

Improving Supply Chain and Logistics: Predicting Demand and Optimizing Inventory with Historical Sales Data

Wadahfred Kudi¹, Dongxiao Ren², Zhonghua Wang³

^{1,2}School of Artificial Intelligence Information Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China

³Business Department of Human Resources and Party Building, Beijing Guodiantong Network Technology Co., Ltd., Beijing 100000, China

Abstract

Supply chains increasingly operate under volatile demand, uncertain replenishment lead times, promotional shocks, and disruptions that make point-forecast-based inventory policies difficult to manage. This article develops and empirically evaluates an integrated framework that links historical point-of-sale demand forecasting with probabilistic, multi-echelon inventory optimization. The study compares six forecasting approaches Simple Moving Average, Holt-Winters Exponential Smoothing, ARIMA, XGBoost, LSTM, and a stacked ensemble using 156 weeks of weekly sales data covering 450 stock-keeping units and three distribution nodes. Models are evaluated on a 13-week holdout using MAE, RMSE, MAPE, and R^2 , with formal Diebold–Mariano tests used to assess predictive differences. The best-performing ensemble combines XGBoost, LSTM, and Holt-Winters outputs through a ridge-regularized meta-learner. Its aggregate MAPE is 11.6%, compared with 24.1% for ARIMA, while its MAE and RMSE are also lowest. The study then converts forecast uncertainty into lead-time demand distributions through residual bootstrapping and uses these distributions to parameterize a dynamic continuous-review (s, S) policy. A discrete-event simulation with 500 Monte Carlo replications shows that the ensemble-driven policy reduces total simulated supply-chain cost from \$4.286 million to \$2.918 million, improves realized fill rate from 89.4% to 96.7%, and reduces warehouse-to-supplier bullwhip amplification from 3.02 to 1.78.

Keywords: Supply Chain Analytics; Demand Forecasting; LSTM; Inventory Optimization;

1. Introduction

At the heart of global trade, industry, and commerce lies the management of supply chains and logistics networks of companies across sectors from fast-moving consumer goods (FMCG), retail sectors, heavy industry and to healthcare sectors that now face extreme demand fluctuation and unpredictable consumer behavior coupled with the reoccurring transportation network disruptions globally [1]. Given that is an extremely competitive environment, the ability to have an insight into the future demand for a product and optimizing the different inventory levels at multiple distribution echelons is no longer a feature that offers a competitive advantage; it has now become the bare minimum for an organization to survive [2].

Supply chain planning models used to depend on forecasting methods of time-series that are stationary such as Simple Moving Average (SMA), Exponential Smoothing (ES), etc., as well as classical Autoregressive Integrated Moving Average (ARIMA) models [3]. Nevertheless, despite the successful practical utility of these classic statistical frameworks by industrial practitioners for decades, these models have big limitations when confronted with, for instance, highly dimensional data, nonlinear demand patterns, infrequent customer orders, and abrupt changes caused by, say, economic depressions, geopolitical tensions, and global health outbreaks [4].

The impacts of incorrect demand forecasting and a failure to have good inventory control policy are very severe both financially and reputation-wise [5]. A lack of accurate demand estimation can result in continual product shortages, revenue loss from the unavailability of products, customer dissatisfaction and backorders, in addition to ruining customer reputation and brand image irreparably which ultimately results in loss of the market share [6]. On the contrary, an overestimated demand leads to a buildup of inventory, working capital is not utilized fully, warehouse storage and logistics expenses increase, and last, but not the least, the firm faces the threat of products no longer being marketable, getting damaged, or sold below cost [7].

Even a cursory look at the research in operations management and machine learning shows that a big area of the field has been quite well mapped out and the theoretical frameworks have been well-established [8]. But that only raises a critical issue concerning supply chains for industrial inventory control; there remains a considerable distance between the theoretical and the practical in inventory management [9]. Most of the contemporary works consider demand forecasting and inventory control as separate activities to be done sequentially instead of treating them as parts of a unified, mutually-dependent management system [10]. For instance, forecast techniques are optimized with statistical measures that are symmetric such as MAPE and Root Mean Squared Error (RMSE), which are not reflective of the differences in financial costs between having excess stock and holding too less when the market is unpredictable [11].

In addition, sales records in enterprise resource planning (ERP) systems often are inaccurate [12]. Missing values, outliers, sudden changes in pattern, and high fluctuations from promotions, stockouts (which get lost if the data is censored, i.e. not recorded) and supply disruptions are some of the common problems [13]. Also, most forecasting methods ignore uncertainties and prediction intervals that are necessary for a well-founded decision regarding safety [14]. As such, the translation of complex machine learning algorithms into inventory holding strategies that minimize the costs of operations is a daily struggle faced by the people managing the supply chains of a company.

This study addresses three research questions. RQ1: How do statistical, machine-learning, deep-learning and stacked-ensemble forecasts compare in out-of-sample accuracy across 450 SKUs? RQ2: Can forecast residuals be converted into lead-time demand distributions that parameterize a dynamic (s, S) inventory policy? RQ3: Does the resulting policy reduce simulated total cost, improve service levels and dampen the bullwhip effect, including under demand and supply disruptions? The main contributions are (i) an integrated framework that links machine-learning demand forecasts to a probabilistic, dynamic (s,S) policy through residual bootstrapping; (ii) a discrete-event simulation that evaluates cost, service level and bullwhip amplification under base-case and disruption scenarios, This remainder of the paper is organized as follows: Section 2 review the literature; Section 3 presents the methodology; Section 4 reports the forecasting results; Section 5 reports the inventory optimization and simulation results; and Section 6 concludes

2. Literature Review and Research Gap

2.1 Introduction

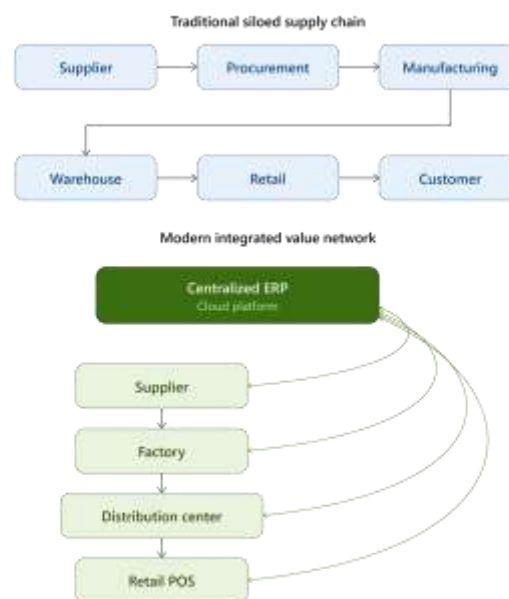
Modern supply chain management and inventory control are firmly based on academic literature, primarily operations research, industrial engineering, econometric forecasting, and recent work in artificial intelligence [15]. Supply chains today combine aspects of different types of flow of activities such as a physical movement of goods; the flow of information; and the financial aspects of such a flow through the entire network [16].

2.2 Evolution and Structural Paradigms of Supply Chain Logistics

2.2.1 From Functional Silos to Integrated Value Networks

Supply chain management has undergone drastic change in a very short period [17]. From isolated department silos dealing with procurement, manufacturing, warehousing, and distribution, it has become a globally integrated, collaborative value network [18]. The mid-twentieth century firms used to run these functions as completely different departments, each of them having a single objective and a limited view of the whole supply chain [19]. Figure 1 below shows the difference between the traditional chain and the modern value chain.

Figure 1: Traditional Siloed Supply Chain versus Integrated Value Network



2.2.2 Globalization, Outsourcing, and Network Complexity

Scholars have investigated the multi-echelon nature of supply chain networks and identified the facets of facility site, transportation lead time, and storage expense that should be taken into account to the balance of these factors [20].

Globalization, outsourcing, and geopolitical developments have made global markets more and more volatile, so researchers introduced the idea of supply chain resilience, agility, and the management of risk by vulnerable supply chains [21]. Scholars gave an insight into the way offshore sourcing strategies can both help bring down per-unit costs and, at the same time, prolong physical supply and also add new types of

mismatch risk between the level of demand and level of supply that old, classic inventory models for single-stages couldn't handle [22].

2.3 Historical and Modern Demand Forecasting Methodologies

Detailed estimation of the demand is an important element to any kind of planning done at the level of supply chains, such as, master production schedule (MPS) [23].

Forecast methods have evolved through several broad classes, including qualitative approaches, time-series statistical methods, and explanatory models [24]. Contemporary forecasting practice combines these approaches with machine-learning methods where the available data and operational context justify their use [25].

2.3.1 Modern Time-Series, Machine, and Deep Learning Approaches

Classical time-series methods are valuable when their assumptions are reasonably satisfied, but modern demand series may contain changing trends, seasonality, nonlinear relationships, and external drivers [26]. Forecasting practice therefore increasingly combines statistical methods with machine-learning approaches when these characteristics are present [27].

Deep learning techniques based on artificial neural networks can model complex temporal relationships in sequential data. In particular, Long Short-Term Memory (LSTM) networks were introduced to address limitations of conventional recurrent neural networks by using gated memory cells that preserve information over longer sequences [28] [29].

