

# Dynamic Policy Optimization for E-commerce Returns: A Reinforcement Learning Approach for SMEs with Limited Data

*Vijay M*

Department of Computer Science and Applications, Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya (SCSVMV), Kanchipuram, Tamil Nadu 631561, India. [vijaymani9989@gmail.com](mailto:vijaymani9989@gmail.com)

**Abstract.** E-commerce returns represent a persistent challenge for small and medium-sized enterprises, with return rates averaging 17.6% across retail categories. I present a reinforcement learning framework that dynamically optimizes return policies for SMEs operating with limited transaction histories (10,000-100,000 records). My approach combines LASSO regression, Gradient Boosting, and customer segmentation with Q-learning to enable real-time policy adjustments. The framework incorporates pre-purchase interventions including augmented reality try-on features and AI-driven size recommendations. Testing on 100,000 German e-commerce transactions alongside deployment in an Indian marketplace showed 32% reduction in returns (from 24.7% to 16.8%), with prediction accuracy reaching 90.8%. The system achieved ROI between 109-736% while maintaining sub-100ms response times on standard cloud infrastructure. Through SHAP-based explainability, I demonstrate how SMEs can adopt sophisticated AI tools despite data constraints.

## 1 Introduction

### 1.1 The Challenge of E-commerce Returns

The rapid expansion of online retail has brought with it an unexpected operational burden: product returns. While convenient for customers, returns create substantial costs for retailers. Current research shows average return rates of 17.6% across e-commerce, with certain categories like fashion reaching 30% or higher[1][2]. These returns generate costs well beyond the lost sale—there's reverse logistics, processing labor, depreciation of returned items, and in many cases, disposal costs when products can't be resold[3].

For small and medium-sized enterprises, the returns problem becomes particularly acute. Unlike major platforms with millions of transactions to analyze, SMEs typically handle between 10,000 and 100,000 orders annually. This data scarcity makes conventional machine learning approaches less reliable. Additionally, SMEs rarely have dedicated data science teams or substantial computing budgets. They need systems that run efficiently on standard cloud services and provide interpretable results that business owners can actually understand and trust.

My review of recent literature (2020-2025) revealed an interesting gap. While plenty of research addresses return prediction for large platforms, very little focuses on the specific constraints facing smaller businesses. Most published studies work with datasets exceeding 250,000 transactions[4][5], and they emphasize prediction accuracy rather than providing actionable policy recommendations[1]. Perhaps most notably, I found no prior work applying reinforcement learning to optimize return policies specifically for data-constrained SME environments. There's also limited research on explainable AI approaches that could help smaller businesses understand and trust automated decisions[2].

### 1.2 My Contributions

This paper addresses these gaps through several interconnected contributions. First, I've developed an RL-powered policy engine using Q-learning adapted for few-shot learning scenarios. This enables SMEs with just 10,000-100,000 transactions to implement dynamic, real-time policy adjustments based on customer behavior patterns.

Second, I present a hybrid forecasting architecture that combines three techniques: LASSO regression handles feature selection,

Gradient Boosting captures complex nonlinear patterns, and threshold-based segmentation groups customers by risk level. This ensemble approach achieved 90.8% accuracy even with limited training data, thanks to carefully engineered features and automated hyperparameter optimization.

Third, I introduce a framework for pre-purchase interventions that serves as the action space for my RL agent. These interventions range from AR virtual try-on experiences to AI-powered size recommendations and proactive customer support. In targeted product categories, these interventions reduced returns by 28-34%.

Fourth, I validated the approach cross-culturally through real deployment in an Indian bathroom accessories marketplace. This demonstrated that models trained on German e-commerce data could generalize to emerging market contexts, though with interesting cultural adaptations.

Finally, I integrated explainable AI using SHAP values throughout the system. This transparency helps SME stakeholders understand why the system makes particular recommendations, building trust in automated decision-making.

## 2 Related Work and Research Context

Karl's recent systematic review (2025) along with work by Duong and colleagues (2022) highlighted three critical gaps in returns research[1][2]. Most studies concentrate on large platforms with extensive transaction histories, leaving smaller businesses without applicable solutions. The field has also emphasized prediction over action—knowing a return is likely doesn't help much if you don't know what to do about it. And there's been surprisingly little work on making these systems interpretable for business users.

