al. 2022](#_Contents_lists_available)), DRL is used to address fluctuating demands in multi-echelon supply chains. Shar’s GymSC simulation environment proves that DRL performs better than traditional methods in un predictable scenarios, this study highlights DRL’s effectiveness and provides a platform for further research and algorithm testing. Enhancing supply chain management through a VMI model (Modares, Azam et al. 2022[Contents lists available at ScienceDirect](#_Contents_lists_available)) focuses on selecting optimal retailers and ensuring vendor and retailer reliability. The study introduces a redundancy allocation problem (RAP) to boost vendor reliability and uses an analytical hierarchical process (AHP) for retailer selection. To handle the model’s complexity, genetic algorithm (GA) and particle swarm optimization (PSO) methods are used, proving the model’s effectiveness with a case study in the electronic supply chain. The use of DRL to SCIM ([Stranieri, Francesco, and Fabio Stella. 2022)](#_Contents_lists_available) presents a new approach to balancing the inventory quantities of products in the warehouses. The paper presents a mathematical model of a stochastic two-echelon supply chain and an open-source repository for DRL methods. Extensive testing showed the PPO algorithm is highly adaptable, outperforming traditional reorder policies and highlighting DRL’s potential in solving complex SCIM challenges and analyzing the

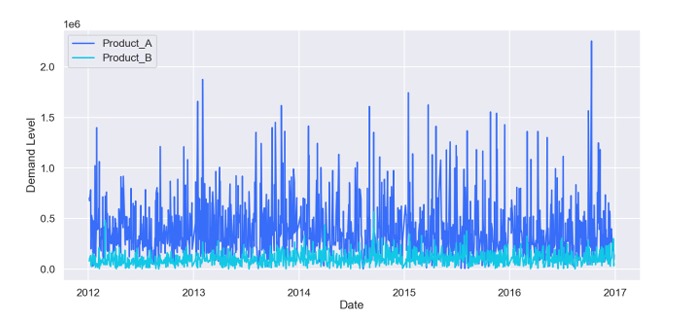
inventory level for demand uncertainty in the VUCA world ([Raghuram, P., Bhupesh, S., Manivannan, R., Anand, P. S. P., & Sreedharan, V. R. (2022)](#_Contents_lists_available)). The current landscape of RL based optimizations in SCM, by exploring and understanding the research papers reviewed above is expansive and dynamic. This survey contributes unique insights and identifies innovative solutions in the field. This Survey underscores the inherent advantage of foundational aspects of the use of RL to optimize the supply chain and enhance profitability.

**3 Problem statement**

The goal is to create and use a system that manages the supply chain using reinforcement learning in a detailed simulation. This involves moving goods between a factory and several storage locations to increase profits. The system must han dle transportation and delivery, adapt to seasonal demand changes, and reduce manufacturing costs, ensuring flexibility in a competitive market.

**4 Dataset Description**

To enable effective learning for an RL agent, a comprehensive dataset spanning roughly 12 years (from May 1, 2012, to March 31, 2024) was used to model the demand patterns of two products, Product A and Product B, incorporat ing Gaussian Noise for variability mirroring real-world fluctuations. Figure 1 depicts the demand characteristics: Product A (blue line) shows significant vari ability with peaks up to 2,260,000 units, while Product B (green line) displays a more stable pattern, peaking at 568,000 units. These observations underscore Product A’s volatility and Product B’s stability over time. Statistical analy sis reveals Product A’s average demand of 390,000 units (standard deviation: 287,000) across 1210 data points, and Product B’s average demand of 101,405 units (standard deviation: 64,071) across 1207 data points. This dataset provides valuable insights into the distinct demand behaviors of the two products, crucial for training RL agents to optimize supply chain decisions effectively. To further enhance the dataset’s utility, various machine learning models were tested to ensure robustness in capturing the nuanced demand patterns of Product A and Product B. This included evaluating neural network architectures and time series forecasting techniques to accurately predict future demand fluctuations.