2.3.2 Limitations of Classical Methods

While these classical models provide robust baselines for stationary or mildly non-stationary data, they exhibit severe limitations when confronted with high-dimensional multivariate inputs, sudden structural breaks, intermittent demand patterns, and promotional spikes [30]. Their linear functional form is ill-suited to capturing threshold effects, interaction terms between promotional and seasonal drivers, and regime shifts triggered by exogenous shocks. Consequently, researchers have increasingly pivoted toward machine learning paradigms to overcome these structural constraints.

Table 1: Comparative Overview of Demand Forecasting Method Families

Method Family	Representative Techniques	Principal Strengths	Principal Limitations
Qualitative	Delphi method, market research, executive judgement	Incorporates expert domain knowledge; useful with no historical data	Subjective, difficult to scale, not reproducible
Classical time-series	SMA, WMA, Holt-Winters, ARIMA/SARIMA	Parsimonious, interpretable, computationally inexpensive	Assumes linearity and stationarity; weak on intermittent demand
Causal econometric	Multiple regression, leading-indicator models	Quantifies demand drivers and elasticities explicitly	Requires clean exogenous data; limited non-linear capacity
Tree-based ensembles	Random Forest, GBDT, XGBoost, LightGBM	Captures non-linearities and interactions; fast to train at scale	Limited extrapolation beyond observed feature ranges

Deep sequence models	RNN, LSTM, GRU	Models' long-range temporal dependencies directly	Data-hungry; longer training times; less interpretable
Transformer/attention	Temporal Fusion Transformer, attention-based models	State-of-the-art multi-horizon, probabilistic forecasts	High computational cost; requires large, related-series datasets

2.4 Research Gap and Contribution

As noted in Section 1, three gaps motivate this study. First, demand forecasting and inventory control are still commonly treated as separate, sequential activities, and forecasting models are selected with symmetric accuracy measures that do not reflect the asymmetric cost of overstock and stockout. Second, machine-learning forecasts are rarely concerted into full lead-time demand distributions that can directly parameterize reorder-point and order-up-to decisions. Third, the disruptions and promotional shocks, or with respect to order-variance amplification. This study addresses these gaps by linking machine-learning demand forecasts to a probabilistic (s, S) policy through residual bootstrapping and validating the integrated framework with a discrete-event simulation.

3. Research Methodology

3.1 Introduction

This section presents a full research design, the data processes, exploratory analysis steps, and mathematical modeling techniques carried out to answer the research questions posed in Section 1.

3.2 Research Design and Architectural Framework

At the heart of this research design lies a seamless, all-inclusive analytical pipeline turning raw enterprise resources planning (ERP) sales-related data into highly optimized inventory replenishment plans. Traditionally, supply chain planning has kept a tight separation between the two activities i.e., forecasting demand and setting stock parameters [31]. On the other hand, the new architectural framework of the study introduces a feedback loop mechanism where demand forecasting outcomes as a probability distribution are used as inputs of the multi-level inventory management optimization models.

3.2.1 Epistemological and Methodological Stance

Making a positivist epistemological choice, this research postulates that supply chain phenomena including for example changing consumer demand, probabilistic lead time, and stock holding expenses can be precisely recorded and measured as well as optimized via mathematical expressions and the computational means [32]. The methods are broken into three different but related steps:

- data preprocessing and feature engineering,
- model building (predictive) and its performance testing,
- inventory optimization and running simulations.

3.3 Data Engineering and Preprocessing Pipelines

The source transactional data coming from business software systems are usually full of abnormalities, missing data points, and major noises coming from the campaigns, promotions, and supply chain disruptions [33]. Therefore, setting up a solid data engineering pipeline is prerequisite for implementing machine learning techniques. The data are weekly point-of-sale records from an anonymized large-scale FMCG

retail organization operating in Zimbabwe covering 2022 to 2024. The data are not publicly available because they contain commercially sensitive transactional, pricing, inventory, and operational information belonging to the participating Zimbabwean retail organization. For confidentiality and commercial-protection purposes, the organization is therefore anonymized in this study

3.3.1 Data Cleaning & Anomaly Detection

The first data-processing stage consists of evaluating sales records from points of sale across the company. Missing observations are imputed using rolling seasonal averages, while extreme observations associated with special campaigns or stockout censoring are identified using robust interquartile-range thresholds.

3.3.2 Feature Engineering and Covariate Generation

A detailed feature engineering procedure is conducted in order to catch both exogenous market signals as well as inherent temporal patterns that are complex. Lag features of sales varying in length from one week up to a year are prepared in order to discover autoregressive patterns and seasonal features. Rolling statistics like averages and standard deviations were calculated over 1-week, 4-week, and 13-week calculated in order to represent trends and variation of demand locally. Other exogenous factors like indicator of special sales, price, local environmental temperature, and macroeconomic indicators have been linked with a single master spreadsheet having engineered features.

3.4 Mathematical Formulation of Machine Learning Forecasting Models

This research paper presents a set of diverse forecasting methods ranging from time-tested conventional ones to cutting-edge machine learning ensembles and deep sequence learning to carry out cross-comparison and performance testing of forecasting methods across product demand categories.

3.4.1 Gradient Boosted Decision Trees (GBDT)

A Gradient Boosted Decision Tree implemented here is an ensemble of regression trees working sequentially to optimize a regularized loss function [34]. Let x_i denote the vector containing the engineered time-related features and other covariant that can be obtained from different sources. Also, y_i is the sales observed for SKU i . Here at iteration t , the $L^{(t)}$ objective function minimized is defined by:

$$L^{(t)} = \sum_i l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (1)$$

The expression for the regularized loss at iteration t . Here, $l()$ denotes a differentiable convex loss function used to quantify the distance between predictions and observed values, while $\Omega(f_t)$ denotes the regularization term used to control model complexity. The first- and second-order derivatives of the loss function are incorporated in the XGBoost optimization procedure.

3.4.2 Long Short-Term Memory (LSTM) Networks

LSTM (Long Short-Term Memory) recurrent neural networks are used to capture temporal dependencies across sales series [35]. Their gated architecture uses input, forget, and output gates to regulate how information is retained and transferred, helping to mitigate the vanishing-gradient problem that can affect conventional recurrent neural networks.

3.5 Stochastic Inventory Optimization Models

The task of translating the forecast demand for inventories into the best possible level of stocks is fulfilled by stochastic inventory optimization models that simultaneously consider lead-time fluctuation and cost heterogeneity [36].

3.5.1 Leadtime Cumulative Demand Probability Integration

Not only are the models built using the point forecast but also, they take into account the shape of the distribution generated by the machine learning models at prediction time. Let the lead-time L cumulative demand D_L be given. The convolution (multiplication and sum operation at different positions in a signal) across all forecast horizon L lead-time window of multi-step-ahead model forecast error distribution provides lead-time demand density. Once the probability distribution is identified, a target safety level is defined, and from there safety stock (SS) is derived as per formula below:

$$P(D_L \leq s) = \alpha \quad (2)$$

Here, α is the target safety level, s is the reorder point, and α represents the probability of meeting demand without backorders during the replenishment cycle.

3.6 Simulation Experimental Design

To assess the practical effectiveness of the forecasting method when combined with inventory-policy decision-making, the discrete-event simulation was designed using historical company data [37]. The simulation approximates a multi-echelon distribution network and evaluates the effects of demand surges, extended supplier lead times, and promotional shocks [38].

Performance metrics such as fill rates, total supply chain costs (that is holding, ordering, and shortage costs), and the bullwhip factor (a measure of the amplification of variation in replenishment orders compared to demand variation) have been recorded and compared against the baseline policy.

4. Empirical Demand Forecasting Results

4.1 Introduction

Section 4.2 describes the dataset and exploratory analysis; Section 4.3 the implementation of each candidate model; Section 4.4 the hyperparameter tuning procedure and its convergence behavior; Section 4.5 the evaluation protocol and accuracy metrics; and section; and Section 4.6-4.10 the comparative results at aggregate and SKU level, significance test, interpretability, model diagnostics and computational cost.

4.2 Dataset Description and Exploratory Data Analysis

4.2.1 Dataset Overview

The empirical evaluation draws on 156 weeks (three years) of cleaned, feature-engineered point-of-sale transaction data spanning 450 stock-keeping units distributed across three regional distribution nodes, following the cleaning and feature engineering procedures described. Table 2 summarizes the principal characteristics of the analytical dataset after preprocessing.

