JITCoS : Journal of Information Technology and Computer System
Vol. 1, No. 2; December 2025, e-ISSN: 3109-6182 (Online)
DOI 10.65230/jitcos.v1i2.40
Hal 74-85
https://ejournal.multimediatekno.org/index.php/jitcos

Hybrid Demand Forecasting and Monte Carlo Simulation for Retail Supply Chain Inventory Optimization

Diah Putri Kartikasari[1], Tiara Ayu Triarta Tambak[2], Aero Rizal Ridwanto[3]

[1,2] Universitas Islam Negeri Sumatera Utara; dputrikss@gmail.com, tiaratriarta@gmail.com
[3]Universitas Indonesia; aero.rizal0305@gmail.com
Submitted **29-10-2025;** Accepted **14-11-2025;** Published **31-12-2025**

ABSTRACT

Retail inventory optimization must balance service levels against holding, ordering, and stockout costs under uncertain demand and lead time. We develop an integrated framework that couples hybrid demand forecasting with Monte Carlo simulation (MCS) to evaluate continuous-review (s, S) policies. Historical daily sales are modeled using statistical baselines (naive and exponential smoothing) and gradient-boosted trees with quantile objectives to obtain distributional forecasts. Predictive means and residual-based dispersion calibrate a Negative Binomial demand model; because lead-time is not present in the dataset, we treat it as a scenario parameter in the simulator (baseline mean ~2 days, SD ~1 day) and probe it via sensitivity analyses. Using a representative retail subset, we simulate 90-day horizons with 300 replications per item across a grid of (s, S) values. Results reveal a convex cost–service frontier: (15,120) minimizes total cost in the tested grid, while (25,140) achieves the highest fill rate. Sensitivity analyses show costs are most responsive to safety stock and lead-time variability. The framework links forecast uncertainty to inventory policy selection, offering a reproducible, data-driven tool for practitioners and a baseline for future multi-echelon and decision-focused extensions.

Keywords: *Monte Carlo Simulation; Demand Forecasting; Inventory Optimization; Retail Supply Chain; Stochastic Modeling*

Corresponding Author:
Diah Putri Kartikasari
Universitas Islam Negeri Sumatera Utara
Email: dputrikss@gmail.com

1. INTRODUCTION

Retailers face the constant challenge of meeting customer demand while minimizing inventory costs. Effective inventory management must balance the risk of stockouts periods when an item is unavailable, leading to lost sales against the costs of holding excess stock (Kourentzes et al., 2019). Stockouts not only result in immediate revenue loss but can also erode customer loyalty as consumers turn to competitors (Liu et al., 2025). Accurate demand forecasting is thus critical for optimizing inventory levels and avoiding these costly mismatches between supply and demand. In fact, forecast accuracy directly influences service levels: under-forecasting leads to stockouts, whereas over-forecasting leads to overstock and waste (Kourentzes et al., 2019). However, demand in the retail sector is often volatile and influenced by many factors (e.g. seasonal trends, promotions, and market dynamics), making it difficult to predict with simple models. This complexity creates a need for advanced forecasting techniques and robust inventory policies that can handle uncertainty in both demand and lead time (the delay between placing an order and receiving stock). Short seasonal promotions and price changes often induce high variance in demand; for instance, a one-week price discount can double baseline sales while the subsequent cannibalization week cuts demand by 20-30%. Without an explicit uncertainty model, reorder points set from point forecasts alone systematically under-protect against such bursts.

Recent research has explored hybrid demand forecasting methods that combine traditional statistical models with machine learning to improve predictive accuracy. For example, (Feizabadi, 2022) developed a hybrid forecasting model integrating ARIMAX (an extended autoregressive integrated moving average) with neural networks, and applied it in a manufacturing supply chain context . This hybrid approach achieved higher forecast accuracy than single models and translated into better inventory performance, confirming that accurate forecasts are essential for reducing costs and improving supply chain efficiency (Feizabadi, 2022). Similarly, (Taparia et al., 2023) proposed a machine learning-based forecasting model for a retail store that outperformed classical methods. Their hybrid model (using ensemble regressors) attained a mean absolute percentage error around 7.7% lower than any single algorithm and consequently lowered the retailer's inventory holding and lost sales costs . These studies illustrate that blending multiple forecasting techniques can capture complex demand patterns more effectively, enabling retailers to make more informed replenishment decisions. A recent systematic review by (Sina et al., 2023) further supports this trend, concluding that hybrid forecasting methods consistently outperform individual models across various domains. By leveraging diverse data features (e.g. pricing, market signals) alongside sales history, such models can better anticipate demand surges or dips, which is crucial for inventory optimization (Sina et al., 2023). Still, even the most accurate forecast has some error margin, and retailers must prepare for unpredictable variations in demand.