**Fig.1.** Data Distribution Visualized

**5 Proposed Methodology**

To implement a profit-maximizing agent using the Twin Delayed DDPG (TD3) algorithm, follow these steps: define the reward function, the states of the supply chain, and possible actions to define the environment. Teach the TD3 agent in this environment to find the best policies to follow. It is also important to assess and confirm the effectiveness of the agent’s performance after the training in relation to the set goals. Use the trained agent to operate the supply chain and make decisions to achieve the maximum profit.

**5.1 Environment**

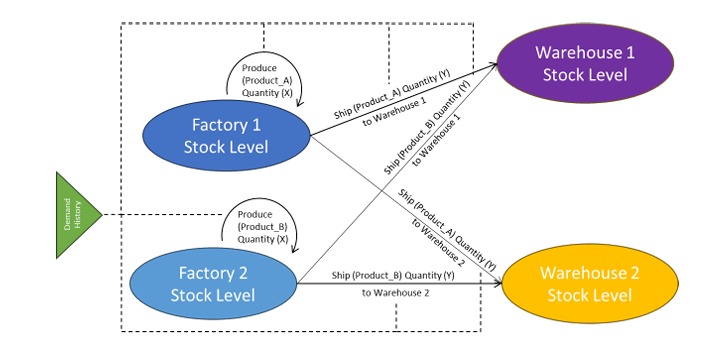
The environment is where the agent interacts and learns its task. However, it is defined by the State Space, Actions, Rewards as well as Penalties acting as its parameters. Supply chain environment is integrated, compatible with the OpenAI Gym interface, as this allowed for the easier integration with multiple reinforcement Learning Algorithms. The components simulate a real-life dynamic supply chain and mimic its complexities. It consists of three major parts: Factory, Warehouse, and Transportation.

**Factory** The Manufacturing hub of the product, the factory is where the products are made it also has a central warehouse where the produced products are stored before being shipped to the subsequent Warehouses. It has 2 major parameters are defined as**:**

1. Production Capacity: The factory has a defined production capacity, representing the maximum number of units it can produce in each time step.
2. Inventory Levels: The factory maintains inventory, which includes raw materials and finished goods.

The inventory levels fluctuate based on production and shipments to warehouses.

**Warehouse** The Warehouse is where the products are received from the factory and stored. It has 3 major parameters:

1. Storage Capacity: Each warehouse has a limited storage capacity that con strains the maximum inventory it can hold.
2. Inventory Levels: Warehouses store the goods received from the factory. The inventory levels at each warehouse change based on shipments received and customer demand fulfillment.
3. Demand Fulfillment: Warehouses fulfill customer demands, which are variable and can change over time based on historical patterns and random fluctuations.

**Transportation** The Transportation has only one parameter, which involves the cost of shipping from the Factory to the Warehouse:

1. Shipping Costs: Costs associated with transporting goods from the factory to various warehouses. These costs are influenced by the distance and volume of goods shipped.

**5.2 State Space**

To define the state space S for a supply chain management environment, the pro cess includes considering all relevant parameters representing the current system state. Figure 3 shows a more complex configuration with multiple factories and warehouses, expanding the state space to include the stock levels at Factory 1, Factory 2, Warehouse 1, Warehouse 2, and the demand history. These parame ters provide the agent with a comprehensive understanding of the supply chain’s conditions to make informed production and distribution decisions.

The state space St at time step t can be represented as:

*St = (Ft,{Wi,t}n i=1,{Di,t−j}n,k i=1,j=1, t) (1)*

The state space S can be described as a multidimensional vector incorporating the following elements:

1. Factory Stock Level (Ft): The current inventory level at the factory at time step t.
2. Warehouse Stock Levels (Wi,t): The inventory level at warehouse i at time step t.
3. Demand History (Di,t−k): Historical demand data for each warehouse i over the past k time steps.
4. Time Step (t): The current time step within the episode
5. n indicates the number of different products or items in the inventory system being analyzed.
6. k often denotes a parameter that could represent time periods, different scenarios, or other dimensions of the model.
7. i and j are indices that iterate over the different items and possibly time periods or lagged periods, respectively.

