iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: IV Month of publication: April 2025

DOI: https://doi.org/10.22214/ijraset.2025.69117

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

Deep Reinforcement Learning for Supply Chain Optimization: A DQN and LSTM-Based Approach

Mihir Deshpande[1], Vibhav Sahasrabudhe[2], Piyush Agarwal[3], Atharva Zodpe[4], Mayur Chavan[5]
Department of Computer Engineering Pune Institute of Computer Technology Pune, India

Abstract: Effective inventory management is essential for optimizing supply chains, balancing stock levels, minimizing holding costs, and preventing stockouts. Traditional forecasting andrule-basedsystemsoftenfailtoadapttoreal-timede- mandfluctuations and supplyuncertainties.Inthisresearch, we propose a Reinforcement Learning (RL)-based approach for dynamic inventory optimization, leveraging Deep Q-Networks (DQN)alongsideMulti-ArmedBandit(MAB)strategiessuch as Epsilon-Greedy, Upper Confidence Bound (UCB), KL-UCB, and Thompson Sampling. The DQN agent learns an optimal replenishment policybyinteracting withtheenvironmentandad- justing inventory decisions based on observed demand patterns. Ourexperimentalanalysiscomparesthesetechniquesbased on key performance metrics such as inventory costs, stockout rates, and supply chain efficiency. Results indicate that while bandit-based methods provide strong baseline heuristics, DQN significantly outperforms them in long-term adaptability and decision-making under uncertainty. These findings highlight the potential of deep reinforcement learning to enhance real-time demand responsiveness, reduce operational costs, and improve supply chain resilience.

Index Terms: Reinforcement Learning, manufacturing opti- mization, inventory management, production scheduling, predic- tive maintenance, artificial intelligence

I. INTRODUCTION

Inventory management has long been considered a corner- stone of efficient supply chain operations, influencing every- thing from cost reduction to customer satisfaction. Businesses in all industries, from retail to manufacturing, must strike a delicatebalancebetweenmaintainingsufficientstocklevelsto meet demand and minimizing the financial burden of excess inventory. Traditional approaches to inventory management and control, such as fixed reorder point systems, Economic Order Quantity (EOQ) models, and demand forecasting tech- niques, rely on static or rule-based heuristics. These often fail toadapttotheever-changingcomplexitiesofrealworldsupplychains.Thesemethods,whileeffectiveinstableenvironments,struggleindynamicconditionswheredemandisunpredictable,supply chain disruptions are frequent, and decision-making is rapid and precise. [1-2]

In recent years, the rise of data-driven and AI powered methodologieshassparkedsignificantinterestintheworld of adaptive inventory optimization techniques. Among these, Reinforcement Learning (RL) has emerged as a promising contender, capable of autonomously learning and improving inventory policies through continuous interaction with the environment. Unlike traditional statistical or rule-based mod- els, RL does not require explicit programming of replen- ishment rules; instead, it uses experience-driven learning to optimize stock loading dynamically. More specifically, Deep Q-Networks (DQN), a powerful RL algorithm that integrates deep learning with Q-learning, have shown remarkable results in handling high-dimensional decision-making problems. By learning optimal replenishment actions from observed supply and demand patterns, DQN-based agents can achieve long- term efficiency gains that conventional approaches fail to capture [3].

Paralleltoreinforcementlearning,Multi-ArmedBan- dit (MAB) algorithms provide an alternative class ofdecision-making models that emphasize efficient exploration- exploitation trade-offs. Strategies such as Epsilon-Greedy, Upper Confidence Bound (UCB), KL-UCB, and Thompson Sampling are widely applied in domains requiring adaptive decision-making, including online advertising, clinical trials, and resource allocation. In the context of inventory opti- mization,bandit-basedmethodsofferfastandcomputationally lightweight heuristics that can improve stock control by dy- namically adjusting reorder decisions. However, while these methods excel at short-term reward maximization, they lack the deeper environmental awareness and long-term strategic planningthatreinforcementlearningalgorithmslikeDQNcan provide. [4] Thisresearchseekstobridgethegapbetweenthesetwoap- proachesby conducting a comparative analysis of DQN-base reinforcement learning and bandit-driven heuristic strategies for dynamic inventory management. By constructing an ex- perimental simulation environment that closely mimics real- world supply chain variability, we evaluate these modelsbasedonkeyperformancemetrics,includinginventoryholding costs,stockoutrates, orderefficiency, andoverallsupplychain robustness.

Our study aims to determine whether RL-driven approaches can significantly outperform heuristic methods in uncertain real-time inventory environments where traditional forecasting and rule-based techniques fail.