The state-of-the-art in return prediction comes from Cui et al. (2020), who achieved 89.3% accuracy using Gradient Boosting on 250,000 transactions[4]. Their work identified key predictive factors: how long customers browse, whether they compare multiple products, and

their historical return patterns. More recently, Krishnaswamy et al. (2025) showed how explainable AI could drive process improvements, cutting undelivered returns from 22.5% to 6.34% using SHAP values to understand model decisions[2]. This work particularly resonated with me because it demonstrated the practical value of interpretability in SME contexts.

However, no previous research has combined reinforcement learning with explainable AI specifically for return policy optimization in data-constrained environments. Ambilkar et al. (2022) actually identified this as a critical research priority in their comprehensive review of returns management[3]. Table 1 shows how my work fits into the existing research landscape, being the first to simultaneously address SME data constraints, dynamic policy optimization through RL, and explainable AI.

Study	Data set	SME Focus	RL Policy	XAI
Cui et al. 2020[4]	250 K	No	No	No
Krishnaswamy 2025[2]	Real	Yes	No	Yes
<b>My Work</b>	<b>100 K</b>	<b>Yes</b>	<b>Yes</b>	<b>Yes</b>

Table 1: Positioning my research relative to existing work

## 3 Methodology

### 3.1 Overall System Design

My framework operates through four integrated stages. First, a hybrid ML ensemble

predicts the probability that any given customer will return their purchase. Second, I segment customers into high-risk and low-risk groups based on these predictions. Third, a Q-learning agent determines the optimal intervention for each transaction. Finally, the system deploys these interventions in real-time during the purchase process. The entire pipeline runs in under 100 milliseconds on standard AWS or GCP infrastructure, processing customer signals at checkout to trigger appropriate actions.

### 3.2 Data and Feature Engineering

I worked with a publicly available German e-commerce dataset containing 100,000 transactions from an 18-month period (January 2022 through June 2023). The dataset includes 47,382 unique customers purchasing 8,743 products across 12 categories, with an overall return rate of 24.7%. All customer identifiers were anonymized using SHA-256 hashing to ensure CCPA compliance.

From this raw data, I engineered 284 features spanning four categories. Behavioral features (87 total) capture actions like browsing duration, product comparisons, review reading time, and search patterns. Temporal features (63 total) encode timing information—time of day, day of week, purchase frequency, and seasonal effects. Product attributes (78 features) include category, price, ratings, and historical return rates by SKU. Customer profile features (56 total) cover return history, account age, loyalty status, location, and payment methods.

This many features would overwhelm a small dataset, so I used LASSO regression with L1 regularization to identify the 47 most predictive variables. This dimensionality reduction maintained accuracy while ensuring computational efficiency for SME deployment scenarios.

### 3.3 Predictive Model Development

I built an ensemble model combining three approaches with weights optimized through cross-validation:

$$P(\text{return}|x) = \alpha \cdot P_{\text{LASSO}}(x) + \beta \cdot P_{\text{GB}}(x) + \gamma \cdot P_{\text{seg}}(x)$$

where the weights sum to one. The LASSO component provides feature selection and an interpretable linear baseline ( $\lambda = 0.01$ , 1000 iterations maximum). Gradient Boosting via XGBoost captures nonlinear relationships (learning rate 0.05, max depth 6, 200 estimators, 80% subsampling with early stopping). The segmentation component adjusts predictions based on which risk segment a customer falls into.

I trained using a 70/15/15 split for training, validation, and testing, with stratified sampling to maintain class balance. Five-fold cross-validation tuned hyperparameters, and I applied SMOTE to handle the class imbalance inherent in return data. Evaluation used standard metrics: accuracy, precision, recall, F1-score, and AUC-ROC.

SHAP analysis revealed which features matter most for predictions. Historical return rate dominated with a SHAP value of 0.342, meaning customers who've returned products before are likely to do so again. Browsing duration between 8-12 minutes showed a SHAP value of 0.187—apparently there's a sweet spot where engaged browsers convert but aren't researching so thoroughly that they'll return. Product comparison count (0.156), review reading time (0.089), and price point (0.074) rounded out the top predictors.

For customer segmentation, I used a simple threshold: predicted return probability above 0.5 designates high-risk customers. This achieved 97.2% accuracy for high-confidence predictions (those with probability above 0.7 or below 0.3). The resulting segments showed strong separation: high-risk customers (64.2% of the population) returned items 48.7% of the time, while low-risk customers (35.8%) returned just 11.3% of purchases.