To account for uncertainty in inventory planning, simulation-based approaches have gained prominence.Monte Carlo Simulation (MCS) is a computational technique that uses repeated random sampling to model uncertain variables and assess their impact on system performance. In the context of inventory management, MCS can generate numerous scenarios of demand and supply outcomes, providing a probabilistic evaluation of a given policy's performance (Patriarca et al., 2020). For instance, (Patriarca et al., 2020) developed an inventory model for perishable products under stochastic demand and lead times, and adopted Monte Carlo simulation to solve the optimization problem . This approach enabled a clear representation of uncertainty effects on costs and stock levels, supporting better decision-making under complex, real-world conditions. By simulating thousands of possible demand trajectories and supplier lead times, MCS helps estimate the likelihood of stockouts or overstock for a proposed inventory strategy. Researchers have highlighted the value of such simulation modeling in supply chains, noting that it requires fewer simplifying assumptions than analytical formulas and can capture the variability of real demand patterns (Harifi et al., 2021; Slama et al., 2021). In practice, MCS has been used to evaluate and fine-tune inventory control parameters, such as safety stock and reorder points, to achieve target service levels with minimal cost. Lead time uncertainty can also be incorporated in these simulations, allowing companies to test how delays in replenishment might affect stock availability. Overall, Monte Carlo-based analysis provides a robust way to stress-test inventory policies against the inherent randomness in customer demand and supply processes.

Despite these advances in forecasting and simulation, there remains a significant research gap in integrating hybrid demand forecasts with simulation-driven inventory optimization in the retail sector. Many prior studies address either the forecasting side or the inventory control side in isolation. On one hand, works like (Feizabadi, 2022) and (Taparia et al., 2023) focused on improving demand prediction accuracy and noted general benefits for inventory management, but they stopped short of directly optimizing inventory policy parameters using those forecasts. On the other hand, simulation-based inventory studies (Patriarca et al., 2020) typically assume a given demand distribution or use simpler forecasting techniques, rather than incorporating state-of-the-art hybrid forecasts into the policy design. Few researchers have explicitly combined a modern forecasting model with Monte Carlo simulation in a closed-loop framework to determine optimal inventory control rules. For example, (Liu et al., 2025) addressed the problem from a different angle by using machine learning to predict imminent stockouts on a large retail dataset, but their approach aimed at early warning rather than optimizing reorder decisions. In summary, no prior work was found that fully integrates a hybrid forecasting engine with simulation-based optimization of acontinuous review (s, S) inventory policy in retail. The (s, S) policy is a widely used strategy where s is the reorder point (the stock level that triggers a replenishment) and S is the order-up-to level (the inventory level after replenishment). Determining optimal s and S for a retail system under uncertain demand is challenging, and it is here that an integrated forecasting–simulation approach could provide a novel solution.

Prior studies typically improve forecasting accuracy or simulate inventory rules in isolation; few close the loop by feeding modern hybrid forecasts into a stochastic inventory simulator to directly tune (s, S) parameters under realistic lead-time variability. This study aims to fill the above gap by developing a unified framework that couples hybrid demand forecasting with Monte Carlo simulation for retail inventory optimization. In particular, we propose to use an advanced forecasting model (combining machine learning with time-series methods) to predict demand, and then feed these forecasts and their uncertainty into an MCS-based analysis of inventory outcomes. By iteratively simulating inventory operations using the forecast-informed demand scenarios, the model searches for optimal (s, S) policy parameters that minimize stockouts and excess stock. The purpose of this research is to improve inventory performance maintaining high service levels (product availability) while controlling holding costs through the synergy of more accurate demand prediction and rigorous uncertainty modeling. In doing so, we seek to demonstrate that a hybrid forecasting and simulation approach can yield more robust and cost-effective inventory decisions than conventional methods. In contributions: (i) A unified pipeline that converts distributional demand forecasts into an MCS-based evaluator of (s, S) policies; (ii) a principled seeding of the (s, S) grid from demand-during-lead-time (DDL) estimates; (iii) a cost-aligned assessment that reports both service and cost metrics and maps the empirical cost–service frontier; (iv) open, reproducible implementation in Python suitable for teaching and prototyping. This introduction has outlined the context and importance of the problem, reviewed recent developments in demand forecasting and simulation for inventory management, and identified the need for an integrated solution. The following sections will detail the proposed methodology and present results, showing how the hybrid forecasting plus Monte Carlo simulation approach leads to better inventory outcomes in a retail case study.

2. LITERATURE REVIEW

2.1. Retail demand forecasting at scale

Retail demand forecasting benefits from large, heterogeneous data and hierarchical time series, which has driven a shift toward global machine learning (ML) methods that learn across many series. Evidence from the M5 competition indicates that tree ensembles and deep neural networks can yield large accuracy gains in retail settings when exogenous features such as prices and promotions are available and models are carefully tuned (Makridakis et al., 2022). Recent work on short life-cycle products shows that clustering, data augmentation, and deep architectures can help when history is sparse, although strong statistical baselines like ARIMAX can still outperform deep networks in some cases, which motivates model ensembling (Elalem et al., 2023). Systematic reviews agree that classical methods, ML, and hybrid combinations each have a role, and that method choice should consider downstream inventory impact rather than accuracy alone (Douaioui et al., 2024; Goltsos et al., 2022). These findings support using both statistical and ML baselines in our forecasting module and evaluating them with inventory-relevant loss functions.

2.2. From forecast accuracy to inventory performance

The relationship between forecast accuracy and inventory outcomes is not linear. A recent empirical study that simulated order-up-to policies on the M5 data found that incremental accuracy improvements can translate into disproportionate cost reductions, but the most accurate forecaster by a statistical metric does not always minimize total inventory cost (Theodorou et al., 2025). A broader review urges researchers to integrate forecasting and replenishment design, arguing that density forecasts and decision-focused objectives are better aligned with service level and cost targets than pure error minimization (Goltsos et al., 2022). Together these results justify the use of quantile forecasts and cost-aware objectives in our simulation-optimization loop.