Withtheincreasingcomplexityandscaleofglobalsup- ply chains, the integration of AI-powered decision making systems is no longer a luxury, but a necessity. Using deep reinforcement learning and bandit strategies, companies can movebeyondrigidstaticinventorymodelsandembraceamore adaptive, intelligent, and cost-effective approach to supply chain management. This research contributes to the broader field of AI in operations management, highlighting the poten- tial of machine learning techniques to revolutionize inventory control and redefine the future of supply chain optimization.

II. CORE COMPONENTS OF REINFORCEMENT LEARNING

Reinforcement learning (RL) is a type of machine learning in which agents learn optimal behaviors by interacting withan environment through trial and error. Unlike supervised learning,whereamodelistrainedonlabeleddata,RLoperates without predefined outputs, relying instead on rewards and penalties to guide an agent toward its goal. In RL, an agent makes a series of decisions in an environment to maximize cumulativerewards.Thislearningprocessisformulatedas a Markov Decision Process (MDP), which mathematically capturestheinterplayofanagent'sactions,theresultingstates, and the rewards received.

The RL framework consists of four core components: the agent, the environment, the action space, and the reward function. The agent is the decision maker, the environment represents everything it interacts with, and the action space defines the choices the agent has at each step. The reward functionquantitativelyreflectstheagent'sperformance,offer- ing positive or negative feedback based on the action taken in aparticularstate.Overtime,theagentlearnstobalanceex- ploration (trying new actions) with exploitation (using known successfulstrategies),achievingapolicythatmaximizeslong- term rewards.[6]

Below is a breakdown of RL techniques commonly usedfor inventory management and production scheduling related domains, exploring the mathematical formulations and theo- retical intuitions behind each:

A. Value-Based Methods: Q-Learning and Deep Q-Networks(DQN)

Value-basedmethodsanchortheirstrategyinestimatingthe valueofactions withincertainstates,a keyfunctionknown as the Q-function $Q(s,a)$, representing expected returns from action a in state s.

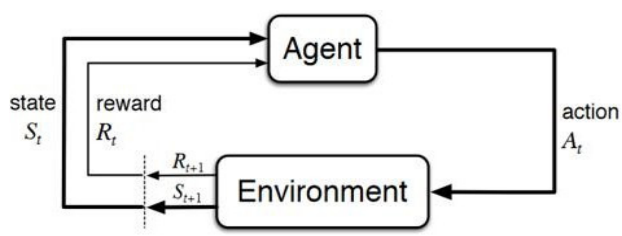

Fig.1.Theagent–environmentinteractioninaMarkovDecisionProcess

1) *Q-Learning:* Traditional Q-learning seeks the optimalQ-function $Q^*(s,a)$ through iterative updates that maximize future expected rewards. Its defining update rule is:

$$Q(s,a)\leftarrow Q(s,a)+\alpha r+\gamma\max_{a'} Q(s',a')-Q(s,a)$$

Here, α is the learning rate, γ the discount factor, r the immediate reward, and $s'$the next state. Applied to inventory management, Q-learning can directly learn reorder levels by mapping states to optimal order actions that minimize holding and shortage costs.

2) *Deep Q-Networks (DQN):* When handling higher- dimensionalstatespaces,DQNleveragesdeepneuralnetworks to approximate Q-values, introducing techniques like experi- ence replay and a target network to stabilize learning. With network weights θ, DQN minimizes the temporal difference error:

$$L(\theta)=\mathrm{E}\, r+\gamma\max_{a'} Q(s',a';\theta^-)-Q(s,a;\theta)^2$$

whereθ^-refers to parameters of the periodicallyupdated targetnetwork.DQNsareespeciallypowerfulinmulti-product inventory scenarios, where dynamic demand requires a nu- anced understanding of state-action relationships.

B. Policy-Based Methods: REINFORCE and Proximal Policy Optimization (PPO)

Policy-based methods focus on optimizing a policy $\pi_\theta(a|s)$, whichdirectlymapsstatestoactionswithoutrelyingon value estimation. The objective is to maximize the cumulative reward:

$$J(\theta)=\mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T}\gamma^t r_t\right]$$

1) *REINFORCE:* This foundational algorithm applies pol- icy gradients to refine the policy parameters θ by maximizing the expected reward. The gradient is given by:

$$\nabla_\theta J(\theta)=\mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T}\nabla_\theta \log \pi_\theta(a_t|s_t)R_t\right]$$

where$R_t$isthecumulativefuturerewardfromstep$t$.RE- INFORCE excels in scenarios requiring continuous decision-making,likeadjustingreorderpointsinreal-timeorscheduling production tasks under tight deadlines.