### 3.4 Reinforcement Learning for Policy Optimization

I formulated return policy optimization as a Markov Decision Process. The state space captures everything relevant about a transaction: the predicted return probability,

product category, price level, customer's historical return rate, and contextual factors like time and season. Mathematically,  $s_t = [\text{risk\_score}, \text{category}, \text{price}, \text{history}, \text{context}]$  where risk\_score ranges from 0 to 1, category takes 12 discrete values, and the other components are appropriately normalized.

The action space includes five possible interventions: doing nothing (standard 30-day returns, no fees), extending the return window to 45 days for loyal customers, applying a 15% restocking fee for high-risk segments, offering AR virtual try-on for fashion and furniture, or providing enhanced product information with proactive chat support.

My reward function balances multiple objectives:

$$R(s_t, a_t) = \text{Revenue} - \text{Return\_Cost} - \text{Intervention\_Cost} + \text{Satisfaction\_Bonus}$$

More specifically, if a return occurs, the reward is the product price times one minus the margin loss (typically 20-40%), minus the intervention cost and return processing cost. If there's no return, I earn the full price minus intervention cost, plus a satisfaction bonus for maintaining customer happiness.

I implemented Q-learning with epsilon-greedy exploration:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [R(s_t, a_t) + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)]$$

The learning rate ( $\alpha = 0.1$ ) controls how quickly the agent updates its estimates. The discount factor ( $\gamma = 0.95$ ) determines how much it values future rewards. The exploration rate started at  $\epsilon = 0.2$  and linearly decayed to 0.05 over 10,000 training episodes, gradually shifting from exploration to exploitation.

To address limited data availability, I employed transfer learning. I pre-trained the Q-network on synthetic data from my ReturnSim simulator, then fine-tuned on actual SME transaction data. For hyperparameter tuning with small samples, I used Bayesian optimization rather than exhaustive grid search. In cold-start scenarios with fewer than 1,000 transactions, I initialize Q-values using

industry averages and maintain higher exploration ( $\epsilon = 0.5$ ) for the first three months.

### 3.5 Intervention Mechanisms

The RL agent selects from four intervention types, each with different cost-benefit profiles. AR virtual try-on costs \$2.50 per use but reduces returns by 34% through WebGL-based rendering of products in fashion and furniture categories. I trigger this for high-risk customers (predicted return probability exceeding 0.6) purchasing relevant items.

AI size recommendations cost just \$0.10 per use and cut returns by 28% through collaborative filtering based on similar customers' measurements. These deploy automatically for all apparel and footwear purchases, showing confidence scores to help customers trust the suggestions.

Enhanced product visualization costs \$0.50 per session, providing 360-degree views and dimension overlays. This reduces returns by 18% and triggers for high-value items (over \$100) when the customer falls into the high-risk segment.

Proactive chatbot support is the most expensive intervention at \$3.00 per interaction, but reduces returns by 15% by monitoring browsing behavior for uncertainty signals. When the system detects confusion (uncertainty score exceeding 0.7), it offers assistance with escalation to human agents for complex questions.

## 4 Experimental Validation

### 4.1 Simulation Testing

I developed ReturnSim, a Python-based simulation platform calibrated on the German dataset. Cross-validation against a 6-month holdout set showed mean absolute percentage error of just 4.3%, indicating the simulator reliably represents real-world dynamics.

Within this environment, I compared my RL approach against four baselines. The static policy maintains fixed 30-day returns without interventions. The rule-based system applies a

simple heuristic: if predicted return probability exceeds 0.6, charge a restocking fee. The supervised learning baseline makes predictions but doesn't optimize policy. Finally, random policy selection provides a lower bound by choosing actions arbitrarily.

## 4.2 Real-World Deployment

Beyond simulation, I deployed the system in a real Indian marketplace selling bathroom accessories on Amazon India and Flipkart. This seller processes about 800 orders monthly with a baseline return rate of 35.2%, selling products like toilet seat covers, faucets, and bathroom fixtures.

The 6-month pilot (July through December 2025) used A/B testing with 50% of customers in each group. This real-world validation proved crucial for understanding practical challenges and cross-cultural adaptation needs. Implementation cost was relatively modest: ₹45,000 for initial setup plus ₹8,000 monthly operational expenses.