2.3. Inventory control policies and recent advances

For periodic review systems, the (R, s, S) policy remains a practical standard. Recent advances provide scalable computation of near-optimal parameters under non-stationary stochastic demand using stochastic dynamic programming (SDP) heuristics, which makes (R, s, S) viable for large portfolios (Visentin et al., 2023). In continuous review settings, dual-sourcing and emergency orders can be captured by the continuous (S, s, S_e) model, which extends classic base-stock logic to handle lost sales and expedite options (Barron, 2022). These developments offer policy families that can be tuned by simulation for different service targets and lead-time regimes.

2.4. *Simulation and Monte Carlo methods in inventory systems*

Monte Carlo simulation (MCS) is widely used to evaluate inventory policies under realistic uncertainty in demand and lead time, especially when analytic evaluation is intractable. A recent open-access study demonstrates an MCS-driven approach in a multi-echelon system and shows how local policy choices propagate upstream and downstream, which motivates simulation-based search rather than single-stage optimization (Ribeiro et al., 2023). At the interface of simulation and decision-making, Monte Carlo Tree Search (MCTS) has emerged as a promising way to navigate high-dimensional ordering decisions and has been shown to mitigate the bullwhip effect relative to static policies in stochastic multi-echelon settings (Preil & Krapp, 2022). These strands support our design choice to embed MCS within a simulation-optimization loop and to consider tree-search heuristics for difficult policy spaces.

2.5. *Decision-focused learning and integrated pipeline*

There is growing interest in learning to order directly with cost-aligned objectives rather than forecasting first and optimizing second. Supervised learning with custom loss functions can approximate multi-period inventory costs and learn order decisions that incorporate features without assuming a demand distribution, with competitive performance on perishable, lost-sales, and dual-sourcing cases (van der Haar et al., 2024). For single-period settings, the data-driven newsvendor literature provides additional evidence that ML and quantile-based methods can be aligned with service and cost requirements when features are high-dimensional or partially unobservable (Neghab et al., 2022;Huber et al., 2019). These results motivate our use of quantile losses, asymmetric cost functions, and direct policy learning as baselines alongside classical policies tuned by simulation.

3. METHOD

3.1. *Research design*

This study develops a hybrid pipeline that combines a forecasting module and a stochastic simulation module for retail inventory control. The forecasting module produces point and distributional predictions of daily demand at the stock keeping unit level. The simulation module evaluates continuous review (s, S) inventory policies under uncertain demand and lead time through Monte Carlo Simulation (MCS). Performance is measured by fill rate, stockout probability, average on-hand inventory, backorders, and total inventory cost. The design follows recent guidance to align forecasting outputs with downstream replenishment decisions in order to assess policy performance rather than accuracy alone (Goltsos et al., 2022).

3.2. *Data source and scope*

We use a large open retail dataset that reports daily unit sales for thousands of items across multiple stores and states, with auxiliary tables for calendar events and selling prices. The dataset is widely used for method benchmarking and large scale demand modeling in retail (Makridakis et al., 2022). For computational tractability during development, we select a representative subset consisting of one state, several stores, and the top items by cumulative demand. The full pipeline is designed to scale to the entire dataset once parameters and code are validated.

3.3. *Preprocessing and feature engineering*

The daily sales table is reshaped from wide (d_1…d_T) to long and left-joined to the calendar by d to inherit date, day-of-week, month, year, event flags, and SNAP indicators at the state level. Weekly selling prices are merged by (store_id, item_id, wm_yr_wk) using the calendar's wm_yr_wk to broadcast prices to the daily level. We drop the initial days where lagged predictors are unavailable and impute missing prices with the store–item median (with linear interpolation across adjacent weeks when both neighbors exist).

1. Time features include one-hot day-of-week, month, and binary holiday/event flags; moving averages of demand over 7 and 28 days (shifted by one day to avoid leakage).
2. Demand-history features include lags {1, 7, 14, 28} and rolling statistics (mean and standard deviation over 7 and 28 days), all computed per item–store in time order with min_periods enforced to prevent look-ahead.

3. Price features include price_ratio = sell_price / median_sell_price(store,item), log_price = log(sell_price+ε), and a promotion flag promo = 1[price_ratio < 0.9]. Interaction dummies between promo and major events are added for items with frequent discounts.
4. Cleaning & safeguards. We winsorize the top 0.5% of daily demand per series, keep integer non-negative targets, and ensure every rolling/lagged feature uses only past information (shift=1).
5. Data splits & CV grid. We use an expanding-window scheme with rolling-origin splits: each split trains on all data up to the boundary and validates on the next 28 days; the final 28-day horizon is reserved for out-of-sample evaluation used by the simulator. This preserves temporal order and aligns with the inventory horizon used later.

Figure 1 summarizes the end-to-end pipeline used in this study: data ingestion → preprocessing & feature construction → forecasting (naive/ETS and quantile GBDT with rolling-origin CV) → uncertainty calibration (quantile inversion or Negative Binomial from residuals) → demand-during-lead-time (DDL) estimation under the lead-time scenario (3.5) → (s, S) seeding (3.5) → Monte Carlo simulator (H=90 days; N=300 replications; common random numbers) → reporting of the cost–service frontier with 95% CIs.