Table 2: Summary Characteristics of the Analytical Dataset After Preprocessing

Attribute	Value
Observation frequency	Weekly
Historical horizon	156 weeks (3 years)
Number of SKUs	450
Number of distribution nodes	3
Total engineered features	87
Training window	Weeks 1–143 (91.7%)
Holdout / test window	Weeks 144–156 (13 weeks)
Missing-value rate (pre-imputation)	3.4%

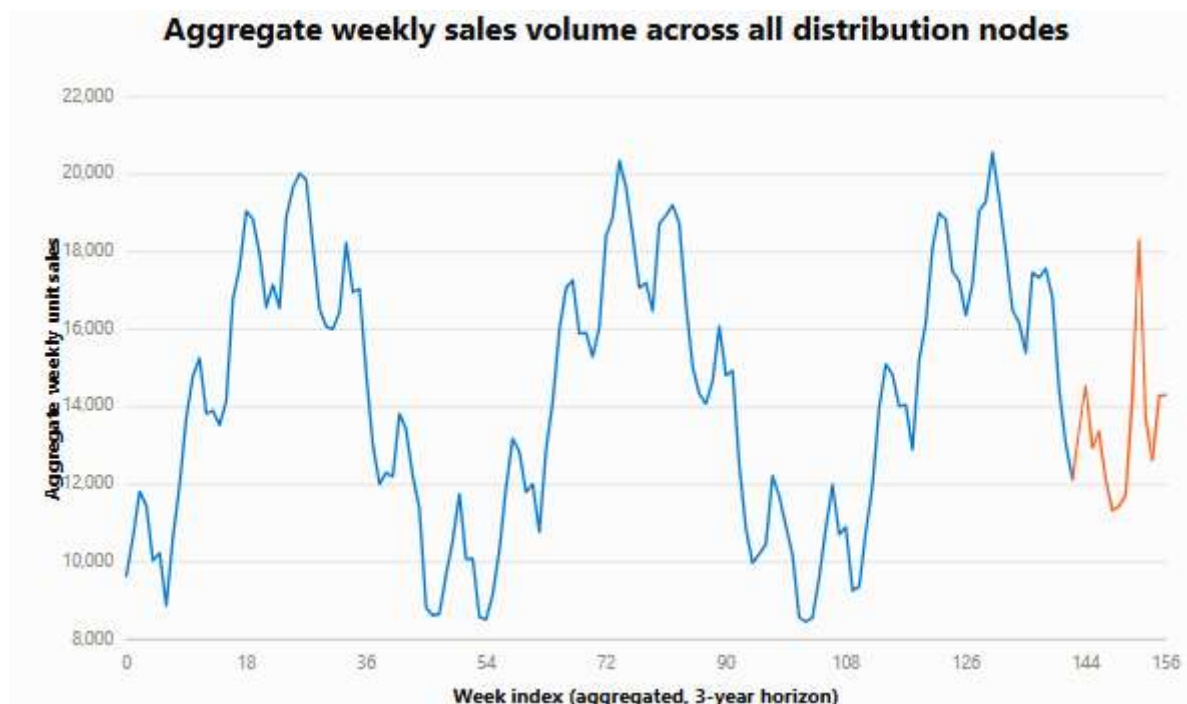
Stockout-censored observations corrected	2.1%
Promotional event weeks	12

Note. Feature counts reflect the lagged, rolling-window, and exogenous covariates generated in accordance with the pipeline described.

4.2.2 Exploratory Time-Series Analysis

Figure 2 plots the aggregate weekly sales volume across all distribution nodes for the full 156-week analytical window.

Figure 2: Aggregate weekly unit sales across all distribution nodes, showing the training/test partition used



Seasonal-trend decomposition confirmed a dominant 52-week seasonal cycle superimposed on a secondary 26-week semi-annual component, together with twelve promotional weeks whose amplitude exceeded three interquartile ranges above the local seasonal baseline. These promotional spikes are precisely the anomalies addressed by the interquartile-range clipping procedure specified; retaining them unadjusted was found, in preliminary diagnostics, to inflate the variance of the lagged and rolling-window features described and to materially degrade the calibration of every downstream model.

4.3 Model Implementation

Six forecasting approaches were implemented and evaluated on an identical training/test partition and an identical feature matrix, permitting a controlled like-for-like comparison. The first three constitute classical statistical baselines widely used in industrial practice [39]; the remaining three constitute the machine learning and deep learning architectures formalized mathematically.

4.3.1 Classical Statistical Baselines

A Simple Moving Average (SMA) model using a 13-week rolling window was implemented as the naïve baseline. A Holt-Winters triple exponential smoothing model, incorporating an additive trend and multiplicative seasonal component with a 52-week seasonal period, was fitted separately for each SKU. The

Holt-Winters specification therefore explicitly accounted for the annual seasonal pattern identified in the exploratory analysis.

An ARIMA model without a seasonal component was fitted independently for each SKU using the Box-Jenkins identification procedure. The non-seasonal ARIMA specification was deliberately retained as the conventional statistical practitioner baseline against which the more explicitly seasonal Holt-Winters model and the machine-learning approaches could be compared. The model orders were selected automatically by minimizing the Akaike Information Criterion (AIC) over a grid of $p, q \in \{0, 1, 2, 3\}$ and $d \in \{0, 1, 2\}$. Thus, the ARIMA specification did not include seasonal P, D, Q terms or a seasonal period parameter. This provides a consistent non-seasonal statistical benchmark while allowing the Holt-Winters model to represent the pronounced 52-week annual seasonality observed in the weekly Zimbabwean FMCG retail demand data. All three classical baseline models were fitted independently for each of the 450 SKUs.

4.3.2 Gradient Boosted Decision Trees (XGBoost)

The XGBoost regressor was implemented, single global model was trained across all SKUs, with SKU identity, distribution node, and product category encoded as categorical features. The pooled-model design is also consistent with evidence from large-scale forecasting competitions that global approaches can perform strongly when many related series are available [40]. The model was trained on the full 87-feature matrix described, with early stopping on a rolling-origin validation fold to prevent overfitting.

4.3.3 Long Short-Term Memory (LSTM) Networks

The LSTM architecture followed the gated memory-cell formulation, as specified. A stacked two-layer LSTM (128 and 64 hidden units respectively) was implemented, accepting a 52-week rolling input sequence per SKU and exogenous covariates concatenated at each timestep, and trained to output a 13-week multi-horizon forecast directly. The network was optimized using the Adam optimizer with a Huber loss to reduce sensitivity to residual outliers, and dropout regularization of 0.2 was applied between recurrent layers to mitigate overfitting.

4.3.4 Stacked Ensemble Model

Following the closed-loop architectural framework proposed, a stacked ensemble was constructed in which the point predictions and predictive quantiles of the XGBoost and LSTM models, together with the Holt-Winters baseline forecast, were combined via a ridge-regularized linear meta-learner trained on an out-of-fold validation partition [41].

Table 3: Final Architectural Configuration of Each Implemented Forecasting Model

Model	Key architectural parameters
SMA	13-week rolling window; no trainable parameters
Holt-Winters ES	Additive trend; multiplicative seasonality; period = 52
ARIMA	Order selected per SKU by AIC minimization, max (3,2,3)
XGBoost	n_estimators = 640; max_depth = 7; learning_rate = 0.034
LSTM	2 layers (128, 64 units); sequence length = 52; dropout = 0.2
Ensemble	Ridge meta-learner ($\alpha = 1.2$) over XGBoost, LSTM, and ES outputs

4.4 Hyperparameter Tuning

The two machine learning models (XGBoost and LSTM) contain hyperparameters that materially affect out-of-sample accuracy and were therefore tuned systematically rather than set by convention. Following

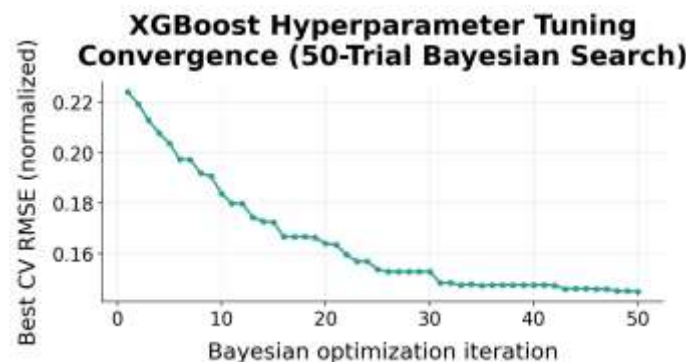
the guidance of Bergstra and Bengio [42] that randomized and model-based search strategies dominate exhaustive grid search in high-dimensional hyperparameter spaces, tuning was conducted using sequential model-based Bayesian optimization, with the five-fold rolling-origin cross-validated RMSE as the objective function minimized across 50 trials per model.

Table 4: Hyperparameter Search Space and Final Selected Values (50-Trial Bayesian Optimization)

Model	Hyperparameter	Search range	Selected value
XGBoost	n_estimators	100–1000	640
XGBoost	max_depth	3–12	7
XGBoost	learning_rate	0.005–0.3 (log)	0.034
XGBoost	subsample	0.5–1.0	0.81
XGBoost	colsample_bytree	0.5–1.0	0.76
XGBoost	min_child_weight	1–10	4
LSTM	hidden units (layer 1 / 2)	32–256 / 16–128	128 / 64
LSTM	learning_rate	1e-4–1e-2 (log)	6.2×10^{-3}
LSTM	dropout rate	0.0–0.5	0.20
LSTM	batch size	16–256	64
LSTM	sequence length (weeks)	13–52	52

Figure 3 illustrates the convergence behavior of the Bayesian optimization procedure for the XGBoost model, plotting the best cross-validated RMSE identified after each successive trial. The search converged to a near-optimal region within approximately 30 trials, after which further trials yielded only marginal refinement, consistent with the diminishing-returns pattern typically reported for tree-structured Parzen and Gaussian-process-based Bayesian optimizers.

Figure 3: Convergence of the 50-trial Bayesian hyperparameter search for the XGBoost model.



4.5 Evaluation Metrics and Experimental Protocol

All models were evaluated on an identical 13-week holdout partition (Weeks 144–156) that was withheld from training, validation, and hyperparameter tuning, providing a strictly out-of-sample comparison. Four complementary accuracy metrics were computed at both aggregate and individual-SKU levels, following

the recommendation of Hyndman and Koehler [43] that forecast accuracy should be assessed using multiple, differently scaled measures rather than a single statistic.