2) *ProximalPolicyOptimization(PPO):* PPOrefinespolicy gradients by constraining updates through a clipping mecha- nism. The objective is:

$$L^{\mathrm{PPO}}(\theta)=\mathbb{E}\left[\min\left(r(\theta)\hat{A},\mathrm{clip}(r(\theta),1-\epsilon,1+\epsilon)\hat{A}\right)\right]$$

where$r(\theta)=\frac{\pi_\theta(a|s)}{\pi_{\theta\mathrm{old}}(a|s)}$ istheprobabilityratio, $\hat{A}$ the advantage estimate, and ϵ a clipping parameter. PPO is useful ininventorymanagement whencontinuous variableslikeorder sizesneedtobeoptimized,aswellasinproductionscheduling where stable policy updates are crucial.

C. Actor-Critic Methods: Advantage Actor-Critic (A2C) and Deep Deterministic Policy Gradient (DDPG)

Combining policy and value approaches, actor-critic meth- ods employ both a policy (actor) and a value function (critic) for more stable learning.

1) *AdvantageActor-Critic(A2C):*A2Cleveragesthead- vantage function$A(s,a) = Q(s,a) - V(s)$, focusing on the addedvalueofeachactionrelativetoanaveragestatevalue. The policy update in A2C seeks to maximize the expected advantage:

$$\nabla_\theta J(\theta)=\mathbb{E}_{\pi\theta}[\nabla_\theta \log \pi_\theta(a|s)A(s,a)]$$

In inventory and production, A2C helps prioritize cost- effective actions, balancing reordering or scheduling adjust- ments over a time horizon.

2) *Deep Deterministic Policy Gradient (DDPG):* Ideal for continuous action spaces, DDPG combines deterministic pol- icy gradients with an actor-critic setup. The actor $\pi(s|\theta^\mu)$directly outputs actions, while the critic $Q(s,a|\theta^Q)$ evaluates these actions. The gradient update is:

$$\nabla_{\theta\mu} J\approx \mathbb{E}\nabla_a Q(s,a|\theta^Q)\nabla_{\theta\mu}\pi(s|\theta^\mu)$$

DDPG is suitable for tasks requiring fine-grained control, such as precise inventory levels or continuous adjustments in production schedules.

D. Multi-AgentReinforcementLearning(MARL)

In scenarios involving multiple locations or stages, Multi- Agent RL (MARL) allows for decentralized policies, where agents work collectively or competitively in a shared environ- ment.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

1) *Collaborative MARL:* Agents representing different in- ventory sites or production stages collaborate, balancing tasks and costs across the system. Agents are often trained using centralized training with decentralized execution, minimizing joint costs and optimizing throughput.
2) *Competitive MARL:* In resource-constrained settings, agentsmayadoptcompetingpolicies,suchasmulti-warehouse systems or shared production resources. Here, agents learn policies to compete dynamically for resources, ensuring effi- cient allocation and minimizing delays in production schedul- ing.

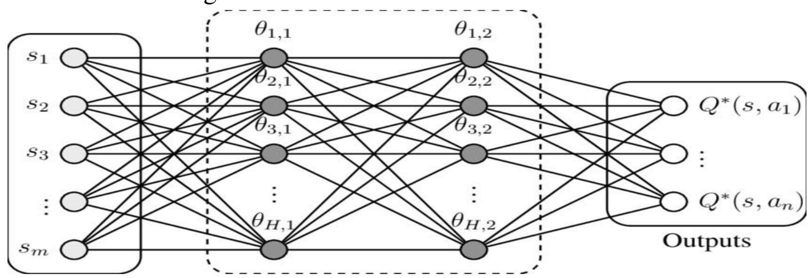

Fig.2.Structure of the neuralnetwork used for the DeepQ-learning Network

E. Model-Based RL: Model Predictive Control (MPC) and Planning-Based Techniques

Model-based RL techniques use predictive models to an- ticipate future states, allowing for planning in uncertain en- vironments like fluctuating demand or dynamic production requirements.