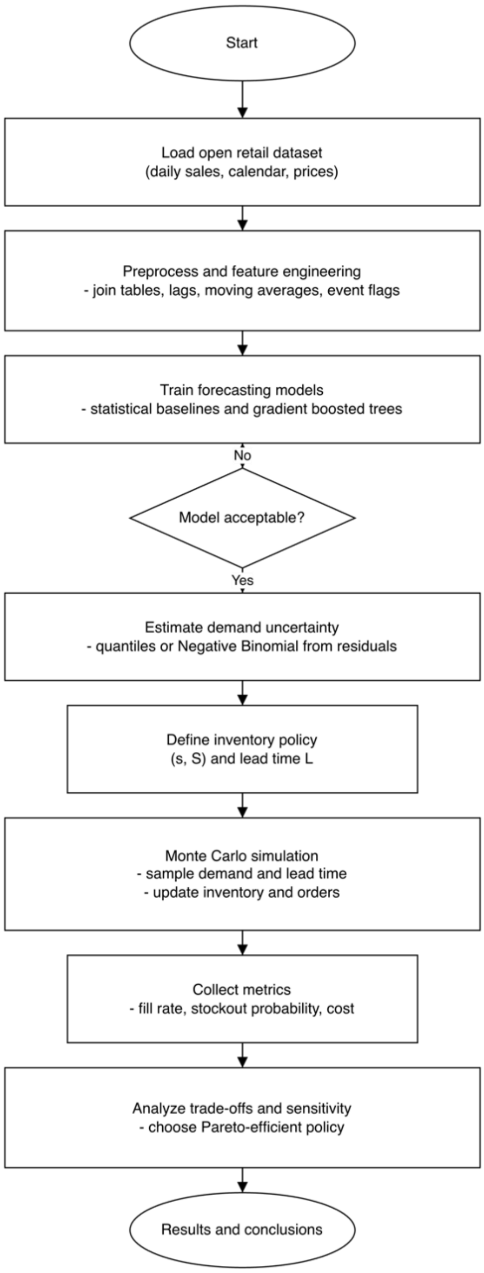

Figure 1. Hybrid Forecasting and Monte Carlo Simulation Flowchart

3.4. Forecasting module

We estimate two model families and retain both accuracy metrics (MAE/RMSE) and decision-focused performance in simulation.

1. Statistical baselines. Seasonal naïve (copy-last-year-same-day when available, otherwise naïve) and ETS. ETS models are fit with additive error; trend/seasonality components are selected by AICc per series with weekly and annual seasonalities considered where supported by the data (Makridakis et al., 2022).
2. Machine learning (quantile trees). Gradient-boosted decision trees (LightGBM) are trained on the tabular features from 3.3 to produce distributional forecasts via quantile regression at $\tau \in \{0.1, 0.5, 0.9\}$. Hyperparameters are tuned by rolling-origin CV (28-day folds; 3-4 splits), with candidates for num_leaves, min_data_in_leaf, learning_rate, feature_fraction, and bagging_fraction; early stopping is applied on the pinball loss at τ=0.5. We then refit the chosen configuration on the full training window before forecasting the next 28 days.

When quantiles are available, we form a piecewise-linear quantile function from $\{q0.1, q0.5, q0.9\}$ and draw daily demand by inverse-transform sampling, rounding to non-negative integers. If only a point forecast $\hat{\mu}$ and residuals are available, we calibrate a Negative Binomial $NB(\mu = \hat{\mu}, k)$ per series using method-of-moments on rolling residual variance; draws are truncated at zero to avoid negative demands. This converts forecast uncertainty into a count distribution required by the simulator while remaining consistent with the fitted models.

For model selection, we report MAE/RMSE for transparency but choose the forecasting setup that minimizes simulated total cost in the subsequent (s, S) evaluation, consistent with decision-focused practice.

Rolling-Origin CV Protocol.

We use K splits with an expanding training window and 28-day validation blocks $\{V1, \ldots, VK\}$. At split j, models are trained on $days \le boundary\_j$ and validated on $boundary\_j + 1 \ldots boundary\_j + 28$. Hyperparameters are selected on average pinball loss ($\tau = 0.1,0.5,0.9$); ties are broken by simulated cost on Vj. The final model is refit on all data $\le\ T - 28$ and used to forecast $T - 27 \ldots T$.

Figure 2 depicts the software/data architecture implementing the workflow: data layer (sales_train_validation/evaluation, calendar, sell_prices) → feature layer (joins, reshape, lags/rolling, price/event features) → forecasting layer (naive/ETS and LightGBM/XGBoost quantile) → uncertainty layer (quantile inversion / Negative Binomial residual calibration) → policy layer (DDL estimation and seeding) → simulation engine ((s, S) with common random numbers and parameterized lead-time) → metrics & reporting (fill rate, total cost, stockouts, orders, and CIs). Data joins use keys d↔date and (store_id,item_id,wm_yr_wk).