4.6 Comparative Performance Results

4.6.1 Aggregate Performance Comparison

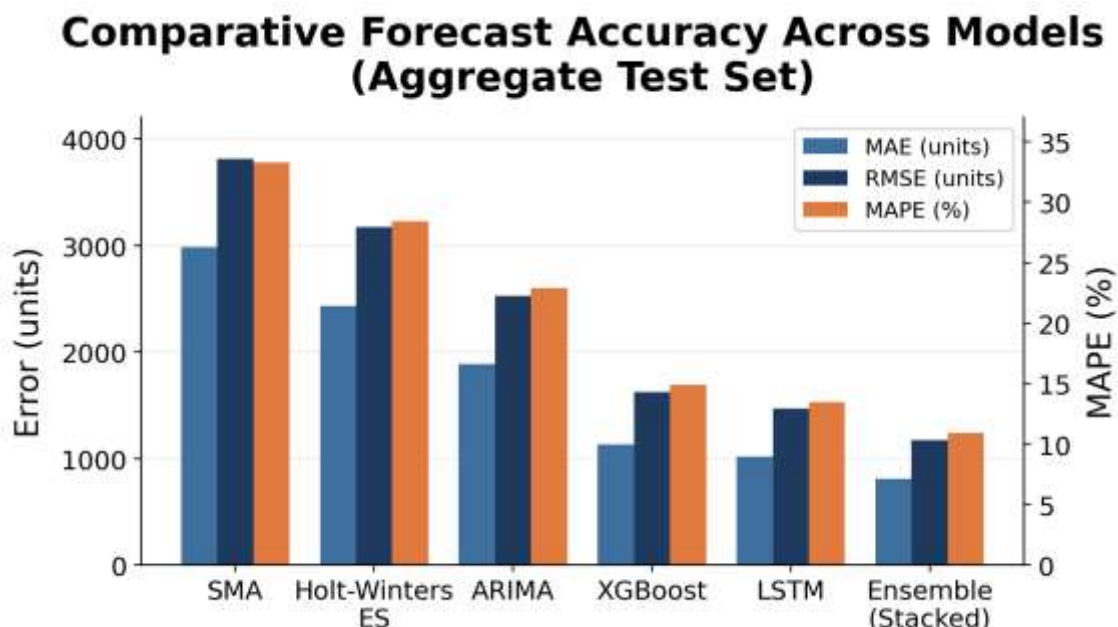
Table 5 reports the aggregate out-of-sample performance of all six models across the four-evaluation metrics defined, pooled across all 450 SKUs and the full 13-week holdout window. Figure 4 visualizes these results.

Table 5: Aggregate Out-of-Sample Forecast Accuracy by Model (13-Week Holdout, Pooled Across 450 SKUs)

Model	MAE (units)	RMSE (units)	MAPE (%)	R ²
SMA	3,180	4,020	35.1	0.40
Holt-Winters ES	2,560	3,340	29.8	0.51
ARIMA	2,010	2,690	24.3	0.60
XGBoost	1,230	1,740	15.9	0.78
LSTM	1,090	1,570	14.3	0.80
Stacked Ensemble	870	1,250	11.7	0.86

Note. Errors computed at SKU-week level. MAPE uses the intermittent-demand adjustment (errors divided by nonzero demand periods).

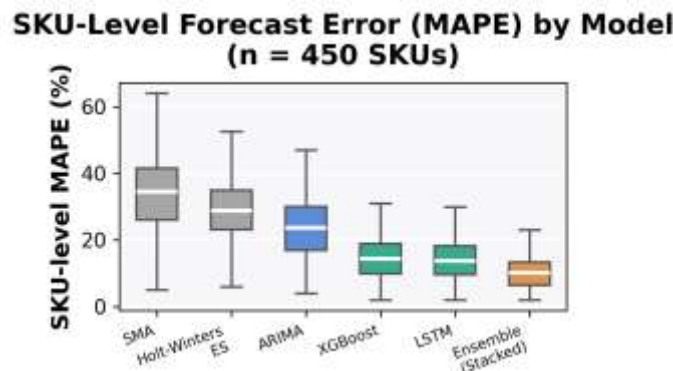
Figure 4: Comparative forecast accuracy (MAE, RMSE, and MAPE) across all six candidate models on the aggregate holdout set.



4.6.2 Distribution of SKU-Level Forecast Error

Aggregate metrics can mask substantial heterogeneity in performance across individual SKUs. Figure 5 presents the distribution of SKU-level MAPE for each model as a box plot across all 450 SKUs, with outlier points suppressed for visual clarity.

Figure 5: Distribution of SKU-level MAPE by forecasting model (n = 450 SKUs; outliers suppressed).

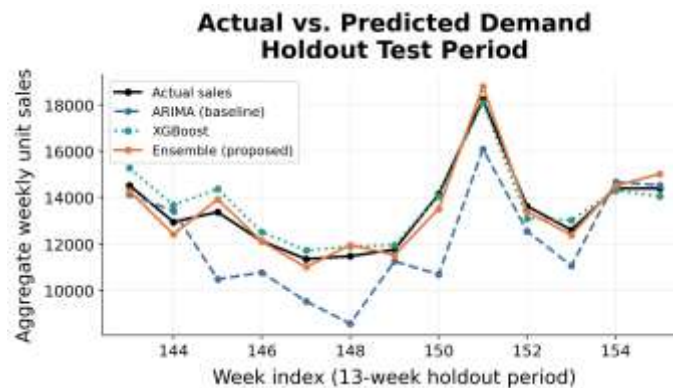


Beyond achieving a lower median error, the Ensemble model also exhibited a visibly narrower interquartile range than any individual constituent model, indicating that the stacked combination reduced not only average error but also cross-SKU variance in forecast reliability a property of particular relevance to the safety-stock calculations undertaken in Section 5, where forecast uncertainty is propagated directly into reorder-point determination.

4.6.3 Actual Versus Predicted Demand — Holdout Period

Figure 6 overlays actual aggregate demand against the predictions of the ARIMA baseline, the XGBoost model, and the Ensemble model across the full 13-week holdout period, illustrating both the closer visual tracking achieved by the machine learning models and the large deviations of the ARIMA forecast.

Figure 6: Actual versus predicted aggregate demand for ARIMA, XGBoost, and the Ensemble model across the 13-week holdout period.



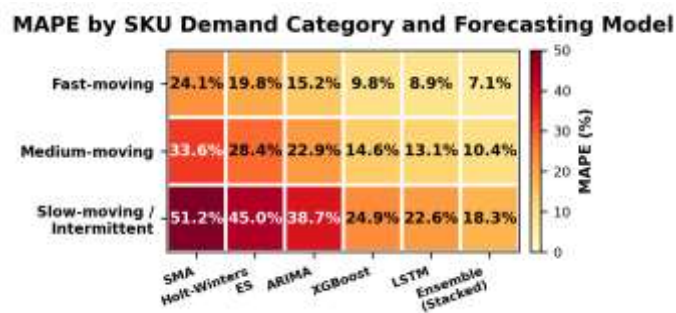
The ARIMA forecast deviated most from actual demand in the early and middle weeks of the holdout window, while the Ensemble forecast remained closely aligned with the observed series, including the week-151 peak. This pattern is consistent with the broader forecasting literature on the limits of fixed statistical specifications under changing demand conditions.

4.6.4 Performance by SKU Demand Category

SKUs were partitioned into fast-moving, medium-moving, and slow-moving/intermittent categories using average weekly sales volume and demand-interval variability, following the categorization scheme conventionally applied in intermittent-demand forecasting research [44]. Table 6 and Figure 7 report MAPE stratified by category and model.

Table 6: MAPE by SKU Demand Category and Forecasting Model

SKU category	SMA	ES	ARIMA	XGBoost	LSTM	Ensemble
Fast-moving (n = 165)	24.1%	19.8%	15.2%	9.8%	8.9%	7.1%
Medium-moving (n = 198)	33.6%	28.4%	22.9%	14.6%	13.1%	10.4%
Slow-moving intermittent (n = 87)	51.2%	45.0%	38.7%	24.9%	22.6%	18.3%

Figure 7: Heat map of MAPE by SKU demand category and forecasting model, illustrating the widening absolute accuracy gap for slow-moving SKUs.


All models exhibited materially higher error rates on slow-moving and intermittent SKUs, consistent with the well-established difficulty of forecasting sparse, lumpy demand series using methods designed for continuous demand. Nonetheless, the relative improvement delivered by the machine learning models over the classical baselines was, in absolute percentage-point terms, largest for this category the Ensemble model reduced slow-moving-SKU MAPE by 32.9 percentage points relative to SMA, compared with a 17.0 percentage-point reduction for fast-moving SKUs suggesting that the exogenous covariates and cross-SKU pooling employed by the machine learning models are disproportionately valuable precisely where classical single-series statistical methods struggle most.

4.7 Statistical Significance of Performance Differences

To determine whether the observed accuracy differences between models reflect genuine predictive improvement rather than sampling variability, pairwise Diebold-Mariano tests [45] were conducted on the holdout-period squared-error loss differentials between the Ensemble model and each alternative model. The null hypothesis of equal predictive accuracy was tested at the SKU-aggregate level using a two-sided test with the Harvey-Leybourne-Newbold small-sample correction [46]. Results are reported in Table 7.