**Fig.2.** State Space Overview for training

**5.3 Actions**

Actions are specific decisions and tasks that are taken in ordering materials, controlling inventory, or planning the shipment. They are carried out to keep the business running efficiently and deliver on its promise to meet customer needs.

The action space At at time step t can be represented as:

At =(Pt,{Si,t} n i=1) (2)

1. Production Level(Pt) : The quantity of goods to produce at the factory.
2. Shipments to Warehouses(Si,t) : Quantities of goods to ship from the factory to each warehouse.
3. n represents the total number of different items, products, or components in the supply chain.
4. i is an index that goes from 1 to n, used to refer to each specific item or component in the list.

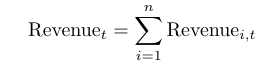
**5.4 Rewards and Penalties**

The reward function guides the agent’s decision-making process by providing feedback on the appropriateness of its actions.

The reward function Rt at time step t can be represented as:

Rt =Revenuet −Costt −Penaltiest (3)

5.4.1 Parameters and Components

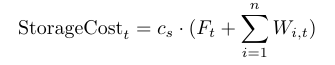
1. Revenue from Sales (Revenuet):

where Revenue i,t is the revenue from sales at warehouse i at time step t.

1. **Costs (Costt):**
2. Production Costs (ProductionCostt):

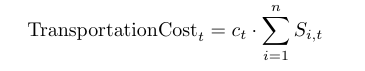
ProductionCostt = cp · Pt

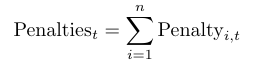
where cp is the cost of production per unit and Pt is the quantity of goods produced at the factory at time step t.

1. Storage Costs (StorageCostt):

where Wi,t is the inventory level at warehouse i at time step t, Ft is the inventory level at the factory, and cs is the cost of storage per unit.

1. Transportation Costs (TransportationCostt):

where ct is the cost of transportation per unit and Si,t is the quantity of goods shipped to warehouse i at time step t.

1. **Penalties for Stockouts (Penaltiest):**

where Penaltyi,t is the penalty for stockout at warehouse i at time step t.

Penaltyi,t = pi · max(0,Di,t − Wi,t) (10)

where Wi,t is the inventory level at warehouse i at time step t, Di,t is the demand at warehouse i at time step t, and pi is the penalty cost per unit of unfulfilled demand.

The reward function Rt balances the following:

* 1. Revenue from sales Generated by fulfilling customer demands at the ware houses.
  2. Production costs Associated with manufacturing goods at the factory.
  3. Storage costs Incurred due to maintaining inventory at the factory and ware houses.
  4. Transportation costs Arising from shipping goods from the factory to ware houses.
  5. Penalties for stock outs Penalizing instances where demand exceeds available inventory.

The trained model is deployed on the data to identify and verify the effective gains made by the use of the RL Agent.

**5.5 Twin Delayed DDPG (TD3) algorithm**

The TD3 is an advanced version of the DDPG algorithm that is enhanced and developed to be more stable, efficient, and effective in solving RL problems with continuous and dynamic action space. The TD3 algorithm has several advantages that make it appropriate for this study, such as double-critical networks, delayed policy updates, target policy smoothing, and double-Q trimmed learning.

**Training TD3 Algorithm** The TD3 agent is trained to achieve an optimal strategy for supply chain optimization before deployment to verify improve ments. Key hyperparameters crucial for its performance in the supply chain are detailed in Table 1 that include the critic and actor learning rates for stable policy updates, batch size for efficient learning, discount factor for long-term profitability, policy noise for exploration, and policy update frequency for sta bility. These parameters enable effective supply chain management, aiming for optimal outcome.