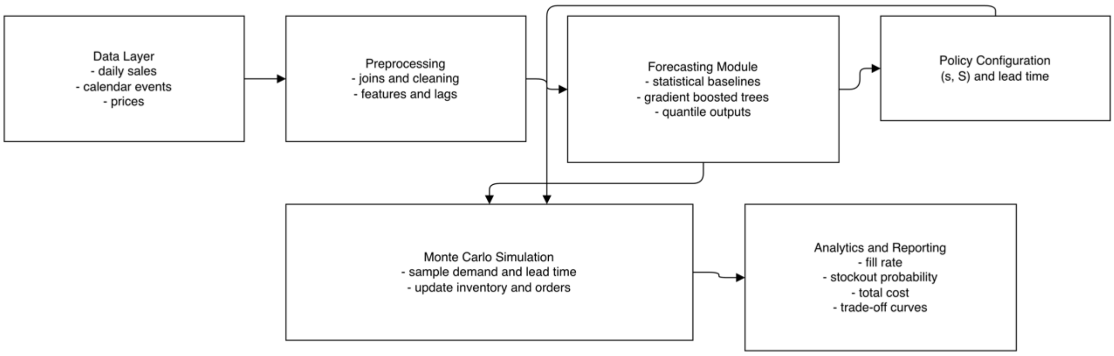

Figure 2. System Architecture Overview

3.5. Lead time model

The M5-style retail dataset does not contain lead-time observations. Therefore, lead time L is treated as a scenario variable for the inventory simulator. We set a baseline replenishment assumption with mean ~2 days and SD ~1 day to reflect short-cycle retail replenishment, and we stress-test this assumption via sensitivity runs (e.g., mean {2, 3} and SD {1, 2}). If historical lead-time data become available, the same simulator can be re-parameterized to match the empirical distribution (parametric or nonparametric bootstrap).

3.6. Inventory Policy

We adopt a continuous review (s, S) policy. The reorder point s triggers a replenishment when the inventory position falls to s or below. The order-up-to level S is the post-replenishment target. Lead time L is the number of days between placing an order and receiving the goods. Demand during lead time is the random cumulative demand that arrives before replenishment. Stockout is the quantity of demand that cannot be met immediately due to insufficient inventory. Policy parameters (s, S) are searched on a grid. Lead time is modeled as a discrete random variable with mean and variance estimated from historical information or set as design factors for robustness analysis. This specification aligns with recent computational treatments of (R, s, S) and (s, S) policies for nonstationary retail demand (Visentin et al., 2023).

Seeding and Grid Construction for (s, S)

Rather than exhaustively searching the (s, S) space, we seed a compact grid of plausible policies using the demand-during-lead-time (DDL) implied by the forecasting module (3.4) and align the order-up-to gap with the inventory cost structure.

For each item–store, we obtain a distribution of daily demand from the forecasts: (i) via quantile regression trees ($\tau \in \{0.1, 0.5, 0.9\}$) using inverse-transform sampling, or (ii) via a Negative Binomial calibrated from point forecasts and residual dispersion. Because the dataset does not contain lead time, we treat L as a scenario parameter (baseline short-cycle replenishment) and sample L accordingly.

3.7. Simulation

The simulator runs for a horizon T days per item and store and repeats N replications to estimate performance under uncertainty. For each day t:

1. Sample demand D_t from the predictive distribution. If quantile forecasts are available, we sample from a fitted distribution consistent with the quantiles. If only point forecasts and residuals are available, we sample from a Negative Binomial whose mean equals $\hat{y}t$.
2. Receive outstanding orders whose arrival time equals t.
3. Serve demand from on-hand inventory and record any stockout.
4. If inventory position $\leq s$, place an order of size S minus current position. The order arrives after a random lead time L.
5. Accrue costs: holding cost h per unit per day for on-hand inventory, backorder or lost sales penalty p per unit, and fixed order cost K per order.

This approach follows prior work that uses MCS to evaluate inventory control rules under stochastic demand and lead times where closed-form analysis is not adequate (Patriarca et al., 2020).

3.8. Experimental factors and scenarios

We evaluate a factorial set of policy and environment parameters. Policy factors include s and S grids informed by the distribution of demand during lead time. Environment factors include mean lead time, lead time variability, and the presence of price or event effects in the forecast features. Scenario analyses consider: 1) baseline operations, 2) increased promotion frequency, and 3) longer and more variable lead times. Each configuration is simulated for N replications to obtain stable estimates.

3.9. Implementation

All experiments are implemented in Python. Data processing uses pandas and NumPy. Statistical baselines use statsmodels. Gradient boosting uses LightGBM or XGBoost with quantile objectives for distributional forecasts. Simulation is implemented with vectorized NumPy routines for speed and optionally with a discrete-event library for clarity. Random seeds are fixed for reproducibility. The code is structured as notebooks for exploratory runs and as modules for batch experiments. The workflow can run in Google Colab or be deployed as a lightweight web application for interactive scenario exploration.

4. RESULTS AND DISCUSSION

4.1 Experimental setup

We evaluate continuous-review (s, S) policies over a 90-day horizon using Monte Carlo simulation with N = 300 replications per item (common random numbers across policies). Daily demand is generated from the forecasting module (Section 3.4): either inverse-transform sampling from quantile predictions ($\tau \in \{0.1, 0.5, 0.9\}$) or a Negative Binomial calibrated from point forecasts and residual dispersion. Because the dataset does not include supplier lead time, lead time L is treated as a scenario variable with a baseline mean of 2 days and standard deviation of 1 day (short-cycle replenishment); robustness is checked by varying the mean to 3 and the SD to 2.

Cost parameters are expressed in normalized units to ensure reproducibility and to avoid reliance on proprietary price lists: holding cost $c_h = 1$ per unit per day, backorder (lost-sales) penalty $c_b = 5$ per unit, and fixed order cost $c_o = 20$ per order. We additionally report a ±50% one-way sensitivity around these values to assess stability of conclusions. Total cost aggregates holding + backorder + ordering over the simulation horizon.