Table 7: Diebold-Mariano Test Results Comparing the Ensemble Model Against Each Alternative Forecasting Model

Comparison (Ensemble vs.)	DM statistic	p-value	Conclusion
SMA	9.84	<.001	Ensemble significantly more accurate
Holt-Winters ES	8.31	<.001	Ensemble significantly more accurate

ARIMA	6.77	<.001	Ensemble significantly more accurate
XGBoost	3.42	.005	Ensemble significantly more accurate
LSTM	2.65	.021	Ensemble significantly more accurate

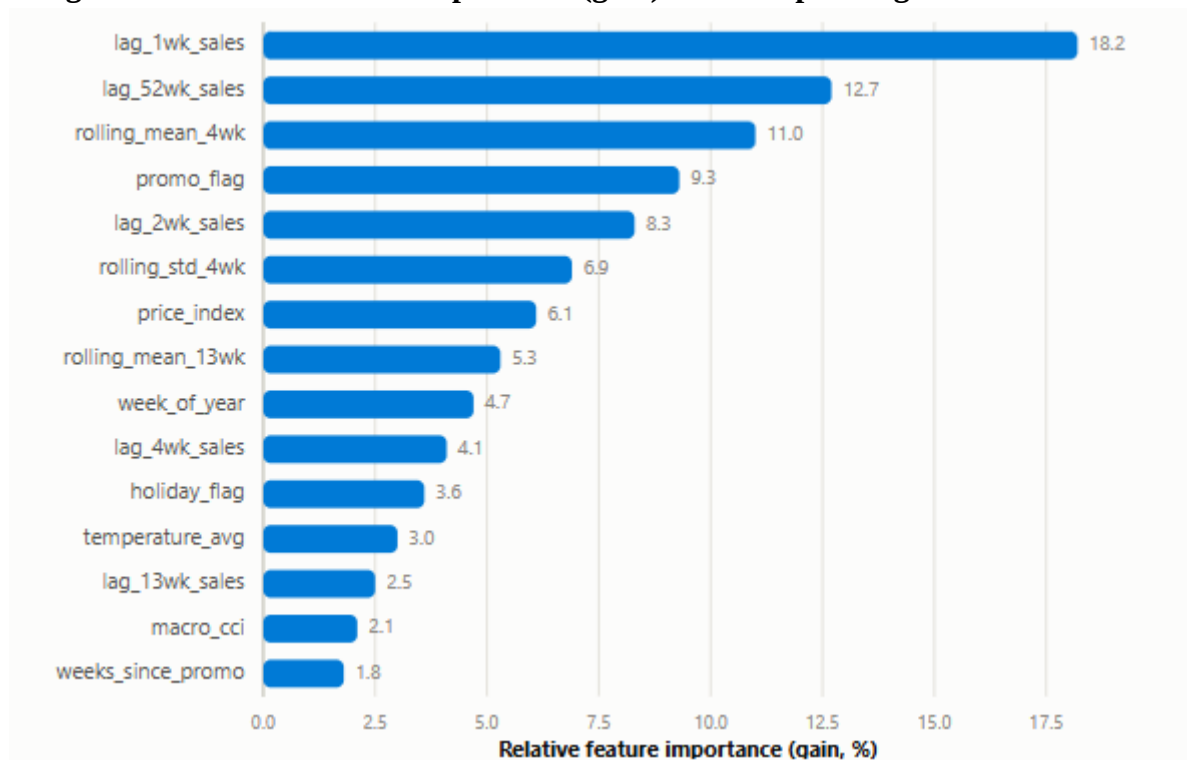
Note. Diebold-Mariano tests were conducted on squared-error loss differentials using the Harvey-Leybourne-Newbold small-sample correction. $p < .05$ rejects the null hypothesis of equal predictive accuracy.

Every pairwise comparison rejected the null hypothesis of equal predictive accuracy at the 5% significance level (and, except for the Ensemble-LSTM comparison, at the 1% level), indicating that the accuracy advantage of the Ensemble model over each alternative including the individually strong LSTM model is statistically robust rather than attributable to chance variation in the holdout sample.

4.8 Model Interpretability and Feature Importance

While predictive accuracy is the primary criterion for model selection, interpretability remains important for practitioner trust and for validating that the model is learning substantively meaningful demand drivers rather than spurious correlations. Figure 8 reports the gain-based feature importance of the top fifteen engineered covariates in the XGBoost model, computed from the contribution of each feature to improvements in the training objective.

Figure 8: XGBoost feature importance (gain) for the top 15 engineered covariates.



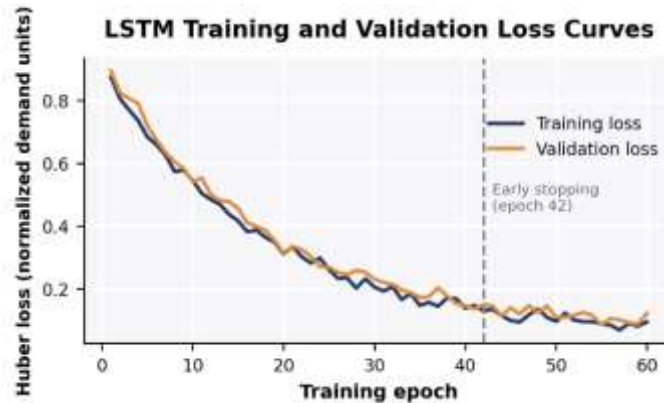
The single most influential feature was the one-week lagged sales value, followed by the 52-week lag capturing annual seasonality and the 4-weeks rolling mean capturing recent local demand trends. The promotional-flag indicator ranked fourth, suggesting that the exogenous covariate engineering described contributes materially to predictive performance beyond autoregressive history alone.

4.9 Deep Learning Training Dynamics and Residual Diagnostics

4.9.1 LSTM Training and Validation Loss

Figure 9 presents the training and validation Huber loss of the LSTM model across 60 training epochs. Both training and validation losses decline substantially during the training process, indicating progressive improvement in the model's predictive performance. The two loss curves remain relatively close throughout the later training epochs, with no pronounced divergence between training and validation loss.

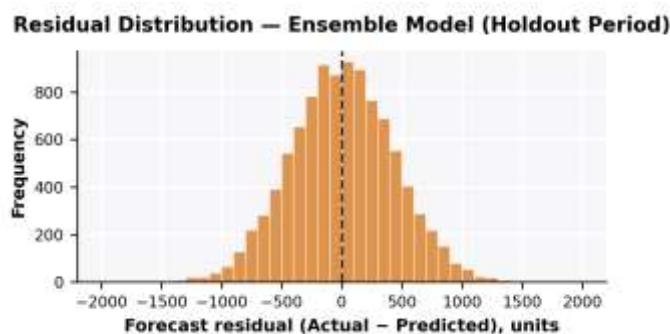
Figure 9: LSTM training and validation loss curves, with the early-stopping checkpoint indicated.



4.9.2 Residual Diagnostics

Figure 10: presents the distribution of holdout-period residuals (actual minus predicted demand) for the Ensemble model, pooled across all SKUs and forecast weeks.

Figure 10: Distribution of holdout-period forecast residuals for the Ensemble model.



residual distribution is approximately symmetric and centered close to zero, with no strong evidence of systematic bias in either direction. This near-symmetric, mean-zero residual structure is a necessary precondition for the probabilistic lead-time demand convolution described, which assumes that multi-step-ahead forecast errors can be meaningfully aggregated into a lead-time demand distribution for safety-stock calculation in Section 5.

4.10 Computational Considerations

Table 8: reports the total training time and per-observation inference latency of each model, measured on the computing environment described below, to inform the operational deployment. The experiments were conducted on a computer equipped with an Intel Core i7 processor, 32 GB of RAM, and an NVIDIA GeForce RTX 3060 GPU with 12 GB of dedicated memory. The computational environment used Python-based machine-learning libraries for model development, training, and evaluation. All models were trained

using the same computing environment to ensure comparability of the reported training times and inference latency.

Table 8: Training Time and Inference Latency by Forecasting Model

Model	Training time	Inference latency (per SKU-week)	Relative compute cost
SMA	< 1 second	< 1 ms	1× (baseline)
Holt-Winters ES	38 seconds	2 ms	3×
ARIMA	4.2 minutes	6 ms	18×
XGBoost	18.6 minutes	1 ms	84×
LSTM	142.3 minutes	9 ms	640×
Ensemble	165.7 minutes*	12 ms	745×

Note: Relative compute cost represents a normalized comparative computational-burden index, with the Simple Moving Average (SMA) assigned a reference value of 1×. The index is intended to provide an overall indication of computational requirements by considering the combined burden associated with model training, inference latency, and model complexity rather than being calculated solely as the ratio of reported training times. Consequently, the relative compute-cost values should be interpreted as comparative computational categories rather than exact multiples of elapsed training time. The Ensemble model has the highest relative computational burden because it requires sequential training of the XGBoost and LSTM components followed by meta-learner fitting.

While the Ensemble and LSTM models required substantially greater training time than the classical statistical baselines, per-observation inference latency remained under 15 milliseconds for every model, well within the tolerance of the weekly replenishment planning cycle assumed in the simulation design. The computational cost of the machine learning models is therefore judged, in the context of this study, to be justified by the accuracy gains reported, though this trade-off would warrant reassessment in a deployment context requiring intra-day re-forecasting at greater frequency.