**5.6 Steps for Target Q-value and Policy Update**

**T**he Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm is a model-free, off-policy actor-critic method specifically designed to improve stability and performance in reinforcement learning tasks with continuous action spaces. *Table I* outlines the core steps involved in the TD3 algorithm, each addressing key limitations present in earlier deterministic policy gradient methods. The algorithm begins by computing the target Q-value using two critic networks and selecting the minimum of the two estimates. This approach helps mitigate the common issue of overestimation bias in Q-learning. Additionally, Gaussian noise is clipped and added to the target action to introduce regularization and promote smoother policy updates. The critic networks are then updated by minimizing the mean squared error between the predicted Q-values and the target values, ensuring accurate estimation of value functions. Unlike standard methods that update the actor and critic simultaneously, TD3 employs delayed policy updates, meaning the actor network is updated less frequently (every d steps). This delay helps stabilize learning by allowing the critic to converge before it starts influencing the actor. The actor update maximizes the estimated Q-value from the first critic, guiding the policy toward actions with higher expected returns. Finally, TD3 updates its target networks for both the actor and the critics through soft updates using Polyak averaging, which helps maintain consistent learning targets and prevents abrupt shifts in network parameters. Altogether, these steps—summarized in *Table I*—make TD3 a powerful and reliable approach for tackling complex control problems in continuous domains.

**Policy Network** To effectively learn and identify the complex patterns and dynamics of the supply chain environment a neural network is used to update the policy and values. The network consists of 2 fully connected hidden layers of 300 and 400 units respectively. This is followed by a Relu activation function

**Table 1**

Steps for Target Q-value and Policy Update

|  |  |
| --- | --- |
| **Step** | **Description** |
| Step 1: Compute the target Q-value | y = r + γ \* min₍ᵢ₌₁,₂₎ Qθ′ᵢ(s′, πϕ′(s′) + ϵ) where ϵ ∼ clip(N(0, σ), −c, c) |
| Step 2: Update the critic | Minimize the loss: L(θᵢ) = E[(Qθᵢ(s, a) − y)²] |
| Step 3: Delayed policy update (if t mod d = 0) | ϕ ← ϕ − α ∇ϕ J(ϕ) where J(ϕ) = E[Qθ₁(s, πϕ(s))] |
| Step 4: Update target networks | θ′ᵢ ← τ θᵢ + (1 − τ) θ′ᵢ ϕ′ ← τ ϕ + (1 − τ) ϕ′ |

**Table 2**

Hyperparameters in Supply chain management

|  |  |  |  |
| --- | --- | --- | --- |
| **Hyperparameter** | **Value** | **Purpose** | **Application in Supply Chain Management** |
| Critic Learning Rate | 0.001 | Controls critic updates for stable Q-value learning. | Helps accurately evaluate future rewards of inventory and shipping decisions. |
| Actor Learning Rate | 0.0001 | Facilitates gradual policy adaptation, avoiding destabilizing updates. | Enables fine-tuning of production and distribution decisions over time. |
| Batch Size | 64 | Balances efficiency and learning effectiveness by determining sample size per update. | Ensures effective learning from diverse demand and supply scenarios. |
| Discount Factor (Gamma) | 0.99 | Prioritizes long-term rewards in decision-making processes. | Focuses on long-term profitability and sustainable inventory levels. |
| Policy Noise (Std Dev) | 0.2 | Encourages exploration by adding noise to policy during training. | Helps explore different production and shipment strategies for effectiveness. |
| Policy Update Frequency | Every 2 critic updates | Ensures stable policy learning by updating less frequently than critic networks. | Maintains stable policy updates for reliable inventory and distribution management. |