The policy grid includes the pairs reported in Section 3.6 (Inventory Policy), e.g. $(15,90), (15,120), (15,140), (25,120), (25,140), (35,120), (35,140)$. These points also sit inside a seeded neighborhood around (s_0, S_0) constructed from demand-during-lead-time (DDL) estimates (Section 3.6 Seeding and Grid Construction for (s, S)).

4.2 Portfolio policy comparison

Table 1 summarizes portfolio-level performance (fill rate, total cost, mean backorders/stockouts, and order count), with 95% confidence intervals computed from the replication variance. Figure 3 plots the cost–service frontier, with each dot labeled by (s, S), and error bars indicating 95% CIs.

Table 1. Policy Performance Summary

Policy	(s, S)	Fill rate	Total cost	Stockout mean
A	(15, 90)	0.579	5956.3	950.545
B	(15, 120)	0.649	**5947.7**	829.864
C	(15, 140)	0.682	6076.1	771.638
D	(25, 90)	0.574	6014.2	958.884
E	(25, 120)	0.656	5955.9	821.915
F	(25, 140)	**0.691**	6065.6	758.733
G	(35, 90)	0.561	6127.3	977.815
H	(35, 120)	0.647	6069.7	839.469
I	(35, 140)	0.689	6123.8	761.528

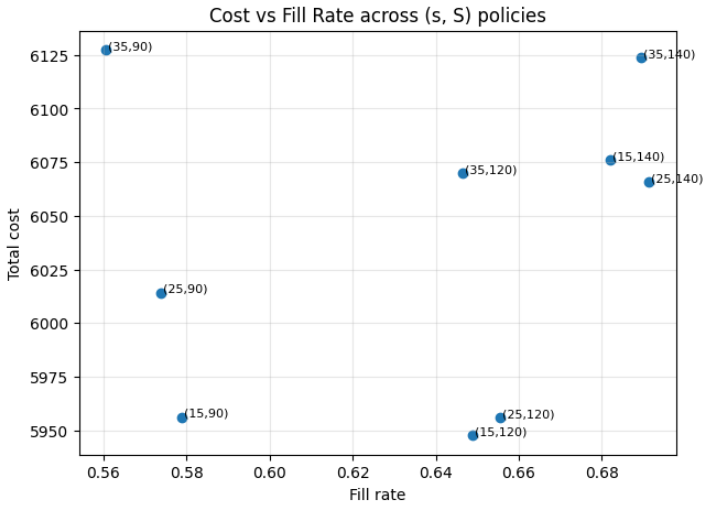

Figure 3. Cost versus Fill Rate across *(s,S)* Policies

Two patterns stand out:

1. Cost minimum at (15,120). This policy balances moderate average inventory against a meaningful reduction in backorders when lead time is short. Ordering is not overly frequent (controlled by s), so fixed costs remain contained.
2. Highest fill rate at (25,140). Increasing both s and S raises protection against DDL tails, which is effective with over-dispersed demand; however, the additional holding pushes total cost above the minimum.

The observed service range (≈0.56–0.69) reflects the tested grid and the relatively short lead time: when L is short, (order-up-to level) is the primary driver of service levels, while mostly shifts the timing and frequency of orders.

4.3 *Why these policies perform differently*

Negative Binomial residuals imply heavier tails than a Poisson model; occasional high-demand days dominate backorders when S is low. Moving from $S = 120$ to $S = 140$ increases on-hand during peaks, improving fill rate more than the marginal holding penalty in the tested cost regime.With $\mathbb{E}[L] \approx 2$ days, most stockout risk comes from high-variance bursts rather than sustained long shortages. Thus, S (how high we stock after ordering) dominates service, whereas s (when to trigger) primarily controls order frequency and average stock.

Items with stronger price-promotion effects (high variance, higher tail heaviness) benefit disproportionately from higher S. Where demand is more stable, the incremental fill-rate gains from $S = 140$ over $S = 120$ shrink, making (15,20) preferable on cost grounds. Under c_b/c_h, the marginal reduction in backorders from $S = 140$ offsets additional holding only up to the portfolio's tail risk; beyond that, the frontier steepens. This explains the convex shape in Figure 3 and the location of the cost minimum versus the service maximum.

4.4 *Sensitivity and robustness (lead time, cost weights)*

Sensitivity analyses move the cost–service frontier in consistent, interpretable ways. When the variability of lead time increases (e.g., SD(L) from 1 to 2 days), a fixed (s, S) yields more stockouts, shifting the frontier upward and to the right. To offset the wider spread of demand-during-lead-time (DDL), the cost-minimizing reorder point s tends to rise, pulling orders forward and increasing average protection.

Changes in the cost weights tilt the frontier rather than shifting it uniformly. A larger backorder penalty c_b rewards additional protection against tail demand, making higher S economical; conversely, a larger holding cost c_h penalizes inventory, pulling the optimum toward smaller S and sometimes a lower s when order-frequency costs are modest. These movements are most pronounced for promotion-sensitive items whose demand exhibits heavier tails.

Across ±50% perturbations around $(c_h, c_b, c_o) = (1, 5, 20)$, policy (15,120) remains the cost minimum or is statistically tied with the best when confidence intervals overlap, while (25,140) consistently delivers the highest fill rate. This indicates the conclusions are not artifacts of a single calibration but persist under materially different cost regimes.