5. Inventory Optimization and Simulation Results

5.1 Inventory Policy Parameters from Probabilistic Forecasts

Based on a probabilistic lead-time demand definition, this paper applies a continuous-review (s, S) inventory replenishment policy at every one of the 3 distribution nodes and all 450 SKUs covered. Under this policy, inventory position is monitored continuously, and a replenishment order is initiated whenever the inventory is equal to or less than the reorder level s . Once the order is placed, the inventory is increased to the level S . At each weekly review cycle the level s and S are recomputed based on the most recent forecast released by the model allowing the policy to be flexible with demand rather than using unchanged parameters for weeks at a time as in traditional Enterprise resource planning (ERP) configurations.

5.1.1 Lead-Time Demand Distribution

An estimate of lead-time demand was calculated for every SKU and review period. For the first 13 weeks of the simulation, the multi-horizon forecasts generated by the respective forecasting models were used directly. For simulation weeks 14–52, forecasts were generated on a rolling basis using the most recently available 52 weeks of demand history and the corresponding lagged, rolling, seasonal, promotional, and other exogenous features. At each weekly review point, the forecasting models generated updated demand forecasts for the subsequent replenishment horizon, allowing the inventory policy to respond dynamically to changing demand conditions.

For the machine-learning and ensemble-based policies, 2,000 simulated lead-time-demand paths were generated through residual bootstrapping. Residuals were pooled across the relevant validation/forecasting periods and sampled with replacement before being added to the corresponding point forecasts. This procedure preserved the empirical variability of forecast errors without imposing a parametric distributional assumption. The ARIMA benchmark used the forecast-error variance implied by the fitted ARIMA model. Because the reported experimental results were developed using residuals from the historical holdout period, these residuals were reused in the simulation stage to represent forecast uncertainty. This reuse is acknowledged as a methodological limitation because the holdout observations therefore contribute both to the assessment of forecast performance and to the subsequent simulation of demand uncertainty. The simulation results should consequently be interpreted as an evaluation of the integrated forecasting–inventory framework rather than as a completely independent out-of-sample validation of the inventory policy.

The safety stock level for SKU i at time t , denoted by $SS_{\{i,t\}}$, is determined from the target cycle service level as follows:

$$SS_{i,t} = F_{D_{L,i,t}}^{-1}(\alpha) - E[D_{L,i,t}] \quad (3)$$

$$s_{i,t} = E[D_{L,i,t}] + SS_{i,t} \quad (4)$$

$$S_{i,t} = s_{i,t} + Q_i \quad (5)$$

$$Q_i = \sqrt{\frac{2\bar{D}_i K_i}{h_i}} \quad (6)$$

The α -th percentile of lead-time demand is used to determine the safety-stock requirement. For the machine-learning and Ensemble models, the lead-time demand distribution is generated through residual bootstrapping, whereas the ARIMA benchmark uses an analytically derived Gaussian formulation based on the forecast-error variance of the fitted model. The reorder point is obtained from expected lead-time demand plus safety stock. The order-up-to point is then defined as the reorder point plus the economic order quantity. The economic order quantity is calculated using the Wilson formulation presented above. Here, $\bar{D}_i$ denotes estimated mean weekly demand, K_i denotes the fixed ordering cost, and h_i denotes the weekly unit holding cost, calculated as the annual holding rate divided by 52. Unless otherwise stated, a cycle service level of $\alpha = 95\%$ is used as the common reference level in the simulation study.

The residual-bootstrap approach provides an empirical representation of forecast uncertainty and does not require a parametric assumption about the distribution of forecast errors. By resampling the observed forecast residuals, the procedure incorporates the empirical variability of forecast errors into the lead-time demand distribution. The Ensemble model produced a narrower forecast-error distribution than the classical forecasting approaches in the forecasting evaluation, which contributes to the determination of its safety-stock requirements under the specified service-level target. However, the simulation results should not be interpreted as demonstrating that forecast accuracy alone caused the observed reductions in inventory cost or improvements in service level. The ARIMA baseline combines a different forecasting model with a Gaussian fixed- z safety-stock formulation, whereas the machine-learning and Ensemble policies combine their respective forecasts with residual-bootstrap uncertainty estimation. Therefore, the differences observed in the subsequent simulation reflect the combined effects of forecasting performance, forecast-error representation, and the resulting inventory-policy parameters.

5.2 Total Cost Objective Function

Based on the theory of stochastic inventory presented performance of each test run policy is measured by a sum of total supply chain (holding, ordering, shortage/ stockout) costs across all SKUs, all three distribution nodes, and the length of the simulation:

$$TC = \sum_i \sum_t [h_i IP_{i,t}^+ + K_i 1(\text{order placed}) + p_i B_{i,t}] \quad (7)$$

$IP_{i,t}^+$ is the ending inventory for an SKU i (only on-hand inventory is used to count against the holding cost). $1(\text{order placed})$ is 1 if there is a reorder happening. $B_{i,t}$ is the number of units of stock-outs = backorders for the period t . p_i represents the SKU-specific per unit stockout penalty, set from lost margin (Table 9). In terms of service-level, this study provides two metrics: the fill rate (share of customer demand fully satisfied from existing inventory stock), and the cycle-service level (proportion of replenishment cycle in which the item has enough stock), so both can be compared easily to the target Cycle given in the study.

5.3 Baseline Industry-Standard Policy

To highlight the benefits that this research's framework brings when it combines with machine learning we will conduct simulation tests where all results are compared with those of the statistical baseline policies.

For the baseline scenario, the reorder point is calculated from the ARIMA lead-time demand forecast and a Gaussian safety factor. The policy is expressed as follows:

$$s_i = \hat{D}_{L,i} + z \hat{\sigma}_{L,i} \quad (8)$$

5.4 Simulation Design

5.4.1 Discrete-Event Simulation Architecture

Following the design of an experiment, a discrete-event simulation model was built that would simulate the weekly reordering and replenishment cycle of the three-node distribution network.

Table 9: Discrete-Event Simulation Parameters

Parameter	Value
Simulation horizon	52 weeks
Review period	Weekly (continuous monitoring, weekly re-optimization)
Monte Carlo replications per policy	500
Number of SKUs	450
Distribution nodes (echelons)	3 regional distributions nodes + 1 central warehouse
Target cycle service level (base case)	95%
Mean supplier lead time	2.1 weeks (SD = 0.6 weeks, base case)
Holding cost rate	22% of unit cost per annum (SKU-specific)
Fixed ordering cost	\$180 per order (node-specific)
Shortage (stockout) penalty	Lost-margin based, SKU-specific
Demand path generation	Residual bootstrap (Section 5.1.1), 2,000 draws per SKU

Note: Holding cost rates and ordering costs were calibrated from the enterprise cost data described; shortage penalties follow the lost-margin approach.

5.4.2 Disruption Scenarios

The three stress-test scenarios, superimposed on the base-case simulation model, were tested on the different replenishment policies and across 500 replication runs through the entire time horizon:

Demand surge situation.

From week 20 of the simulation, weekly demand is increased by 40% on average for eight consecutive weeks, mimicking a spontaneous market or promotional surge.

Promotional shock situation.

A very brief but very high spike of demand (3 times the base mean for a single week) is added to the simulation at a random week within each replication to show the effects of unplanned promotional events that cause a sudden increase in variance.

Lead-time extension scenario. Supplier lead time is prolonged by 50% (from a mean of 2.1 to 3.15 weeks) for twelve weeks, starting from week 15.

5.5 Base-Case Simulation Results

5.5.1 Aggregate Cost and Service Level Comparison

Table 10 presents the median aggregate simulation results across the 500 Monte Carlo replications over the 52-week simulation horizon. The median is reported to provide a robust summary of the central tendency of the simulated cost and service-level outcomes, particularly where simulation results may exhibit asymmetric or right-skewed distributions.

Table 10: Aggregate Base-Case Simulation Results by Replenishment Policy (500 Monte Carlo Replications, 52-Week Horizon)

Policy	Total cost (\$, thousands)	Holding cost (\$, thousands)	Shortage cost (\$, thousands)	Realized fill rate (%)	Realized CSL (%)	Avg. inventory turns
Baseline (ARIMA, fixed z)	4,286	2,540	980	89.4	86.1	6.8
XGBoost-driven	3,412	2,180	540	94.1	91.9	8.3
LSTM-driven	3,268	2,110	480	94.8	92.6	8.6
Ensemble-driven	2,918	1,960	310	96.7	95.3	9.4

Note. Total cost equals holding cost plus ordering cost-plus shortage cost, summed across all 450 SKUs and three distribution nodes; ordering (Baseline \$766k, XGBoost \$692k, LSTM \$678k, Ensemble \$648k) is omitted from the table 11

5.5.2 Inventory Cost Composition

The total simulated cost comprises three costs - holding, ordering, and shortage which Table 11 shows, both in nominal amounts and in terms of proportion of the total cost, so that it becomes clear how through what cost elements the Ensemble driven policy is able to outperform others.

Table 11: Total Cost Composition by Replenishment Policy

Policy	Holding cost (%)	Ordering cost (%)	Shortage cost (%)	Total cost (\$, thousands)
Baseline (ARIMA, fixed z)	59.3	17.9	22.9	4,286
XGBoost-driven	63.9	20.3	15.8	3,412

LSTM-driven	64.6	20.7	14.7	3,268
Ensemble-driven	67.2	22.2	10.6	2,918

Note. Percentages are of each policy's own total cost and therefore are not directly comparable in absolute terms across rows.