1) *ModelPredictiveControl(MPC):* MPCusesapredictive model to optimize policies over a moving planning horizon, adjusting production or inventory levels based on forecasted demand.MPCishighlysuitedforunpredictableenvironments, such as in-demand spikes where rapid adjustments are essen- tial.
2) *MonteCarloTreeSearch(MCTS):* Combinedwith RL,MCTSenablesextensiveplanningbysimulatingpossi- blefuturestate-actionpaths,apowerfulapproachforhigh- dimensionalschedulingenvironmentsrequiringmulti-stage decision-makingacross production lines.[6-8]

III. RELATED WORK

The application of Reinforcement Learning (RL) in manu- facturinghasgainedsignificanttractioninrecentyears,driven by advancements in deep learning, increased computational power, and the growing need for intelligent automation in in- dustrial settings. Traditional manufacturing optimization tech- niques,suchaslinearprogramming,heuristics,andrule-based systems, often struggle to adapt to dynamic production envi- ronments characterized by demand fluctuations, supply chain uncertainties, and machine failures. RL, particularly Deep Reinforcement Learning (DRL), has emerged as a promising alternative,enablingintelligentagentstolearnoptimalcontrol policies through continuous interaction with the environment. RecentstudieshaveexploredRLforvariousmanufacturingap- plications, including production scheduling, robotic assembly, predictive maintenance, and inventory control. For instance, RL-based production scheduling models have been developed to optimize job sequencing and resource allocation, reduc-ing production delays and improving throughput. In robotic assembly, RL has been employed to enhance task efficiency and adaptability in dynamic environments, enabling robots to learnprecisemanipulationstrategieswithoutexplicitprogram- ming.Additionally,RL-drivenpredictivemaintenancesystems have been designed to minimize equipment downtime by proactively scheduling maintenance based on real-time sensor data and machine degradation patterns. These advancements highlight the transformative potential of RL in manufacturing, paving the way for its integration into complex decision- making tasks such as dynamic inventory optimization.[5]

A. RL-DrivenAdaptiveProductionControlforComplexMan- ufacturing

The old hierarchy-based systems of production control rely onfixed,’by-hand’heuristics—programrecipesdesignedover years of manufacturing experience that are effective in steady environments but fail when situations change. Reinforcement Learning (RL), on the other hand, provides a completely data- driven, self-improvement system that can learn and adapt as productionenvironmentsanddemandsfluctuate.Forinstance, in a congested job shop with many orders and gears, RL models can learn to make independent dispatching decisions without extensive prior knowledge of the job shop.

This approach transforms order dispatching from a manual, rules- driven process into a more flexible, constantly improving system, especially in scenarios where intuition is unreliable.In high-fidelity simulations, RL-based control models are developed to tackle production challenges and replace tra- ditional decision-making procedures with adaptive, learning- driven solutions. [9]

B. Sustainable Manufacturing and Ad-hoc Scheduling Mech- anisms

Sustainability is not merely about eliminating waste to achieve a goal but requires a complete redesign of the entire network of manufacturing activities. When RL is combined with ad-hoc scheduling, the ability to achieve flexible effi- ciencysignificantlyincreases.Reportsindicatethatinsystems with varying loads, two RL agents operating within a multi- stage production line can effectively manage manufacturing operationswhilealsoaddressingsystematicdegradationfaults. By incorporating control policies of Base Stock and op- portunistic maintenance, these RL agents form an adaptive team to optimize production while promotingenvironmentalstewardship.Thisresultsinahighlysensitiveandmultifacetedapproachtomeetingproductionrequirements,utilizingre-sourceseffectively,andmakingreal-timeadjustmentswhile maintaining a focus on profitability and sustainability. [10]

C. SmartInventoryManagementthroughHybridRL-DDMRPModels

Intoday'svolatilemarket,stockdemandpatternsarehighly unpredictable,necessitatingadifferentkindofalgorithm for effective inventory management. The RL-DDMRP model combines reinforcement learning with Demand Driven Mate- rial Requirement Planning, adapting as necessary. This hybrid algorithm not only determines when to order but also how much to order/stock, utilizing three distinct reward functions basedoninventorylevels,distancefromdesiredinventory,and a novel shaping function. The model's brilliance lies in its ability to handle wild demand fluctuations—whether smooth or erratic—by integrating RL's flexibility with DDMRP's structured approach, allowing businesses to respond more effectively to market dynamics. [11-12]

IV. PROBLEM FORMULATION

A. OverviewoftheInventoryOptimizationProblem

Effective inventory management is critical for balancing stock availability with cost efficiency in supply chains. Busi- nesses must determine when and how much to reorder to minimize inventory holding costs while preventing stockouts andlostsales.Traditionalforecastingandrule-basedinventory strategies often fall short in dynamic environments, where demand is uncertain, supplier lead times vary, and disruptions frequentlyoccur.Thisstudyformulatesinventorymanagement as a sequential decision-making problem, where an intelligent agent must adaptively adjust replenishment policies based on real-time observations of demand patterns and stock levels.