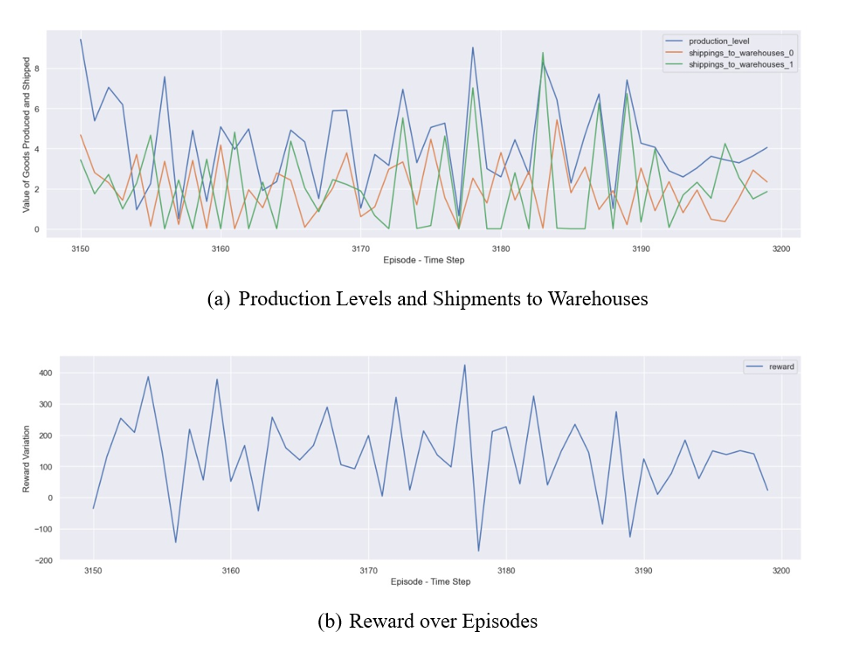
The Table 2 summarizes key hyperparameters used in the TD3 algorithm and their relevance to supply chain management applications. Each parameter plays a crucial role in stabilizing and optimizing the learning process. For example, the critic and actor learning rates control how quickly the networks adapt to new information, while the batch size balances learning efficiency. The discount factor emphasizes long-term profitability, and policy noise encourages exploration of different strategies. Lastly, updating the policy less frequently than the critic ensures more stable and reliable learning for effective inventory and distribution management.

**6 Results**

The Agent was used to predict optimize a scenario where 2 products Product A and Product B were produced at the factory, and it had to supply 2 warehouses with a mixture of both the products based on demand.

The assumptions made for this scenario were:

1. Factory produces a product with a fixed cost
2. Warehouses have a fixed capacity
3. Tangible but fixed transportation and storage costs
4. Any unfulfilled demand is carried over to the next time step with a penalty.

**6.1 Production & Distribution Management and Random Optimisation**

The TD3 agent successfully coordinated the inventory of ProductA and Pro ductB in the factory and warehouses by controlling production and shipment according to demand and cost. In Figure 3(a), it illustrates the Production Lev els and Warehouse Shipments. The x-axis is ‘Episode-Time Step’ and the y-axis is ‘Stock’. The green line shows that factory stock reduces from 2 to 0 whereas the blue line shows that the warehouse stock rises from 0 to nearly 2, indicating stock transfers between them. The performance of the agent over time is presented in Figure 3(b), where the agent’s performance increases over time gradually. The reward function was the sales revenue that was compared to the cost of production, storage, transportation, and penalties for unmet demand. Thus, this function was maximized by the agent as it showed better decision making and efficiency in its learning process. This improvement also proves that reinforcement learning is useful in enhancing the supply chain operations.