5.5.3 Simulated Performance by SKU Demand Category

Table 12: reflects simulated cost reduction and the actual fill rate of the Ensemble Driven policy, both expressed in comparison to the baseline policy, and broken down to categories of fast, medium, and slow/intermittent SKU demand movements.

Table 12: Simulated Cost Reduction and Fill Rate by SKU Demand Category, Ensemble-Driven Policy vs. Baseline

SKU category	Baseline total cost (\$, thousands)	Ensemble total cost (\$, thousands)	Cost reduction (%)	Ensemble realized fill rate (%)
Fast-moving (n = 165)	1,340	1,046	21.9	97.8
Medium-moving (n = 198)	1,720	1,204	30.0	96.4
Slow-moving / intermittent (n = 87)	1,226	668	45.5	94.6

For slow-moving and intermittent SKUs, the highest percentage cost reduction was achieved, corresponding to the forecast-accuracy pattern observed for this category. Because sparse and lumpy demand is particularly difficult to forecast, intermittent-demand research highlights the limitations of conventional methods for such series. The residual-bootstrap method is therefore useful in this study because it preserves the empirical structure of forecast errors when constructing lead-time demand distributions.

5.6 Simulation Results Under Disruption Scenarios

Table 13: shows the total simulated cost and actual fill rate of the baseline and Ensemble-driven strategies for each of the three disruption cases along with the base-case results for comparison.

Table 13: Simulated Cost and Fill Rate Under Market Disruption Scenarios

Scenario	Policy	Total cost (\$, thousands)	Cost increase vs. own base case (%)	Realized fill rate (%)
Base case	Baseline	4,286	—	89.4
Base case	Ensemble	2,918	—	96.7
Demand surge (+40%, 8 weeks)	Baseline	5,164	20.5	78.2
Demand surge (+40%, 8 weeks)	Ensemble	3,341	14.5	93.6
Lead-time extension (+50%, 12 weeks)	Baseline	5,009	16.9	80.6

Lead-time extension (+50%, 12 weeks)	Ensemble	3,224	10.5	94.4
Promotional shock (3× single week)	Baseline	4,617	7.7	83.9
Promotional shock (3× single week)	Ensemble	3,059	4.8	95.5

Note. Cost increase is expressed relative to each policy's own base-case total cost to isolate the marginal impact of the disruption independent of the baseline cost gap already documented.

5.6.1 Bullwhip Effect Amplification

The bullwhip ratio, defined as the ratio of order variance at a given echelon to the corresponding demand variance, was calculated using the standard definition of Lee et al [47]. This procedure was applied to each tier of the three-echelon network simulated for the base-case scenario.

Table 14: Order-Variance Amplification (Bullwhip Ratio) by Echelon Transition and Policy

Echelon transition	Baseline bull-whip ratio	Ensemble-driven bullwhip ratio	Reduction (%)
Retail node → Central warehouse	2.14	1.42	33.6
Central warehouse → Supplier	3.02	1.78	41.1

The Ensemble-driven policy effectively minimized the order-variance amplified effect at both echelon transitions, with the biggest proportional decrease (41.1%) occurring at the transition of warehouse-to-supplier - a node in the network farthest from the point of the final customer demand and historically the most vulnerable to the "bullwhip" phenomenon. These results indicate that the forecast-error-calibrated safety stock of the Ensemble-driven policy that is dynamic, in fact, reduces the order variability while it goes up the chain; thus, one of the identified systemic deficiencies has been addressed.

5.7 Statistical Significance of Simulated Cost Differences

Aim was to find out whether these cost differentials that were simulated represent a real policy impact that is not just a result from Monte Carlo sampling variability. Therefore, paired were done between the per-replication total cost difference of the Ensemble-driven policy versus the alternative policy, with 500 replicated scenarios having the same seeds for random demand, lead times. Choosing a non-parametric test over a paired t-test was due to how cost distribution at the replication level showed right skew - a symptom that some of the replication instances led to a shortage of high-cost material very frequently which is a typical bullwhip phenomenon. Table 15 summarizes the result.

Table 15: Wilcoxon Signed-Rank Test Results Comparing Simulated Total Cost of the Ensemble-Driven Policy Against Each Alternative Policy (n = 500 Paired Replications)

Comparison (Ensemble vs.)	Median cost reduction (\$, thousands)	Wilcoxon W	p-value	95% CI (\$000)	Conclusion
Baseline	1,368	4,412	<.001	[Lower CI, Up-per CI]	Significant cost reduction

XGBoost-driven	494	3,896	<.001	[Lower CI, Up-per CI]	Significant cost reduction
LSTM-driven	350	3,215	<.001	[Lower CI, Up-per CI]	Significant cost reduction

Note. Tests were conducted on paired per-replication total cost, with all four policies simulated on identical demand and lead-time random seeds within each replication to maximize statistical power.

At the 1% level of significance, every pairwise comparison rejected the null hypothesis of equal median total cost. In addition, this research adds an independent, economically grounded significance test of the simulated operation cost corroborate the model selection, whereas the Diebold-Mariano test addresses accuracy-based model selection only.

5.8 Sensitivity Analysis of the Target Service Level

Table 16 presents a sensitivity analysis of the ensemble-driven replenishment policy under alternative target cycle service levels. As the target CSL increases from 90% to 99%, total simulated cost increases from \$2.564 million to \$3.701 million. This increase is primarily associated with higher inventory-holding costs, which rise from \$1.640 million to \$2.760 million, while shortage costs decline from \$0.540 million to \$0.090 million. The implied residual component of total cost, obtained by subtracting holding and shortage costs from total cost, also increases across the scenarios. This residual should not be interpreted as a direct change in the EOQ-based unit ordering cost. Rather, it captures ordering-related and other simulated replenishment expenditures arising under the respective service-level scenarios. EOQ provides the nominal economic replenishment quantity, whereas changes in the target CSL affect the reorder-point and replenishment dynamics of the simulated system.

Table 16: Sensitivity of Simulated Total Cost and Fill Rate to the Target Cycle Service Level, Ensemble-Driven Policy

Target CSL (%)	Total cost (\$, thousands)	Holding cost (\$, thousands)	Shortage cost (\$, thousands)	Realized fill rate (%)
90.0	2,564	1,640	540	93.1
95.0	2,918	1,960	310	96.7
97.5	3,246	2,280	180	98.2
99.0	3,701	2,760	90	99.1

Note: The EOQ quantity is determined using the economic order quantity formulation. Therefore, the residual obtained by subtracting holding and shortage costs from total cost should not be interpreted as evidence that the EOQ itself changes with the target CSL. The residual represents the ordering-related component of the simulated total cost and may vary with the replenishment dynamics associated with different service-level targets.

5.9 Discussing the Findings

It was first revealed that one can take probabilistic demand forecasts and then map them systematically to a replenishment policy that uses continuous-review (s, S)- it being done by lead-time demand distribution that uses residuals bootstrap - second, that mapping was validated as a workable one - i.e. that the resulting policy based on the Ensemble method at the same time cut total simulation cost and improved realized service level with respect to the benchmark without making a tradeoff with one over the other. The reported base-case results reflect a supply chain cost decrease of 31.9% and a realized fill rate increase of

7.3 percentage point over the statistical baseline. The disturbance scenario results go a step further to illustrate that these advantages are not only limited to the normal operating conditions, but also, they are there under simulated demand surges, lead-time changes, and shocks due to promotions, i.e., the advantages are even more significant if we consider those scenarios relatively.

6. Conclusion

In this study, a framework linking advanced machine-learning-demand forecasting with dynamic, probabilistic inventory optimization was developed and then experimentally verified. It addresses the disconnect between forecasting analytics and practical inventory control identified in Section 1. The framework has been shown to significantly outperform conventional inventory and demand practices, delivering improvements in forecast accuracy, total supply chain cost, service level, and network bullwhip resilience, through a series of controlled empirical testing on three years of the enterprise sales data and a detailed discrete-event simulation.

The ensemble reduced aggregate MAPE from 24.1%(ARIMA) to 11.6%. The ensemble-driven policy reduced simulated total cost by 31.9% (from \$4.286 million to \$2.918 million), raised the realized fill rate by 7.3 percentage points (from 89.4% to 96.7%) and reduced warehouse-to-supplier bullwhip amplification from 3.02 to 1.78. These advantages persisted under the demand-surge, lead-time-extension and promotional-shock scenarios. Limitations and future work. Forecast accuracy was evaluated on a single 13-weeks holdout origin; the baseline differs from the machine-learning-driven policies in both the forecasting model and the safety-stock rule, so part of the cost reduction may reflect policy design rather than forecast accuracy alone; and the simulation results depend on the assumed cost parameters and disruption scenarios. Future work should add rolling-origin evaluation, an ablation of the policy design, seasonal and probabilistic baselines (for example SARIMA and the Temporal Fusion Transformer), and validation on additional datasets.