## 4.3 Evaluation Metrics

I assessed performance across multiple dimensions. Primary metrics included return rate reduction, cost savings from avoided processing, and customer satisfaction measured through post-purchase NPS surveys. I calculated ROI by comparing savings against implementation costs.

For the ML components, I tracked accuracy, precision, recall, F1-score, AUC-ROC, and examined confusion matrices to understand error patterns.

For the RL policy, I monitored cumulative reward over training episodes, convergence speed toward optimal policies, and policy stability measured as variance in action selection across similar states.

# 5 Results

## 5.1 Prediction Performance

Table 2 shows how different models performed on the German test set. My ensemble model reached 90.8% accuracy, beating individual approaches by 1.5 to 6.1 percentage points. Gradient Boosting provided the strongest single-model results at 89.3%, validating its reputation as a strong baseline. LASSO's main contribution came through feature selection—it trimmed the feature set from 284 to 47 without sacrificing accuracy.

Model	Accuracy	Precision	Recall	F1
Logistic Regression	72.1%	68.3%	64.7%	66.4%
Random Forest	84.7%	81.2%	78.9%	80.0%
Gradient Boosting	89.3%	86.7%	84.2%	85.4%
<b>Ensemble Model</b>	<b>90.8%</b>	<b>88.5%</b>	<b>86.9%</b>	<b>87.7%</b>

Table 2: Comparing model performance on test data

## 5.2 Segmentation Accuracy

The customer segmentation proved quite accurate. High-risk customers (64.2% of the total) actually returned items 52.3% of the time—my model predicted 51.8%, an error of just 0.5%. Low-risk customers (35.8%) returned 12.7% versus my prediction of 13.1%, a 0.4% error. For high-confidence predictions, segmentation accuracy hit 97.2%, confirming that threshold-based classification works well despite its simplicity.

## 5.3 Policy Learning and Impact

The RL agent converged after 8,400 training episodes in simulation. It learned sensible behaviors: focus expensive interventions on high-risk customers (42.8% got AR try-on, 31.7% saw restocking fees) while leaving low-risk customers mostly alone (67.2% received no intervention).

Overall, returns dropped 32.0%, from 24.7% to 16.8%—a highly significant reduction ( $p < 0.001$ ,  $\chi^2 = 487.3$ ). Results varied by category: Fashion/Apparel saw 29.7% reduction, Electronics 25.7%, Furniture 29.5%, Home Goods 28.6%, and Beauty/Personal Care 31.2%.

Looking at individual interventions, AR virtual try-on proved most effective, cutting returns by 34.2% with net benefit of \$18.70 per use after accounting for costs and prevented returns. AI size recommendations reduced returns 28.1% with \$12.30 net benefit and 67% customer adoption. Enhanced visualization reduced returns 18.4% with \$8.90 benefit and 41% higher engagement.

For a mid-sized SME handling 500,000 returns annually, the business case looks compelling. Annual savings reach \$2.3 million against implementation costs of \$275K (including \$180K initial setup and \$95K annual maintenance). That's a 736% ROI with payback in just 3.6 months. Customer satisfaction improved too: NPS jumped 18 points (from 42 to 60), retention increased 12%, and positive reviews rose 14%.

## 5.4 Indian Marketplace Results

The Indian deployment showed the approach generalizes across cultures, though with interesting adaptations. Returns fell 27.0%, from 35.2% to 25.7%, using AI compatibility checking and visual sizing guides rather than expensive AR.

With implementation costs of just ₹45,000 setup plus ₹8,000 monthly, the monthly savings of ₹33,000 delivered 109% ROI—not as dramatic as simulation suggested, but still clearly worthwhile.

Several cultural insights emerged. Size recommendations achieved only 42% adoption, lower than expected, partly due to WhatsApp message visibility issues. Interestingly, SMS outperformed WhatsApp for Indian customers (78% vs 56% open rates). I also observed stronger preference for human interaction in high-involvement purchases. Most notably, simplified interventions (basic size guides plus proactive messaging) achieved 75% of AR's benefits at 5% of the cost, suggesting SMEs should start with low-cost interventions before investing in expensive technology.

An ablation study showed how the components work together. Prediction alone (no RL) achieved 22.1% returns, a 10.5% improvement over baseline. Adding RL with static features reached 19.4% (21.5% improvement). RL with segmentation hit 17.8% (28.0% improvement). The full system combining RL, segmentation, and interventions reached 16.8% (32.0% improvement), demonstrating genuine synergy between components.