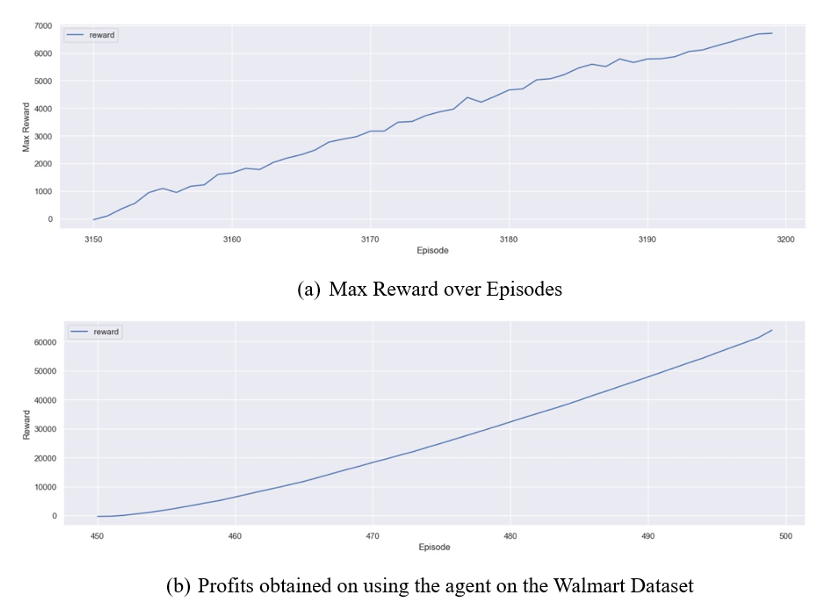
**Fig.3**. TD3 agent training performance

**6.2 Profit**

The trained TD3 agent demonstrated a strong ability to maximize profits in a simulated supply chain environment, achieving a reward of 6,078, as illustrated in Figure 4(a). This indicates the agent’s capability to navigate complex supply chain challenges and make strategic decisions that significantly improve profitability.

To evaluate the agent's real-world applicability, the “Walmart E-commerce Product Data” from Kaggle was used for validation. Focusing on a specific calendar quarter and analyzing three product categories, a manually calculated profit of $60,344.57 was established as a baseline.

The TD3 agent, trained using 12 years of synthetically generated data, surpassed this baseline and achieved a profit of $63,981.76, representing a 6.02% increase, as shown in Figure 4(b). However, when the training dataset was limited to just 100 days, the agent's profit dramatically decreased to $6,330, marking an 89.51% decline in performance over the same quarter.



**Fig.4.** TD3 agent testing performance

**7 Discussion and Conclusion**

This study presents a reinforcement learning-based approach for optimizing multi-echelon supply chain management using the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm. The developed agent demonstrated the ability to efficiently manage production and distribution operations by maintaining optimal stock and production levels across the supply chain network.During training, the TD3 agent achieved a profit of 6,078 units, reflecting its learning effectiveness and reward maximization capability. For real-world evaluation, the Walmart E-commerce Product Data was employed. The agent, when tested on a specific calendar quarter and three product categories, achieved a profit of $63,981.76, representing a 6.02% increase over the manually calculated profit of $60,344.57. This underscores the agent’s capacity to generalize and perform in real-world settings, especially when trained on comprehensive datasets.

However, when the training data was restricted to only 100 days, the agent's performance sharply declined, producing a profit of just $6,330, an 89.51% decrease, indicating that limited data significantly affects policy effectiveness. These findings highlight the critical importance of long-term, diverse, and contextually rich data for training reinforcement learning agents in supply chain applications.

Overall, the results validate the applicability of TD3 in real-world supply chain scenarios, showcasing its potential to drive profitability and responsiveness through autonomous decision-making. Future research could explore expanding the model to more complex environments with additional variables such as transportation delays, market demand fluctuations, and dynamic pricing. Moreover, further refinement of the reward function to align more closely with business objectives may enhance decision quality and operational alignment.

**CRediT authorship contribution statement**

**Pranav Ha**: Conceptualization, Methodology, Software, Visualization, Investigation, Writing – original draft.  
**Sri Harshini Muppavarapu**: Conceptualization, Methodology, Investigation, Validation, Visualization, Writing – original draft, Writing – review & editing.  
**Tarun Rachuri**: Conceptualization, Methodology, Investigation, Validation, Visualization, Writing – review & editing.  
**Nippun Kumaar A.A.**: Supervision, Project administration, Resources, Writing – review & editing.