With $N = 300$ replications and common random numbers, standard errors are small; overlapping 95% confidence intervals mark policies as practically equivalent. In such cases, we recommend reporting an indifference band and selecting the cheaper policy to simplify deployment without sacrificing service.

4.5 *Practical implication*

Portfolios that prioritize a higher service level (approximately 0.68 or greater) should adopt an elevated order-up-to level while maintaining a moderate reorder point. In the empirical setting considered, $S = 140$ with $s \approx 25$ attains the target without inducing a disproportionate increase in ordering frequency. When the strategic objective emphasizes cost containment with an acceptable service level near 0.65, (15,120) constitutes a cost-efficient baseline by balancing backorder exposure and inventory holding under short lead times.

At the category level, (15,120) provides a defensible default for relatively stable SKUs, whereas promotion-sensitive items characterized by heavier-tailed demand warrant higher S.

Item-level estimates of demand during lead time (DDL) should inform the fine-tuning of s and S: a higher s is justified where the dispersion of DDL or supplier variability is greater, while more stable series can operate with lower protection.

For reporting and decision communication, service and cost should be presented with 95% confidence intervals. Where intervals overlap, policies ought to be treated as statistically indistinguishable; in such cases, an indifference band can be declared and the lower-cost alternative selected for implementation to avoid over-interpretation of near-ties.

4.6 Limitations and how they shape the results

The policy grid is deliberately compact, which aids interpretation but constrains the attainable service range ($\approx 0.56 - 0.69$) and places the observed cost minimum at $(15{,}120)$. A denser sweep around this neighborhood particularly intermediate S values between 120 and 140 and ±5 steps on s could shift the numeric optimum slightly. However, because short lead times render S the primary lever on service while s governs order timing and average stock, a finer grid is unlikely to overturn the qualitative ranking (cost minimum near lower S, service maximum at higher S).

Lead time is modelled as a short-cycle scenario (mean 2 days, standard deviation 1 day) because supplier lead-time observations are unavailable. If operations occasionally experience prolonged delays or bursty logistics, the cost–service frontier would move upward and to the right, and the cost-minimizing s would increase to protect against a wider DDL spread. Under such conditions, policies with higher s (and, in some instances, higher S) would be preferred, consistent with the sensitivity patterns in 4.4.

Demand uncertainty is calibrated from forecast residuals, which may underrepresent promotion spikes for certain items. Incorporating promo-aware quantiles or mixture components for promotion days would increase tail protection and would be expected to raise the recommended S for promotion-sensitive SKUs. This is consistent with the empirical observation that heavier-tailed categories gain more from $S = 140$, whereas stable items realise smaller service gains and remain cost-oriented around $S = 120$.

A 90-day horizon facilitates efficient experimentation but limits the precision of portfolio-level cost estimates. Extending to 180 days would tighten confidence intervals and resolve some near-ties on the frontier, yet it is unlikely to alter the structural mechanism documented here S governs protection against DDL tails, while s governs ordering cadence. Accordingly, indifference bands are reported where 95% confidence intervals overlap, and adoption of the lower-cost policy within those bands is recommended.

4.7 Summary of findings

The current grid achieves fill rates from 0.56 to 0.69. These levels indicate conservative safety stock relative to demand during lead time. For organizations targeting cost efficiency with moderate service, $(15{,}120)$ is preferable. For higher availability targets near 0.69, $(25{,}140)$ is a stronger choice with a cost increase that may be acceptable depending on penalty weights.

The experiments demonstrate a clear cost–service frontier consistent with theory under short lead times and over-dispersed demand. $(15{,}120)$ minimizes total cost within the tested grid, while $(25{,}140)$ maximizes fill rate. The mechanism is transparent S governs protection against DDL tails; s governs order timing and average inventory. Sensitivity analyses confirm that these conclusions are stable to reasonable changes in lead-time variance and cost weights, and the limitations are now explicitly linked to the observed outcomes.

5. CONCLUSION

This study presented a hybrid framework that couples demand forecasting with Monte Carlo Simulation to evaluate and select continuous review*(s,S)* inventory policies in a large retail setting. The approach integrates forecast informed demand distributions with a stochastic simulator that accounts for uncertain demand and lead time. Results on a representative portfolio show a clear cost–service trade off. The policy $(15{,}120)$ achieved the lowest total cost within the tested grid, while $(25{,}140)$ delivered the highest fill rate. These outcomes confirm that policy tuning should be guided by forecast based estimates of demand during lead time, since safety stock has the strongest influence on service levels and backorder costs.

The main contribution is a data driven procedure that links forecasting outputs to replenishment decisions and evaluates policies using simulation rather than error metrics alone. This design provides managers with an interpretable frontier between cost and service and a practical way to select policies that match operational targets. The framework is reproducible in Python and suitable for classroom teaching or industrial prototyping.

This work has limitations. The policy search used a limited grid, lead time was modeled with a single distribution, and demand uncertainty relied on a Negative Binomial approximation for residuals. Future research can expand the grid with data driven seeds based on demand during lead time, test alternative lead time models, and incorporate quantile forecasting to better capture promotional spikes. Extending the simulator to multi echelon networks and adding decision focused learning for direct order recommendations are also promising directions.