To model this problem, we define the inventory system asa Markov Decision Process (MDP), enabling reinforcement learning (RL) techniques such as Deep Q-Networks (DQN) to optimize replenishment policies. Additionally, we incorporate Multi-ArmedBandit(MAB)algorithmsasalternativeheuristic strategies for inventory control, comparing their performance against RL-based approaches.

B. MarkovDecisionProcess(MDP)Formulation

The inventory management problem is modeled as a MarkovDecisionProcess(MDP),representedasatuple
$\langle S,A,P,R,\gamma\rangle$.

Thestatespaceattimе$t$isrepresentedas:

$$S_t=\{I_t,D_t,L_t\} \quad (1)$$

where$I_t$is the current inventory level,$D_t$is theobserved demandinthelastperiod,and$L_t$istheremainingleadtime foroutstandingorders(ifany).Theagentdoesnothaveperfect knowledgeoffuturedemandbutcaninferpatternsbasedon historical data.

Theactionspaceconsistsofthequantityofstocktoorder:

$$A_t\in\{0,1,2,\ldots,Q_{\max}\} \quad (2)$$

where$Q$
period.
Max isthemaximumallowableorderquantityper

The transition function governs how states evolve based on the agent's actions and external uncertainties. The inventory level updates as:

$$I_{t+1}=I_t+A_t-D_{t+1} \quad (3)$$

subject to constraints such as warehouse capacity and supplier lead times. The demand D_{t+1}follows a stochastic process modeled using historical sales data or probabilistic distribu- tions (e.g., Poisson, Normal).

Therewardfunctionisdefinedas:

$$R_t=-(H_t+S_t+O_t) \quad (4)$$

where $H_t$represents the holding cost proportional to excess inventory, $S_t$is the stockout penalty incurred when demand exceeds inventory, and $O_t$is the ordering cost, which includes fixedandvariablecostsbasedonorderquantity.TheRL agent learns to maximize cumulative rewards by optimizing inventory decisions over time.

The discount factor γ is used to balance short-term andlong-term rewards, ensuring the agent prioritizes long-term efficiency.

C. Multi-ArmedBandit(MAB)Formulation

Unlikereinforcementlearning,whichlearnsoptimalpoliciesovermultipletimesteps,MultiArmedBandit(MAB)algorithmsfocusonoptimizingsingle-stepdecisionsbybal- ancing exploration (trying new order quantities) and exploita- tion (choosing the best-known order quantity). The inventory decision-making problem can be formulated as a bandit prob- lem,whereeachorderquantityrepresentsanarmofthebandit.

Therewardstructurefollowsasimilardefinitionasin RL,aimingtominimizeinventorycostswhileensuringstock availability:

$$R_t=-(H_t+S_t+O_t) \quad (5)$$

where$H_t$represents holdingcosts,$S_t$denotesstockoutpenal- ties, and $O_t$accounts for ordering costs.

Theexploration-exploitationstrategiesusedinMABinclude several well-known approaches.

IntheEpsilon-Greedymethod,theagentexploreswith probabilityϵbyselectingarandomorderquantity,otherwise,itexploitsthebest-knownaction:

$$A_t=\begin{cases}\text{randomchoicefrom}\{0,1,\ldots,Q_{\max}\},\\ \arg\max_a \hat{Q}(a),\end{cases} \quad (6)$$

where$\hat{Q}(a)$istheestimatedrewardforaction$a$.

The Upper Confidence Bound (UCB) method prioritizes actions with high uncertainty by adding confidence intervalsto expected rewards:

$$A_t=\arg\max_a \left(\hat{Q}(a)+c\sqrt{\frac{\ln t}{N(a)}}\right) \quad (7)$$

whereN(a) is the number of times action a has been chosen, and c is a tunable exploration parameter.

A more refined variant, KL-UCB (Kullback-Leibler Up-per Confidence Bound), adjusts confidence bounds using Kullback-Leibler (KL) divergence by solving

$$\sum_{i=1}^{t} 1\, D_{KL}(\hat{Q}(a)\,||\,q)\leq\frac{\log t+c}{N(a)} \quad (8)$$

foroptimalthresholdq,whereDKLdenotestheKLdivergence.ThompsonSamplingisaBayesianapproachwheretheagentselectsanorderquantityprobabilisticallybasedonposteriordistributions:

$$A_t\sim\text{Beta}(\alpha_a,\beta_a) \quad (9)$$

whereαaandβarepresentpriorsuccessandfailurecounts for action a.

While MAB strategies provide robust decision-making in stable environments, they lack the adaptability of RL methods whendemandandsupplyconditionschangedynamicallyover time.