**Declaration of competing interest**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Acknowledgments**

The authors would like to thank three reviewers for their sugges tions that helped improve the clarity and quality of the manuscript.

**Appendix**

The SCM codes for the given application and proposed method ology in this study can be found here:

**Data availability**

Study data is available via the given link in the Appendix

**References**

1. Nair PR. An overview of ICT tools for supply chain management. IEEE India Info. 2019;14(2):107-11.

2. Nair, Prashant R., and S. P. Anbuudayasankar. ”An Investigation on the benefits of ICT deployment in Supply Chain Management (SCM).” Indian Journal of Science and Technology (2016).

3. Krishna, G. Sai Radha, and P. Rekha. ”Food supply chain traceability system using blockchain technology.” 2022 8th International Conference on Signal Processing and Communication (ICSC). IEEE, 2022.

4. Rahimi, Iman, Amir H. Gandomi, M. Ali ¨ Ulk¨ u, and Simon James Fong. ”1 Big Data Analytics in Supply Chain Management.” Big Data Analytics in Supply Chain Management: Theory and Applications (2020): 1.

5. Giannoccaro, Ilaria, and Pierpaolo Pontrandolfo. ”Inventory management in supply chains: a reinforcement learning approach.” International Journal of Production Economics 78.2 (2002): 153-161.

6. Kumaar, AA Nippun, and Sreeja Kochuvila. ”Reinforcement learning based path planning using a topological map for mobile service robot.” 2023 IEEE Interna tional Conference on Electronics, Computing and Communication Technologies (CONECCT). IEEE, 2023.

7. Jiang, Chengzhi, and Zhaohan Sheng. ”Case-based reinforcement learning for dy namic inventory control in a multi-agent supply-chain system.” Expert Systems with Applications 36.3 (2009): 6520-6526.

8. Ravulapati, Kiran Kumar, Jaideep Rao, and Tapas K. Das. ”A reinforcement learn ing approach to stochastic business games.” Iie Transactions 36.4 (2004): 373-385.

9. Chaharsooghi, S. Kamal, Jafar Heydari, and S. Hessameddin Zegordi. ”A reinforce ment learning model for supply chain ordering management: An application to the beer game.” Decision Support Systems 45.4 (2008): 949-959.

10. Shar, Ibrahim El et al. “Deep Reinforcement Learning toward Robust Multi echelon Supply Chain Inventory Optimization.” 2022 IEEE 18th International Con ference on Automation Science and Engineering (CASE) (2022): 1385-1391. 11. Modares, Azam et al. “A vendor-managed inventory model based on optimal retail ers selection and reliability of supply chain.” Journal of Industrial and Management Optimization (2022): n. pag.

12. Stranieri, Francesco, and Fabio Stella. ”A deep reinforcement learning approach to supply chain inventory management.” arXiv preprint arXiv:2204.09603 (2022)

13. Raghuram, P., Bhupesh, S., Manivannan, R., Anand, P. S. P., & Sreedharan, V. R. (2022). Modeling and analyzing the inventory level for demand uncertainty in the VUCA world: evidence from biomedical manufacturer. IEEE Transactions on Engineering Management, 70(8), 2944-2954.

1. \* Corresponding author.   
    *E-mail addresses:* [nippunkumaar@blr.amrita.edu](mailto:nippunkumaar@blr.amrita.edu)( Nippun kumaar A.A), [pranav.03.h@gmail.com](mailto:pranav.03.h@gmail.com)( Pranav H). [harshini.bannu2004@gmail.com](mailto:harshini.bannu2004@gmail.com) (Sri Harshini Muppavarapu), [tarunrachuri0303@gmail.com](mailto:tarunrachuri0303@gmail.com) (Tarun Rachuri) [↑](#footnote-ref-1)