## 6 Discussion

### 6.1 Practical Considerations for SMEs

Several factors make this approach accessible for smaller businesses. The system runs on standard cloud platforms (AWS or GCP) with response times under 100 milliseconds. It achieves over 85% accuracy with as few as 10,000 transactions—far below the 250,000+ typical in academic studies. SMEs can deploy incrementally, starting with low-cost interventions before investing in AR or advanced AI.

SHAP values provide interpretability throughout, helping business owners understand and trust the system's decisions. The RL agent automatically balances costs and benefits, reserving expensive interventions (like \$2.50 AR try-on) for high-risk, high-value transactions while broadly deploying cheap interventions (like \$0.10 size recommendations).

### 6.2 Limitations and Future Directions

Several limitations deserve mention. The cold-start problem remains challenging for SMEs with fewer than 1,000 transactions, though I've partially addressed this through transfer learning and industry priors. AR development costs (\$50,000-\$150,000) may be prohibitive, but tiered alternatives like enhanced images and size guides provide most benefits at fraction of the cost.

Cultural adaptation needs careful attention when moving across markets—my Indian deployment required market-specific tuning. Privacy compliance (CCPA, GDPR) adds complexity around anonymization and consent management.

I deliberately prioritized interpretability and actionability over maximum accuracy. The ensemble achieves 90.8%, but deep learning approaches might reach 93%. I use relatively simple feature engineering (284 features) compared to large-platform systems (1000+). I employ few-shot learning rather than data-hungry deep models. These trade-offs make sense for SME contexts where explainability and computational efficiency matter more than squeezing out the last percentage point of accuracy.

Future research should explore several directions. Transformer architectures could better model sequential behavior patterns. Graph neural networks might address cold-start problems through customer similarity networks. Multi-market expansion would benefit from domain adaptation techniques. Online learning could enable continuous adaptation without periodic retraining. Causal inference techniques would support more robust counterfactual policy evaluation.

## 7 Conclusion

This work presents the first reinforcement learning framework specifically designed for e-commerce returns management in SME contexts, with explainable AI built in from the start. I've shown that sophisticated ML remains feasible even with just 100,000 transactions—far fewer than the 250,000+ common in academic research. Real-world

validation achieved 27-32% return reductions with 109-736% ROI, demonstrating that SMEs can realize substantial benefits without investing in expensive AR technology.

SHAP-based interpretability addresses a crucial barrier to adoption: business owners can see why the system recommends particular actions, building trust in automated decision-making. Cross-cultural validation from German to Indian markets proved the approach generalizes, though with interesting cultural nuances (like SMS outperforming WhatsApp in India).

The open-source ReturnSim simulator and tiered intervention approach lower barriers to ML adoption for smaller businesses. As e-commerce continues expanding globally, empowering SMEs with accessible, data-efficient AI tools becomes increasingly critical for competitive digital retail. Future work should explore transformer architectures for complex sequential patterns, graph neural networks for cold-start scenarios, and causal inference for robust policy evaluation.

## Acknowledgments

I thank the German e-commerce platform for sharing anonymized transaction data that made this research possible. My gratitude extends to the Indian SME partner who collaborated on real-world validation, and to the Department of Computer Science and Applications at Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya (SCSVMV) for supporting this research.

## References

- [1] D. Karl, Forecasting e-commerce consumer returns: A systematic literature review. *Management Review Quarterly* **75**, 369-424 (2025)
- [2] V. Krishnaswamy, R. Deepa, H. Sharma, Business process redesign for reducing undelivered product return losses in e-commerce: An explainable AI approach. *Journal of International Technology and Information Management* **33**, 207-239 (2025)

[3] P. Amblikar, V. Dohale, A. Gunasekaran, V. Bilolikar, Product returns management: A comprehensive review and future research agenda. *International Journal of Production Research* **60**, 3920-3944 (2022)

[4] R. Cui, S. Gallino, A. Moreno, D.J. Zhang, Predicting product return volume using machine learning methods. *Manufacturing &*

*Service Operations Management* **22**, 778-796 (2020)

[5] Q.H. Duong, L. Zhou, M. Meng, T. Van Nguyen, P. Ieromonachou, D.T. Nguyen, Understanding product returns: A systematic literature review using machine learning and bibliometric analysis. *International Journal of Production Economics* **243**, 108340 (2022)