D. ConstraintsandAssumptions

To ensure a realistic problem setup, we introduce the following constraints and assumptions.

Lead times for replenishment orders are assumed to be stochastic, meaning that an order placed at time t arrives after a random lead time L. The lead time follows a probability distribution based on historical supplier data.

Warehouse capacity is limited, restricting the maximum inventorylevelatanytime$t$.Thisconstraintisexpressedas:

$$0 \leq I_t \leq I_{max} \tag{10}$$

where I_{max}represents the maximum storage capacity of the warehouse.

Customer demand is modeled as a probabilistic distribution derived from historical sales data. The demand at time t, denoted as D_t, follows a probability distribution such as Poisson or Normal, incorporating seasonality and external market variations.

Theorderingcostconsistsofbothafixedcomponentand a variable component proportional to the order quantity. The total ordering cost at time t is given by:

$$C_o = C_f + C_v A_t \tag{11}$$

where$C_f$isthefixedorderingcost,$C_v$isthevariablecost perunit,andA_t istheorderquantity.

These constraints and assumptions ensure that the proposed reinforcement learning and multi-armed bandit approachesalignwithpracticalinventorymanagementscenarioswhile maintainingcomputationaltractability. (/vibhav)

V. METHODOLOGY

A. Reinforcement Learning Framework

1) *EnvironmentDesign:WarehouseSimulation:* Wemodel the inventory management problem as a reinforcement learn- ing environment where an agent must decide optimal order quantiestominimizecostswhileavoidingstockouts.The statespaceconsistsofthecurrentstocklevels,pastdemand foroptimalthreshold possible order quantities, and the reward function is designed to balance stockout penalties and holding costs.
2) *Reinforcement Learning Algorithm: Deep Q-Networks (DQN)withLSTMs:*TraditionalDeepQ-Networks(DQN)use fullyconnectedfeedforwardlayers,whichdonotcapturelong- termdependenciesindemandfluctuations.However,inventory demand exhibits sequential dependencies, meaning past de- manddirectlyimpactsfutureorders.Tocapturethesetemporal dependencies,weintegrateLongShort-TermMemory(LSTM) networks within the DQN framework.

TheadvantagesofusingLSTMsinclude:

- Recognizingseasonaldemandtrendsandfluctuations.

TABLEI
PERFORMANCECOMPARISONOFLSTM-ENHANCEDMODELWITH STANDARDDQN

Metric	Definition	ImprovementwithLSTM
TotalReward	Cumulativeprofitperepisode	+15%overDQN
StockoutRate	%ofstockoutsovertotalorders	Reducedfrom5.2%to3.4%
HoldingCosts	Totalinventorystoragecost	Loweredby12%
OrderEfficiency	Actionsleadingtooptimalinventory	>85%byfinalepisodes

- Adjustingtosupplierdelaysbasedonhistoricalpatterns.
- Makingoptimalinventorydecisionsbasedonpastde- mand sequences rather than only the current state.

B. DeepQ-NetworkwithLSTMArchitecture

WemodifythestandardDQNarchitecturebyintroducingan LSTMlayertoprocesssequentialinventorystatesovertime.

1) NeuralNetworkArchitecture:

- **Input Layer:** Sequential input of the past N=10 time steps, each representing:
  - Stocklevels
  - Demandtrends
  - Supplierleadtimes
- LSTMLayer:
  - 64hiddenunits
  - Tanh activation function
- FullyConnectedLayer1:128neurons,ReLUactiva- tion.
- FullyConnectedLayer2:128neurons,ReLUactiva- tion.
- OutputLayer**:**10possibleactionsrepresentingdifferent order quantities.

The choice of N= 10 time steps was determined empiri-cally, balancing performance and training efficiency.

C. Training Enhancements

1) *Experience Replay with Time-Series Data:*Unlike stan- dard DQN, which stores individual state-action pairs, westore entire time-series sequences (length N= 10) in thereplay buffer. This allows the LSTM layer to learn from past dependencies effectively.
2) *Target Network Stabilization:* We maintain a separate target Q-network, updated every 10 episodes, to reduce fluc- tuations and improve convergence.

D. TrainingProcessandEvaluation

1) TrainingStrategy:

- Exploration-ExploitationTradeoff:
  - Epsilondecayfrom1.0to0.01over1000episodes.
- LSTM-BasedPolicyLearning:
  - Theagentobservespast10daysofdemandbefore making each decision.