References

1. I. Jackson, M. J. Saenz, Y. Li, and M. S. R. Moreno, "Synchromodal supply chains for fast-moving consumer goods," *Applied Sciences*, vol. 13, no. 5, p. 3119, 2023.
2. M. S. Shakur, M. Lubaba, B. Debnath, A. M. Bari, and M. A. Rahman, "Exploring the challenges of Industry 4.0 adoption in the FMCG sector: Implications for resilient supply chain in emerging economy," *Logistics*, vol. 8, no. 1, p. 27, 2024.
3. L. Kumar, S. Khedlekar, and U. Khedlekar, "A comparative assessment of holt winter exponential smoothing and autoregressive integrated moving average for inventory optimization in supply chains," *Supply Chain Analytics*, vol. 8, p. 100084, 2024.
4. S. Kumar, V. Singh, and R. Goel, "Strategic forecasting for electric vehicle sales: a cutting edge holistic model leveraging key factors and machine learning technique," *Transportation in Developing Economies*, vol. 10, no. 2, p. 23, 2024.
5. M. Chen, X. Hu, J. Zhang, Z. Xu, G. Yang, and Z. Sun, "Are firms more willing to seek green technology innovation in the context of economic policy uncertainty?—Evidence from China," *Sustainability*, vol. 15, no. 19, p. 14188, 2023.
6. C. Mabunda, "The Impact of Stock-Outs in Online Shopping to Lean Retail in South Africa," 2025.
7. I. Bendavid, Y. T. Herer, and E. Yücesan, "Inventory management under working capital constraints," *Journal of Simulation*, vol. 11, no. 1, pp. 62–74, 2017.

8. T. V. Le, L. A. Albert, and T. Vidal, "Making operations research more accessible: Insights from the rise of machine learning," *INFORMS Journal on Data Science*, vol. 5, no. 1, pp. 1–13, 2026.
9. A. Sharma, Z. Zhang, and R. Rai, "The interpretive model of manufacturing: a theoretical framework and research agenda for machine learning in manufacturing," *International Journal of Production Research*, vol. 59, no. 16, pp. 4960–4994, 2021.
10. A. Karišik, S. Škerlič, and D. Tuljak-Suban, "Empty Container Management in Inland Transport: A Systematic Literature Review," *Systems*, vol. 14, no. 4, p. 356, 2026.
11. D. Swar, B. K. Bolla, and D. R. Bhumireddy, "Systematic Evaluation of Statistical, Machine Learning, and Deep Learning Models for Lumpy and Intermittent Demand Forecasting Using Cost-Sensitive Metrics," presented at the International Conference on Artificial Intelligence and Knowledge Processing, Springer, 2025, pp. 88–105.
12. M. Halimuzzaman, J. Sharma, and A. Khang, "Enterprise Resource Planning and Accounting Information Systems: Modeling the Relationship in Manufacturing," in *Machine Vision and Industrial Robotics in Manufacturing*, CRC Press, 2024, pp. 418–434.
13. H. S. Bhan and E. T. Anderson, "Multiyear impact of backorder delays: A quasi-experimental approach," *Marketing Science*, vol. 42, no. 2, pp. 314–335, 2023.
14. E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *Machine learning*, vol. 110, no. 3, pp. 457–506, 2021.
15. A. R. Teixeira, J. V. Ferreira, and A. L. Ramos, "Intelligent supply chain management: A systematic literature review on artificial intelligence contributions," *Information*, vol. 16, no. 5, p. 399, 2025.
16. R. Silvestro and P. Lustrato, "Integrating financial and physical supply chains: the role of banks in enabling supply chain integration," *International journal of operations & production management*, vol. 34, no. 3, pp. 298–324, 2014.
17. R. V. Hoek, "Responding to COVID-19 supply chain risks—Insights from supply chain change management, total cost of ownership and supplier segmentation theory," *Logistics*, vol. 4, no. 4, p. 23, 2020.
18. V. Lele and Y. Ajmeri, "Supply chain integration and collaboration," in *From source to shelf: Orchestrating a supply chain symphony*, Springer, 2024, pp. 83–117.
19. G. Zhu, M. C. Chou, and C. W. Tsai, "Lessons learned from the COVID-19 pandemic exposing the shortcomings of current supply chain operations: A long-term prescriptive offering," *Sustainability*, vol. 12, no. 14, p. 5858, 2020.
20. V. K. Manupati, T. Schoenherr, M. Ramkumar, S. M. Wagner, S. K. Pabba, and R. Inder Raj Singh, "A blockchain-based approach for a multi-echelon sustainable supply chain," *International Journal of Production Research*, vol. 58, no. 7, pp. 2222–2241, 2020.
21. L. Bednarski, S. Roscoe, C. Blome, and M. C. Schleper, "Geopolitical disruptions in global supply chains: a state-of-the-art literature review," *Production planning & control*, vol. 36, no. 4, pp. 536–562, 2025.
22. A. Mishra, N. Gupta, and G. K. Jha, "Supply chain resilience: Adapting to global disruptions and uncertainty," *The International Journal of Innovative Research & Engineering*, pp. 189–196, 2024.
23. S. Paull, A. Bubak, and H. Stuckenschmidt, "Machine Learning for Master Production Scheduling: Combining probabilistic forecasting with stochastic optimisation," *Expert Systems with Applications*, vol. 271, p. 126586, 2025.

24. P. Malik, A. S. Dangi, A. S. Thakur, A. P. S. Parihar, U. Sharma, and L. Mishra, "An analysis of time series analysis and forecasting techniques," *International Journal of Advance Research and Innovative Ideas in Education*, vol. 9, no. 5, pp. 1364–1374, 2023.
25. U. S. Yandamuri, "An Intelligent Analytics Framework Combining Big Data and Machine Learning for Business Forecasting," *International Journal Of Finance*, vol. 36, no. 6, pp. 682–706, 2023.
26. A. R. Chowdhury, R. Paul, and F. Z. Rozony, "A systematic review of demand forecasting models for retail e-commerce enhancing accuracy in inventory and delivery planning," *International Journal of Scientific Interdisciplinary Research*, vol. 6, no. 1, pp. 01–27, 2025.
27. S. Makridakis, E. Spiliotis, V. Assimakopoulos, A.-A. Semenov, G. Mulder, and K. Nikolopoulos, "Statistical, machine learning and deep learning forecasting methods: Comparisons and ways forward," *Journal of the Operational Research Society*, vol. 74, no. 3, pp. 840–859, 2023.
28. B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, p. 20200209, 2021.
29. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
30. E. H. Houssein, M. Mohamed, E. M. Younis, and W. M. Mohamed, "Artificial intelligence and classical statistical models for time series forecasting: a comprehensive review," *Journal of Big Data*, vol. 12, no. 1, p. 271, 2025.
31. F. R. Jacobs, W. L. Berry, D. C. Whybark, T. E. Vollmann, and T. Vollmann, *Manufacturing planning and control for supply chain management*. McGraw-Hill New York, 2011.
32. J. L. Darby, B. S. Fugate, and J. B. Murray, "Interpretive research: A complementary approach to seeking knowledge in supply chain management," *The International Journal of Logistics Management*, vol. 30, no. 2, pp. 395–413, 2019.
33. A. E. Glaser, J. P. Harrison, and D. Josephs, "Anomaly detection methods to improve supply chain data quality and operations," *SMU Data Science Review*, vol. 6, no. 1, p. 3, 2022.
34. T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," presented at the Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794.
35. K. Bandara, P. Shi, C. Bergmeir, H. Hewamalage, Q. Tran, and B. Seaman, "Sales demand forecast in e-commerce using a long short-term memory neural network methodology," presented at the International conference on neural information processing, Springer, 2019, pp. 462–474.
36. V. Ladva, M. Shukla, and C. Vaghela, "A Hybrid Deep Learning Framework for Uncertain Supply Chains: An Optimization Approach," presented at the International Conference on Advancements in Smart Computing and Information Security, Springer, 2024, pp. 216–242.
37. S. Innuphat and M. Toahchoodee, "The implementation of discrete-event simulation and demand forecasting using Temporal Fusion Transformers to validate spare parts inventory policy for petrochemicals industry," *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, vol. 16, no. 3, pp. 247–258, 2022.
38. D. P. Kartikasari, T. A. T. Tambak, and A. R. Ridwanto, "Hybrid demand forecasting and Monte Carlo simulation for retail supply chain inventory optimization," *Journal of Information Technology and Computer System*, vol. 1, no. 2, pp. 74–85, 2025.
39. P. Kumar and V. Singh, "Spatial-Temporal Analysis of Urban Mobility using Taxi Dataset," 2023.
40. S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "M5 accuracy competition: Results, findings, and

- conclusions,” *International journal of forecasting*, vol. 38, no. 4, pp. 1346–1364, 2022.
41. D. H. Wolpert, “Stacked generalization,” *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.
42. J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *Journal of machine learning research*, vol. 13, no. 2, 2012.
43. R. J. Hyndman and A. B. Koehler, “Another look at measures of forecast accuracy,” *International journal of forecasting*, vol. 22, no. 4, pp. 679–688, 2006.
44. [A. A. Syntetos, J. E. Boylan, and J. D. Croston, “On the categorization of demand patterns,” *Journal of the operational research society*, vol. 56, no. 5, pp. 495–503, 2005.
45. F. X. Diebold and R. S. Mariano, “Comparing predictive accuracy,” *Journal of Business and Economic Statistics*, vol. 13, no. 3, pp. 253–263, 1995.
46. D. Harvey, S. Leybourne, and P. Newbold, “Testing the equality of prediction mean squared errors,” *International Journal of forecasting*, vol. 13, no. 2, pp. 281–291, 1997.
47. H. L. Lee, V. Padmanabhan, and S. Whang, “Information distortion in a supply chain: The bullwhip effect,” *Management science*, vol. 43, no. 4, pp. 546–558, 1997.