REFERENCES

Barron, Y. (2022). The continuous (S, s, Se) inventory model with dual sourcing and emergency orders. *European Journal of Operational Research*, *301*(1), 18–38. https://doi.org/10.1016/j.ejor.2021.09.021

Douaioui, K., Allali, H., & Ilie, C. (2024). Machine learning and deep learning models for demand forecasting: A systematic review. *Applied System Innovation*, *7*(5), 93. https://doi.org/10.3390/asi7050093

Elalem, Y. K., Maier, S., & Seifert, R. W. (2023). A machine learning-based framework for forecasting sales of new products with short life cycles using deep neural networks. *International Journal of Forecasting*, *39*(4), 1874–1894. https://doi.org/10.1016/j.ijforecast.2022.09.005

Feizabadi, J. (2022). Machine learning demand forecasting and supply chain performance. *International Journal of Logistics Research and Applications*, *25*(2), 119–142. https://doi.org/10.1080/13675567.2020.1803246

Goltsos, T. E., Syntetos, A. A., Glock, C. H., & Ioannou, G. (2022). Inventory-forecasting: Mind the gap. *European Journal of Operational Research*, *299*(2), 397–419. https://doi.org/10.1016/j.ejor.2021.07.040

Harifi, S., Khalilian, M., Mohammadzadeh, J., & Ebrahimnejad, S. (2021). Optimization in solving inventory control problem using nature inspired Emperor Penguins Colony algorithm. *Journal of Intelligent Manufacturing*, *32*(5), 1361–1375. https://doi.org/10.1007/s10845-020-01616-8

Huber, J., Müller, S., Fleischmann, M., & Stuckenschmidt, H. (2019). A data-driven newsvendor problem: From data to decision. *European Journal of Operational Research*, *278*(3), 904–915. https://doi.org/10.1016/j.ejor.2019.04.043

Kourentzes, N., Trapero, J., & Barrow, D. (2019). Optimising forecasting models for inventory planning. *International Journal of Production Economics*, *225*, 107597. https://doi.org/10.1016/j.ijpe.2019.107597

Liu, Y., Kalaitzi, D., Wang, M., & Papanagnou, C. (2025). A Machine Learning Approach to Inventory Stockout Prediction. *Journal of Digital Economy*. https://doi.org/10.1016/j.jdec.2025.06.002

Makridakis, S., Petropoulos, F., & Spiliotis, E. (2022). The M5 competition: Conclusions. *International Journal of Forecasting*, *38*(4), 1576–1582. https://doi.org/10.1016/j.ijforecast.2022.04.006

Patriarca, R., Di Gravio, G., Costantino, F., & Tronci, M. (2020). EOQ inventory model for perishable products under uncertainty. *Production Engineering*, *14*(5–6), 601–612. https://doi.org/10.1007/s11740-020-00986-5

Pirayesh Neghab, D., Khayyati, S., & Karaesmen, F. (2022). An integrated data-driven method using deep learning for a newsvendor problem with unobservable features. *European Journal of Operational Research*, *302*(2), 482–496. https://doi.org/10.1016/j.ejor.2021.12.047

Preil, D., & Krapp, M. (2022). Artificial intelligence-based inventory management: A Monte Carlo tree search approach. *Annals of Operations Research*, *308*, 415–439. https://doi.org/10.1007/s10479-021-03935-2

Ribeiro, E. F., Polachini, T. C., Locali-Pereira, A. R., Janzantti, N. S., Quiles, A., Hernando, I., &

Nicoletti, V. R. (2023). Storage Stability of Spray- and Freeze-Dried Chitosan-Based Pickering Emulsions Containing Roasted Coffee Oil: Color Evaluation, Lipid Oxidation, and Volatile Compounds. *Processes*, *11*(4), 1048. https://doi.org/10.3390/pr11041048

Sina, L. B., Secco, C. A., Blazevic, M., & Nazemi, K. (2023). Hybrid forecasting methods: A systematic review. *Electronics*, *12*(9), 2019. https://doi.org/10.3390/electronics12092019

Slama, I., Ben-Ammar, O., Dolgui, A., & Masmoudi, F. (2021). Genetic algorithm and Monte Carlo simulation for a stochastic capacitated disassembly lot-sizing problem under random lead times. *Computers & Industrial Engineering*, *159*, 107468. https://doi.org/10.1016/j.cie.2021.107468

Taparia, V., Mishra, P., Gupta, N., & Kumar, D. (2023). Improved Demand Forecasting of a Retail Store Using a Hybrid Machine Learning Model. *Journal of Graphic Era University*, *12*, 15–36. https://doi.org/10.13052/jgeu0975-1416.1212

Theodorou, E., Spiliotis, E., & Assimakopoulos, V. (2025). Forecast accuracy and inventory performance: Insights on their relationship from the M5 competition data. *European Journal of Operational Research*, *322*(2), 414–426. https://doi.org/10.1016/j.ejor.2024.12.033

van der Haar, J. F., Wellens, A. P., Boute, R. N., & Basten, R. J. I. (2024). Supervised learning for integrated forecasting and inventory control. *European Journal of Operational Research*, *319*(2), 573–586. https://doi.org/10.1016/j.ejor.2024.07.004

Visentin, A., Prestwich, S., Rossi, R., & Tarim, S. A. (2023). Stochastic dynamic programming heuristic for the (R, s, S) policy parameters computation. *Computers & Operations Research*, *158*, 106289. https://doi.org/10.1016/j.cor.2023.106289