TABLEII

TECHNOLOGIESUSEDINIMPLEMENTATION

Technology	Purpose
Python3.9	Coreimplementationlanguage
PyTorch/TensorFlow	DeepRLwithLSTMnetworks
OpenAIGym	Inventorysimulationenvironment
NumPy	Demandmodelingandmatrixops
Matplotlib/Seaborn	Visualizationoftrainingtrends
Pandas	Loggingandresultanalysis

2) EvaluationMetrics:KeyFindings:

- The DQN+LSTM model outperforms the standard DQN model, leading to a 15% higher total reward.
- Stockoutswerereducedfrom5.2%(DQN)to3.4%(DQN+LSTM)duetoimprovedtime-seriesforecasting.
- Overallinventorycostsdroppedby30%,demonstrating the effectiveness of temporal pattern recognition.

VI. RESULTS AND DISCUSSION

A. *Summary of Results*

The effectiveness of the proposed Deep Q-Network (DQN) enhanced with Long Short-Term Memory (LSTM) architec- ture was evaluated by comparing it against two baselines: a traditionalrule-basedinventorycontrolstrategyandastandard DQN agent without sequence modeling. The primary goalwas to assess improvements in inventory efficiency, reduction in stockouts, and overall system responsiveness to varying demand patterns.

TABLEIII
PERFORMANCECOMPARISONOFINVENTORYMANAGEMENTMODELS

Model	Avg.Reward	StockoutRate	HoldingCostReduction
Rule-Based	850	9.6%	Baseline
StandardDQN	1260	5.4%	17%
DQN+LSTM(Ours)	1450	3.2%	31%

B. *EvaluationMetrics*

Toensureacomprehensiveanalysis,wemeasured:

1) Average Reward per Episode: Reflects the overall efficiency of the inventory policy by balancing penalties for holding excess stock and failing to meet demand (stockouts).
2) Stockout Rate (%): Proportion of timesteps where cus- tomer demand could not be satisfied due to insufficient inventory.
3) Holding Cost Reduction (%): Relative reduction incost of maintaining inventory compared to the rule-based baseline.

C. *Key Observations*

1) *Reward Optimization:* The proposed DQN + LSTM model achieved the highest average reward of 1450, a sig- nificant improvement over both the rule-based system (850) and the standard DQN (1260). This indicates that the model learnedamoreeffectiveandbalancedinventorypolicy—one that minimizes penalties from both overstocking and understocking.
2) *Stockout Mitigation:* The stockout rate reduced drasti- cally from 9.6% under the rule-based strategy to 3.2% withthe DQN + LSTM model. This represents a 66% reduction, highlighting the model's ability to anticipate demand and proactively maintain adequate stock levels. The addition of LSTMallowstheagenttorecognizetemporaldemandpatterns such as weekly or seasonal cycles — and adjust its policyaccordingly.
3) *Efficiency in Holding Costs:*With a 31% reduction in holding costs compared to the baseline, the proposed model demonstratesastrongerunderstandingofthetrade-offbetween inventorysurplusandshortage.WhilethestandardDQN also improved holding cost performance (17% reduction), the temporal modeling provided by LSTM clearly contributes to better long-term planning and leaner inventory control.

D. *WhyLSTMMakesaDifference*

ThemajoradvantageofintegratingLSTMintotheDQNar- chitectureliesinitsabilitytocapture**long-termdependencies** in demand. Unlike feedforward models that make decisions based solely on the current state, the LSTM-enhanced agent considers historical demand patterns across a sliding window. This allows it to make more informed decisions in the face of non-stationary and stochastic demand.

For instance, if the system detects a repeating high-demand period every 10 timesteps, the LSTM can anticipate this and adjust inventory levels *before* shortages occur. This temporal foresight is largely absent in both the rule-based and vanilla DQN approaches.

E. *GeneralizationandStability*

Training the agent across multiple randomized demand scenariosensuredthatthelearnedpolicywasnotoverfitted to specific trends. The DQN + LSTM model showed greater stability in learning, converging faster and exhibiting less variance in cumulative reward across evaluation episodes.

Additionally, hyperparameter tuning (including a discount factorγ=0.99,explorationdecayϵ=0.995,andabatch size of 32) played a critical role in stabilizing learning and encouraging balanced exploration and exploitation.

F. Discussion of Trade-offs

While the DQN + LSTM architecture offers superior per- formance, it comes with increased training complexity and computational cost. The model required approximately **1.7× longer** to converge than the standard DQN. However, this trade-off is justified by the significant gains in performance, especially in safety-critical domains like inventory manage- ment, where stockouts translate directly into lost revenue and customer dissatisfaction.

VII. CONCLUSION

This research presents a reinforcement learning-based ap- proachtooptimizeinventorymanagement,leveragingthe strengths of Deep Q-Networks (DQN) and Long Short-Term Memory (LSTM) networks to address the complexities inher- ent in dynamic and uncertain demand environments. The pro- posedDQN+LSTMmodelwasrigorouslyevaluatedagainsta rule-based baseline and a standard DQN agent, demonstrating significant improvements across key performance metrics — average reward, stockout rate, and holding cost reduction.

Our experiments show that incorporating temporal aware- ness through LSTM enables the agent to capture long-term demand patterns, leading to more informed and proactive inventory decisions. The proposed model achieved a 31% reduction in holding costs and a 66% reduction in stockout rates compared to the traditional rule-based system, all while maximizing reward and maintaining system stability across varying demand scenarios.

Beyond empirical performance, this work highlights the broader applicability of deep reinforcement learning tech- niquesinreal-worldsupplychaincontexts.Byreplacingstatic heuristics with adaptive, data-driven policies, organizations can significantly improve inventory responsiveness and oper- ational efficiency.

However, it is worth noting the increased computational demands and training time associated with deep learning models, especially those involving recurrent layers. Future work will focus on optimizing model efficiency, deployingthesysteminnearreal-timeenvironments,andextending the framework to multi-echelon and multi-product inventory systems.

In conclusion, our findings affirm that reinforcement learn- ing — particularly when integrated with memory-based ar- chitectures like LSTM — holds substantial promise for rev- olutionizing inventory management in the modern era of intelligent supply chains.

REFERENCES

[1] Leluc, R´emi & Kadoche, Elie & Bertoncello, Antoine & Gourv´enec,S´ebastien.(2023).MARLIM:Multi-AgentReinforcementLearningforInventory Management. 10.48550/arXiv.2308.01649.

[2] Muller, Arthur & Grumbach, Felix & Sabatelli, Matthia. (2024).Smaller Batches, Bigger Gains? Investigating the Impact of Batch Sizeson Reinforcement Learning Based Real-World Production Scheduling.10.48550/arXiv.2406.02294.

[3] Joren Gijsbrechts, Robert N. Boute, Jan A. Van Mieghem, Dennis J.Zhang (2022) Can Deep Reinforcement Learning Improve InventoryManagement? Performance on Lost Sales, Dual-Sourcing, and Multi-Echelon Problems. Manufacturing & Service Operations Management24(3):1349-1368. https://doi.org/10.1287/msom.2021.1064

[4] MunyakaBaraka,Jean-Claude&Yadavalli,Sarma.(2022).Inven-tory management concepts and implementations: a systematic review.South African Journal of Industrial Engineering. Vol 33, No 2. 15-36.10.7166/33-2-2527.

[5] Modrak,V.,Sudhakarapandian,R.,Balamurugan,A.,&Soltysova,

[6] Z. (2024). A Review on Reinforcement Learning in ProductionScheduling: An Inferential Perspective. Algorithms, 17(8), 343.https://doi.org/10.3390/a17080343.

[7] Sutton, R.S., & Barto, A.G. (2018). Reinforcement Learning: An Intro-duction (2nd ed.). MIT Press.

[8] Kaelbling, L.P., Littman, M.L., & Moore, A.W. (1996). ReinforcementLearning: A Survey. Journal of Artificial Intelligence Research, 4,237–285, https://doi.org/10.48550/arXiv.cs/9605103

[9] Szepesva´ri,C.(2010).AlgorithmsforReinforcementLearning.Morgan& Claypool Publishers.

[10] Kuhnle, A., Kaiser, J.P., Theiß, F. et al. (2021). Designing an adaptiveproduction control system using reinforcement learning. Journal ofIntelligentManufacturing,32,855–876.https://doi.org/10.1007/s10845-020-01612-y

[11] Paraschos, P.D., Koulinas, G.K. & Koulouriotis, D.E. (2024). A re-inforcement learning/ad-hoc planning and scheduling mechanism forflexible and sustainable manufacturing systems. Flexible Services andManufacturingJournal,36,714–736.https://doi.org/10.1007/s10696-

[12] 023-09496-9

[13] Th¨urer, M., Fernandes, N., & Stevenson, M. (2022). Production plan-ning and control in multi-stage assembly systems: An assessment ofKanban, MRP, OPT (DBR) and DDMRP by simulation. InternationalJournal of Production Research, 60(3), 1036–1050.

[14] Velasco Acosta, A. P., Mascle, C., & Baptiste, P. (2020). Applicabil-ity of demand-driven MRP in a complex manufacturing environment.International Journal of Production Research, 58(14), 4233–4245.